Domänenspezifische Evaluation und Optimierung von Datenstandards und Infrastrukturen

Dem Fachbereich Mathematik, Physik und Informatik der
Universität Bayreuth
zur Erlangung des akademischen Grades eines
Dr. rer. nat.

genehmigte Dissertation

von
Herrn Dipl. Wirtschaftsmathematiker Tobias Schneider
aus
Würzburg

1. Gutachter: Prof. Dr.-Ing. Stefan Jablonski
2. Gutachter: Prof. Dr. Bernhard Westfechtel

Tag der Einreichung: 2013/07/04
Tag des Kolloquiums: 2013/11/14

Zusammenfassung

In den letzten Jahren und Jahrzehnten ist die Menge der Daten, die in wissenschaftlichen Projekten erhoben wird, exorbitant gestiegen. Mit diesem gestiegenen Datenaufkommen geht eine Zunahme der Anforderungen an die Datenspeicherung und den Datenaustausch einher. Dazu wurden in den letzten Jahrzehnten Infrastrukturen entwickelt, welche diese Aufgaben für eine bestimmte Domäne übernehmen. Insbesondere in der Domäne der Lebenswissenschaften sind sogenannte Megascience-Plattformen entstanden, welche für den globalen Datenaustausch verantwortlich sind. Ein wesentliches Merkmal dieser Infrastrukturen sind die unterstützten Datenstandards. Um die Eignung eines Datenstandards oder einer Infrastruktur in einer Domäne zu bestimmen, werden generische Evaluationssysteme benötigt. Auf Basis der Evaluation mit diesen Systemen ist es möglich, Schwachstellen in Infrastrukturen und Datenmodellen zu erkennen und diese Mängel zu beseitigen. Grundlage hierfür ist die Strukturierung der Anwendungsdomäne in Prozessen.

In dieser Arbeit werden die Themenbereiche der Datenstandards und Infrastrukturen in der Anwendungsdomäne der Biodiversitätsinformatik getrennt untersucht. Im ersten Teil der Arbeit werden Datenstandards betrachtet. Ausgangspunkt sind die Datenstandards, da diese den Datenaustausch in einer Infrastruktur limitieren. Dazu wird ein generisches Evaluationssystem für Datenstandards entwickelt, domänenspezifisch angepasst und auf Standards der Biodiversitätsinformatik angewendet. Kern des Frameworks ist die Vollständigkeitsanalyse mit der 'Process Oriented Schema Evaluation' (POSE), welche die Basis des Frameworks bildet. Dabei konnte keiner der untersuchten Datenstandards die Anforderungen der Biodiversitätsinformatik vollständig erfüllen. Auf Basis dieser Erkenntnisse wurde mit der 'Process Oriented Data Schema Language' (PODSL) mit PODSL-Biodiv ein flexibles Datenmodell für die Datenspeicherung in der Biodiversität entwickelt.

Der zweite Teil der Arbeit behandelt Infrastrukturen. Dazu wird zunächst mit dem 'Infrastructure Evaluation Framework' (IEF) ein Evaluationssystem für Infrastrukturen auf der Grundlage von Prozessen entwickelt und auf die wichtigsten Infrastrukturen in der Biodiversitätsinformatik angewendet. Dabei ist für die Qualität einer Infrastruktur neben der technischen Umsetzung auch deren Organisation dieser maßgeblich. Im Rahmen der Evaluation konnte ermittelt werden, dass die Infrastruktur der 'Global Biodiversity Information Facility' (GBIF) als wichtigste Infrastruktur der Biodiversitätsinformatik die Kriterien an die Funktionalität einer Infrastruktur nicht vollständig erfüllen kann. Mit BDEI wird ein Konzept zur Weiterentwicklung von GBIF vorgestellt, um diese Mängel zu beseitigen.

Abstract

During the recent years and decades the amount of data collected in scientific projects has been growing dramatically. The demands on data persistence and infrastructures increased analogically. To cope with these challenges infrastructures for data exchange in specific domains have been built. Especially in the domain of life-sciences so called megascience platforms emerged which are responsible for the global data exchange. An essential attribute of these infrastructures are the supported data models and standards. To determine the appropriateness of a given data standard for a specific domain generic evaluation systems are needed. With the help of these evaluation systems weaknesses in existing standards and infrastructures can be identified and eliminated. The fundament of this evaluation is to structure the application domain into processes.

In this thesis all areas of data standards and infrastructures will be analyzed separately behind the background of biodiversity and biodiversity informatics as application domains. The first part of this thesis covers data standards. The results in the areas of evaluation of data standards and infrastructures are generic and can be applied to other domains. As data standards are a fundamental part of an infrastructure, this subject will be discussed first. To accomplish this, a generic system for the evaluation of data standards is developed and applied on the most important data standards in the domain of biodiversity. Basis of this system is the analysis of completeness of a data standard with the 'Process Oriented Schema Evaluation' (POSE). It is shown that none of these data standards can fulfill the demands on a standard of the domain of biodiversity informatics completely. On basis of these findings PODSL-Biodiv is developed with the 'Process Oriented Data Schema Language' (PODSL) as a flexible data standard for the domain of Biodiversity informatics.

The second part of this thesis is concerned with infrastructures. At first 'Infrastructure Evaluation Framework' (IEF) is developed as an evaluation system for infrastructures on the basis of processes and applied to important infrastructures in the domain of biodiversity informatics. It is shown, that the organization of an infrastructure has a large impact on the quality beside the pure technical capabilities. The 'Global Biodiversity Information Facility' (GBIF) network is identified as the most important infrastructure in the domain of biodiversity. Nonetheless even the GBIF network cannot fulfill the requirements on an infrastructure completely. To overcome these deficits, the concept of BDEI as a further development of GBIF is proposed.

Danksagung

Zuallererst möchte ich mich ganz herzlich bei meinem Doktorvater Prof. Dr.-Ing Stefan Jablonski bedanken, der mir wiederholt die Chance gegeben hat, an seinem Lehrstuhl mitzuwirken. So konnte ich unter seiner Anleitung die Welt der Prozessmodellierung und der Datenbanken entdecken und in verschiedenen Projekten einsetzen. Bei der Erstellung der Doktorarbeit hat mich seine konstruktive Kritik immer wieder gefordert womit sich nicht nur die Qualität und der Forschungsstand diese meiner Arbeit, sondern auch meine fachlichen Kompetenzen verbessert haben. Auch in persönlicher Hinsicht war Prof. Jablonski für mich stets eine wertvolle Person, deren Rat und Meinung mir in vielen Lebenslagen sehr wichtig war und noch immer ist. Ich denke gerne an unsere fachlichen, und mit besonderer Vorliebe auch auch an unsere persönlichen Gespräche zurück, die mich ausgesprochen geprägt haben. Vielen Dank!

Darüber hinaus bedanke ich mich bei allen meinen Kollegen vom Lehrstuhl Angewandte Informatik IV der Universität Bayreuth für die gute Zusammenarbeit und die wohltuende Atmosphäre am Lehrstuhl. Mein Dank geht in diesem Zusammenhang besonders Dr. Bernhard Volz und Claudia Piesche. Dr. Bernhard Volz war für mich stets in fachlichen Fragestellungen ein höchst kompetenter und damit wichtiger Ansprechpartner. Dies geht sogar noch vor die Zeit des IBF-Projekts zurück, in er ich mit ihm zusammen die Übungen in objektorientierter Programmierung für Hörer anderer Fächer geleitet habe. Claudia Piesche schätze ich für ihre offene, unkomplizierte Art und die vielen Gespräche, die mir einen tieferen Einblick in das Leben, das Universum und den ganzen Rest geben konnten. Ich möchte an dieser Stelle Christoph Günther, Robin Hecht, Bastian Roth, Mattias Jahn, Mischa Nietfeld, Michael Zeising, Stefan Schönig und Lars Ackermann für eine konstruktive Arbeitsatmosphäre, fachliche Unterstützung und ein sehr gutes kollegiales Verhältnis danken, genauso wie meinen ehemaligen Kollegen Dr. Michael Igler, Dr, Matthias Faerber, Dr. Manuel Götz, Dr. Stefanie Meerkamm, Dr. Ramzan Talib, Dr. M. Abdul Reman, Florent Jochaud, Peter Karich, Tobias Jansen und Sebastian Dornstauder.

Ich bedanke mich bei Christine Leinberger und Kerstin Haseloff, die mir in organisatorischen und administrativen Fragestellungen stets zur Seite standen, sowie Bernd Schlesier der mich bei der Beschaffung von Hardware stets unterstützt hat. Außerdem danke ich Georg Rollinger, der mich bei der Programmierung von DiversityMobile unterstützt und sich dabei deutlich mehr engagiert hat, als ich es von einer wissenschaftlichen Hilfskraft erwarten durfte.

Es hat mir über die letzten Jahre hinweg sehr viel Freude bereitet iim IBF-Projekt zu arbeiten und wertvolle interdisziplinäre Erfahrungen zu machen. Deshalb mpchte

ich an dieser Stelle allen Projektmitarbeitern danken. Dabei möchte ich ganz besonders Dr. Markus Weiss, Wolfgang Reichert und Dieter Neubacher für die gemeinsame Arbeit bei der Entwicklung von DiversityMobile und die unzähligen Telefonate bedanken, die es stets ermöglicht haben, fachliche Probleme zeitnah zu lösen. Ich danke Tanja Weibulat, Dr. Dagmar Triebel und Dr. Konstanze Bensch für einen guten fachlichen Austausch und Unterstützung in domänenspezifischen Fragestellungen. Ich danke Josef Simmel, Wolfgang Ahlmer, Jürgen Klotz, Wolfgang von Brackel, Alexander Guhr, Dr. Alexandra Kehl für exzellente Hilfestellung in domänenspezifischen Fragestellungen und den Test von DiversityMobile. Ich möchte außerdem Dr. Dagmar Triebel, Prof. Dr.-Ing Stefan Jablonski, Prof. Dr. Peter Poschlod und Prof. Dr. Rambold für die Koordination der Projektaufgaben danken.

Ich bedanke mich außerdem sehr herzlich bei Prof. Dr. Bernhard Westfechtel für die Übernahme der Aufgabe als Zweitgutachter und bei Prof. Dr. Thomas Rauber für die Übernahme des Vorsitzes des Prüfungsausschusses, sowie Prof. Dr. Gerhard Rambold als viertem Prüfer.

Zuletzt möchte ich mich bei allen meinen Testlesern bedanken, die in unermüdlicher Arbeit meine Dissertation auf Tippfehler und sprachliche Mängel untersucht haben.

Diese Promotion wurde in Rahmen des DFG-LIS Projektes 'Aufbau eines Informationsnetzes für biologische Forschungsdaten von der Erhebung im Feld bis zur nachhaltigen Sicherung in einem Primärdatenrepositorium' (IBF) unter den folgenden Fördernummern durchgeführt:

- 2009: GZ: INST 106535/1-1, INST 21946/1-1, INST 2850/1-1, INST 747/1-1
- ab 11/2011: GZ: HA 2598/15-2, JA 561/4-2, PO 491/6-2, RA 731/11-2

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Definitionsverzeichnis

Kapitel 1

Einleitung

Seit der Einführung der relationalen Datenbanken [34] werden diese in den verschiedensten Bereichen angewendet. Dabei wird das Problem, Daten aus verschiedenen Quellen zu kombinieren und Nutzern eine einheitliche Sicht auf diese Daten zur Verfügung zu stellen, als das Datenintegrationsproblem bezeichnet [138]. Dieses konnte als eine der ersten Herausforderungen [267] im Datenbankenbereich identifiziert werden. Die Lösung dieses Problems ist für praktische Fragestellungen von entscheidender Bedeutung [94, 138] und hat auch heute nichts an Aktualität verloren. So wurden gemäß einer Anfrage auf Scholar-Google allein 2012 über 100.000 Artikel zu diesem Themenkomplex veröffentlicht.

Das Problem der Datenintegration ist allerdings häufig für Systeme – beispielsweise innerhalb eines Wirtschaftsunternehmens – formuliert, bei denen verschiedene Produktivdatenbanken des Unternehmens zu einer typischen Datewarehousingstruktur vereint werden [30]. Man kann in den typischen Anwendungsfällen von Datenintegration davon ausgehen, dass es bekannt ist, welche Datenspeicher an einer Infrastruktur teilnehmen und auch dass die Gründe der Teilnahme dafür bekannt sind. Das heißt, der Teilnehmerkreis einer Infrastruktur ist in sich abgeschlossen.

Betrachtet man hingegen das Erstellen einer einheitlichen Sicht für eine offene Infrastruktur, wird die Fragestellung deutlich komplexer. Hierbei ist die Anzahl der Teilnehmer am System von vornherein nicht bekannt und auch als flexibel zu betrachten. Die Motivation zur Teilnahme an einem entsprechenden System ist im Allgemeinen bei den jeweiligen Teilnehmern sehr unterschiedlich, da diese aus der Perspektive ihres jeweiligen Projekts Daten erheben, sammeln, speichern oder auswerten möchten. Diese Projekte haben meistens nur gemein, dass sie derselben Anwendungsdomäne angehören und eine gemeinsame Infrastruktur nutzen. Diese Infrastruktur gibt ein Datenschema vor, das von allen Teilnehmern des Systems verwendet

werden muss. Folglich ist die Entwicklung von domänenspezifischen Datenstandards und Austauschprotokollen eine Schlüsseltechnologie in diesem Bereich. Dabei ist es unerheblich, ob ein Schema primär zur Datenübertragung oder zur Datenspeicherung verwendet wird, weshalb im Folgenden allgemein der Terminus Datenstandard[1] verwendet wird.

Systeme zum Datenaustausch sind als zentrale Systeme in der Wissenschaft zu finden, wie z.B. BRAHMS für die Herbarienverwaltung [235] oder aber die 'Global Biodiversity Information Facility' (GBIF) [82] und 'Encyclopedia of Life' (EOL) für die Domäne der Biodiversitätsforschung. In der Biodiversitätsforschung übernehmen diese Infrastrukturen in ihrer Funktion als 'Megascience-Plattformen' die wichtige Aufgabe der Langzeitarchivierung von Daten [242].

Diese offenen Infrastrukturen zum Datenaustausch werden mit ganz anderen Herausforderungen im Bezug auf die Datenintegration konfrontiert, als diese beispielsweise in einer klassischen Data-Warehousing-Lösung (vgl. [30]) auftreten. Da diese mit einer flexiblen Anzahl an Teilnehmern aus verschiedenen Teilbereichen einer Disziplin arbeiten, müssen sie ein gemeinsames Schema zur Verfügung stellen. So treten zwangsläufig Datenverluste auf, wenn das gemeinsame Datenschema eines Datenspeichers für Daten aus einem spezifischen Projekt keine Möglichkeit der Speicherung bereitstellt. Darüber hinaus ist es möglich, dass Daten falsch interpretiert werden, wenn die Speicherung der Daten zwar möglich ist, aber die Daten in einen anderen Kontext gesetzt werden und somit ein Bedeutungswandel stattfindet.

Die vorliegende Arbeit hat es sich zur Aufgabe gemacht, den Datenaustausch insbesondere in der Biodiversitätsinformatik zu verbessern, so dass Daten vergleichbar sind und auch heterogenen Quellen ausgewertet werden können. Dazu werden zwei verschiedene Ansätze verfolgt:

- Die Verbesserung der Datenstandards
- Die Verbesserung der Infrastrukturen

Ein wesentlicher Schritt ist die Evaluation vorhandener Standards und Infrastrukturen. Diese Problemstellung wird aus einer prozessorientierten Perspektive betrachtet und im Kern auf die 'perspektiven-orientierte Prozessmodellierung' (POPM) [114] zurückgeführt.

[1] Streng genommen beinhaltet der Begriff 'Standard' die Zertifizierung durch ein Fachgremium.

1.1 Ausgangslage

Im Rahmen des von der DFG geförderten Projektes (Fördernummern Teilprojekt Jablonski: INST 106535/1-1 und JA 561/4-2) 'Aufbau eines Informationsnetzes für biologische Forschungsdaten von der Erhebung im Feld bis zur nachhaltigen Sicherung in einem Primärdatenrepositorium' (IBF) [116, 52, 241, 260] konnten im Projektverlauf für den Bereich der Biodiversitätsforschung und der Biodiversitätsinformatik die Anforderungen im Hinblick auf die Datenspeicherung und Datenübertragung analysiert werden. Dabei konnten sehr verschiedenartige Anforderungen identifiziert werden. Es verwundert nicht, dass insbesondere im Bereich der Biodiversitätsinformatik offene Infrastrukturen mit einer unbestimmten Anzahl von Teilnehmern weit verbreitet sind. Dies liegt unter anderem daran, dass aufgrund der sehr unterschiedlichen Anforderungen Wissenschaftler traditionell die Arbeit mit eigenen Schemata und Speichersystemen bevorzugen und somit der Datenaustausch zwischen verschiedenen Forschungsgruppen erst langsam Einzug hält. Die proprietären Schemata folgen im Allgemeinen keinem Standard und spiegeln hauptsächlich die Anforderungen des aktuellen Projektes wieder, so dass häufig die Daten einer Forschergruppe über verschiedene Projekte hinweg nicht miteinander kompatibel sind. Dabei werden die Daten bislang häufig nur teilweise oder nicht digitalisiert, was den Austausch zusätzlich erschwert.

Die Notwendigkeit zur Abkehr von dieser gängigen Praxis ist ein stetiger Prozess, der nicht zuletzt durch die wachsenden Anforderungen an das wissenschaftliche Arbeiten wie z.B. die Empfehlungen der DFG im Bezug auf die Datenhaltung von Primärdaten aus wissenschaftlichen Projekten [49] in Gang gesetzt wurde. Auch Organisationen wie 'Biodiversity Information Standards' (TDWG) nehmen sich der Aufgabe an, Daten aus verschiedenen Projekten zu vereinheitlichen [232], und fördern damit eine standardisierte, zentralisierte Datenspeicherung für die gesamte Domäne der Biodiversitätsinformatik. So haben sich je nach Anforderungen aus den Projekten seit Beginn der 80er Jahre [41] eine Reihe von Standards entwickelt, welche von Organisation wie der TDWG als Standards ratifiziert werden [232]. Außerhalb der TDWG wurden mit der 'Ecological Metadata Language' (EML) [128], der 'Extensible Observation Ontology' (OBOE) [149], oder dem 'Common Data Model' (CDM) [58] weitere Möglichkeiten der strukturierten Datenspeicherung geschaffen. Für die wichtigsten Standards wie z.B. 'DarwinCore'(DwC) und 'Access to Biological Collection Data' (ABCD) wurden Erweiterungen für Subdomänen entwickelt, die auch teilweise in neuere Versionen des ursprünglichen Standards aufgegangen sind [232]. Zusätzlich zu diesen offiziellen Erweiterungen der TDWG wurden für DwC durch

Communities von Subdomänen weitere Erweiterungen geschaffen [1, 67], welche nicht mit den offiziellen Erweiterungen der TDWG kompatibel sind. Darüber hinaus gibt es eine Vielzahl von Programmen zur Unterstützung der wissenschaftlichen Arbeit, die nur teilweise mit den Anforderungen der Wissenschaftler im Bezug auf die Datenspeicherung harmonieren und häufig auch nicht oder nur teilweise die gängigen Datenstandards unterstützen. Einen Überblick über Programme zur Unterstützung der Arbeit in der Biodiversitätsinformatik ist über den 'Biodiversity Service and Application Tracker' verfügbar [57].

Aktuell steht dem Wissenschaftler in der Biodiversitätsforschung damit eine Vielzahl von Formaten zur Verfügung. Andererseits erhält ein Wissenschaftler auf eine Anfrage die Daten nicht in einem Format zurück, welches eine weitergehende Verarbeitung unterstützt. Die Anforderungen an die Datenspeicherung sind so komplex, dass nicht sofort erkennbar ist, ob ein Datenspeichersystem die Anforderungen des Wissenschaftler erfüllt, beziehungsweise welches System der Datenspeicherung die Anforderungen am Besten erfüllt. So wurde beispielsweise für die Speicherung beschreibender Daten in der Biodiversitätsforschung in [90] ein umfassender Katalog mit über 200 Anforderungen allein für diese Datenklasse aufgestellt, der bisher von keinem Datenstandard vollständig implementiert werden konnte. Prinzipiell steht somit ein Wissenschaftler vor der Frage, ob eine der bereits existierenden Speichermöglichkeiten für sein Forschungsvorhaben geeignet ist oder ob er für seine speziellen Anforderungen ein eigenes Schema entwickeln muss. Hierbei steht aktuell keine Methode zur Verfügung, welche es erlaubt nach einer Auswahl von Kriterien die Eignung eines Datenstandards im Bezug auf die Anforderungen eines Projektes zu überprüfen. Da aber gerade die Wahl eines Datenstandards eine grundlegende Entscheidung ist, die zu einem späteren Zeitpunkt nur mit einem sehr hohem Aufwand korrigiert werden kann, ist eine Unterstützung des Wissenschaftlers an dieser Stelle von entscheidender Bedeutung.

Falls der Wissenschaftler zu dem Schluss gekommen sein sollte, dass die Entwicklung eines eigenen Schemas notwendig ist, entstehen Kompatibilitätsprobleme, sobald er seine Daten mit anderen Wissenschaftlern austauschen muss oder seine Daten in einem anderen Kontext wiederverwendet werden sollen. In diesem Fall muss er für die Übertragung der Daten ein geeignetes Schema identifizieren. Auch die langfristige Speicherung in einem zentralen Repositorium ist nur möglich, wenn das proprietäre Schema mit dem Datenschema des Repositoriums kompatibel ist. Ist keine vollständige Kompatibilität gewährleistet, entstehen bei der Übertragung in das Zielschema Datenverluste oder aber auch Artefakte, welche die Originaldaten verfälschen. Ein

Wissenschaftler muss über eine Methode verfügen, mithilfe derer dieser die Integrität der Daten gewährleistet kann oder aber zumindest Intergrationsmängel identifizieren kann. Deshalb wird eine Möglichkeit benötigt, um die Qualität von Datenstandards zu evaluieren.

Auf der anderen Seite hat ein Konsument von Daten ein Interesse an integeren und möglichst vollständigen Daten, die bei dem Bezug aus verschiedenen Quellen zudem vergleichbar sein müssen. Hinter dem Zugriff auf diese Daten steht eine komplette Infrastruktur von der ursprünglichen Datenerhebung, der Zwischenspeicherung in verschiedenen Systemen (z.B. PC, Repositorium eines Instituts), der Datenübertragung an einen zentralen Datenspeicher und letztlich die Datenübertragung an ein Portal mit der anschließenden Distribution. Bei dem Zusammenspiel einer derartigen Vielzahl von Komponenten ist der Verlust von Daten beziehungsweise der Genauigkeit von Daten, sowie der mögliche Bedeutungswandel von Daten ein nicht zu unterschätzendes Problem. Darüber hinaus können Daten beim Transfer durch eine Infrastruktur dupliziert werden, was die Auswertung der Daten verfälscht. Die Dokumentation von Herkunft und Transformationen innerhalb einer Infrastruktur (Data Provenance) ist von entscheidender Bedeutung. Aus diesen Gründen ist die Qualität einer Infrastruktur besonders wichtig. Da es aktuell keine Möglichkeit gibt, die Qualität einer solchen Infrastruktur zu messen, besteht an der Entwicklung eines Instruments zur Evaluation von Dateninfrastrukturen ein erheblicher Bedarf.

1.2 Problemstellung

Es lassen sich aus der Perspektive des Datenanbieters und Datenkonsumenten folgende, grundlegende Fragestellungen identifizieren:

P1 Wie kann die Qualität eines Datenstandards bewertet werden?

P2 Wie sieht ein geeignetes Datenschema für die Biodiversitätsinformatik aus?

P3 Wie kann die Qualität einer Infrastruktur bewertet werden?

P4 Wie ist eine geeignete Infrastruktur für die Biodiversitätsinformatik aufgebaut und aus welchen Komponenten besteht diese?

Infrastrukturen sind für den Datenaustausch zwischen verschiedenen Teilnehmern verantwortlich. Ausgangspunkt hierfür ist im allgemeinen eine Organisation in der Anwendungsdomäne, welche die Datenspeicher koordiniert. Teil der Infrastruktur

sind damit alle zum Datenaustausch benötigten technischen Vorrichtungen, Konventionen wie z.B. das Austauschformat sowie die soziale Struktur der Teilnehmer.

Hierbei kann der Eindruck entstehen, dass für einen Datenanbieter die Qualität des Datenstandards im Vordergrund steht, wogegen ein Datenkonsument primär an der Qualität der Infrastruktur interessiert ist. Diese Denkweise ist zu kurzsichtig, da ein Datenanbieter über eine Infrastruktur die Daten in einen zentralen Speicher überführt und einem Datenkonsumenten an möglichst guten Datenspeicher- und Übertragungssystemen gelegen ist, da eben auch die von ihm gesuchten Informationen zunächst in einem entsprechenden System abgelegt werden müssen. Es ist aber ersichtlich, dass zwei verschiedene Themenbereiche vorliegen, die getrennt voneinander zu behandeln sind.

1.2.1 Datenstandards

Ein Datenstandard ist eine zentrale Komponente einer Infrastruktur, die im Hinblick auf ihre Eignung im Bezug auf die Speicheranforderungen eines spezifischen Projektes möglichst vollständig analysiert werden muss. Für die Analyse eines Datenstandards müssen zunächst Kriterien identifiziert werden, die eine Messung der Qualität erlauben. Dabei werden unter anderem Antworten auf folgende Fragen gesucht, die primär das Schema eines Datenstandards betreffen:

1. Was kann ein Datenstandard inhaltlich aufnehmen (Vollständigkeitsanalyse)?
2. Wie flexibel ist ein Datenstandard bezüglich neuer Anforderungen (Flexibiltätsanalyse)?
3. Ist der Datenstandard redundant (Redundanzanalyse)?
4. Welche Strukturen im Datenschema speichern die Herkunft der Daten (Data Provenance-Analyse)?
5. Sind alle Kriterien gleich wichtig (Abwägung der Gewichtung)?

Insbesondere die Antwort auf die ersten beiden Fragen ist im Anwendungsbereich der Biodiversitätsinformatik von entscheidender Bedeutung. Die Daten aus einem wissenschaftlichen Projekt sollen nach Möglichkeit ohne Informationsverlust in eine gemeinsame Sicht übertragen werden. Um dies beurteilen zu können, muss zunächst eine Methode vorhanden sein, welche es ermöglicht, den Inhalt eines Datenspeichersystems zu messen. Dabei ist es das primäre Ziel, Aussagen über einen Gegenstand

aus der realen Welt strukturiert zu speichern. Zu jedem Datensatz, der in einem System gespeichert ist, existiert eine Aussage in einer Welt, die mit diesem Datensatz korreliert (vgl. semantisches Datenmodell [96]). Das heißt, es muss konkret messbar sein, ob eine Aussage, die im Rahmen eines solchen wissenschaftlichen Projektes getroffen wurde, ohne Informationsverlust gespeichert werden kann. Dementsprechend ist ein Kriterium, das die Vollständigkeit eines Datenschemas misst, immer im Bezug auf einen Anwendungsfall zu betrachten. Das Ergebnis einer Vollständigkeitsanalyse ist somit nicht, dass ein Datenstandard an sich gut oder schlecht ist, sondern dass dieser im Hinblick auf einen Anwendungsfall gut oder schlecht geeignet ist.

Darüber hinaus soll ein Datenstandard nicht nur für ein einzelnes Projekt die Grundlage der Datenspeicherung bieten, sondern für eine Vielzahl an unterschiedlichen Projekten, die einem steten Wandel unterworfen sind. Somit ist nicht nur von Bedeutung, wie gut die jetzigen Projektanforderungen im Datenschema erfüllt sind, sondern auch wie robust das Datenschema im Hinblick auf zukünftige Änderungen ist und wie gut es die Anforderungen verschiedener Projekte erfüllt. Im Gegensatz zu der vorherigen Fragestellung ist es bei dieser Fragestellung von Bedeutung, wie gut die Domäne an sich von einem Datenstandard modelliert wird. Es handelt sich somit um ein allgemeines Kriterium. Für dieses Kriterium ist es notwendig, eine Methode zu entwickeln, die messen kann, ob ein Datenstandard diesen Anforderungen im Bezug auf die Flexibilität genügt.

Eine übliche Vorgehensweise im Biodiversitätsbereich zur Lösung dieser Problematik ist es, einen Standard zu entwickeln, der um domänenspezifische Erweiterungen ergänzt wird (wie z.B. bei DwC oder dem ABCD-Standard [232]). Allerdings entsteht durch diese Praxis das Problem, dass für jeden Anwendungsfall, der nicht ausreichend im Datenstandard abgebildet werden kann, eine projektspezifische Erweiterung entwickelt werden muss. Somit kann eine Vielzahl von Erweiterungen entstehen, die auch sich überschneidende Themengebiete modellieren aber zueinander inkompatibel sind. Ist dieser Fall eingetreten, können die Daten nicht mehr miteinander verglichen werden. Darüber hinaus sind die verschiedenen Datenstandards der TDWG [232] einer Versionierung unterworfen. Der Kernstandard muss weiterentwickelt werden, damit die Kompatibilität der erfassten Daten gewährleistet bleibt. Diese Probleme zeigen, dass die aktuellen Datenstandards in der Biodiversitätsinformatik über erhebliche Mängel im Hinblick auf die Flexibilität verfügen und diese Praxis durch eine bessere Methode ersetzt werden muss. Dies ist auch das primäre Kriterium bei der Entwicklung eines Datenstandards für die Biodiversitätsinformatik.

Die weiteren Fragen zielen auf die Konsistenz eines Datenschemas und die Unterstützung von Data Provenance ab. Ein Framework zur Evaluation von Datenstandards wird auf Basis der 'Process Oriented Schema Evaluation' (POSE) für die Vollständigkeitsanalyse in Kapitel 4 neu eingeführt.

Überdies ist zu beachten, dass ein Datenspeichersystem nicht zwangsläufig ein relationales Datenbankmanagementsystem (RDBMS) sein muss. Insbesondere bei der Datenübertragung im Biodiversitätsbereich ist die Verwendung von Standards auf Basis von XML (DwC, ABCD) üblich [232]. Allerdings werden auch Ontologien verwendet [149] und auch die Anwendung neuartiger Speichersysteme wie z.B. NoSQL Datenbanken muss berücksichtigt werden. Dazu müssen zunächst geeignete Kriterien identifiziert werden, die unabhängig von der Art der Datenspeicherung angewendet werden können. Hierzu gibt es bereits Ansätze in der Literatur [171, 168, 85, 130, 123], die sich aber vornehmlich mit der Evaluation von relationalen Datenbanksystemen beschäftigen. Die Evaluation von XML-Schemata wurde bisher nur von wenigen Arbeiten untersucht, wie z.B. von Lammel und Visser [143, 248]. Die bisherigen Kriterien und besonders ihre Implementierungen sind allerdings nicht generisch und können deshalb nur eingeschränkt auf Datenspeichersysteme im Allgemeinen angewendet werden. So wird häufig die Anzahl der Entitäten [168, 85] oder auch die Anzahl bestimmter Relationen (z.B. 1:n Beziehungen [85]) als Maß verwendet. Bei der Analyse der Vollständigkeit eine Datenmodells wird bei Moody [168] auf die Befragung der Nutzer und Manager des Systems verwiesen. Dies ist in einem offenem System aufgrund der flexiblen Anzahl der Systemteilnehmer jedoch nicht möglich. Eine umfassende Übersicht über Arbeiten, welche sich mit der Evaluation von Datenstrukturen auseinandersetzen findet sich in Kapitel 3.

Darüber hinaus sind die Kriterien nicht unabhängig voneinander. Bereits in [171] wurde dargestellt, dass die Verbesserung eines Kriterium häufig zwangsläufig nur durch die Verschlechterung eines anderen Kriteriums erreicht werden kann. Somit ist bei der Aufstellung des Kriterienkatalogs nicht nur zu beachten, welche Kriterien in diesen aufgenommen werden sollen, sondern auch eine Hierarchie zwischen den Kriterien zu beachten.

Dementsprechend ergibt sich für den Teilbereich von Datenstandards folgende Aufgabenstellung:

1. Vorstellung von Arbeiten zur Bewertung von Datenstandards (Kapitel 3)

2. Entwicklung eines generischen Evaluationsframeworks für Datenstandards (Kapitel 4)

3. Entwicklung eines Datenstandards für die Biodiversitätsinformatik (Kapitel 5)

1.2.2 Infrastrukturen

Eine Infrastruktur besteht aus mehreren Einzelkomponenten zum Speichern und Austausch von Daten, wie z.B. Schnittstellen, welche den Übergang von einem Datenspeicher in einen anderen Datenspeicher ermöglichen. Anders als bei der Analyse eines Datenstandards muss deshalb das Zusammenspiel der Einzelkomponenten, sowie dessen Auswirkung auf die Gesamtarchitektur bewertet werden. Darüber hinaus ist in einem offenen System häufig mehr als ein Weg verfügbar, über den Daten ausgetauscht werden können, da theoretisch jeder Datenspeicher des Systems mit beliebigen anderen Datenspeichern des Systems Datenaustausch betreiben kann. Des Weiteren besteht eine Infrastruktur nicht nur aus ihren technischen Komponenten sondern verfügt auch über eine soziale und eine funktionale Struktur. Die soziale Struktur ist dafür verantwortlich, wie neue Mitglieder eingebunden werden und wie der einzelne Nutzer innerhalb einer Infrastruktur unterstützt wird. In der funktionalen Struktur ist spezifiziert, welche Datenstandards unterstützt werden, über welche Prozesse Daten übertragen werden und wie die Integrität der Daten gewährleistet wird.

Vor diesem Hintergrund stellen sich folgende Fragen:

1. Was sind die Komponenten einer Dateninfrastruktur?
2. Wie kann der Datenaustausch zwischen zwei Einzelkomponenten bewertet werden?
3. Welche Daten können übertragen werden?
4. Welche Daten gehen bei der Übertragung verloren?
5. Wie können Duplikate identifiziert werden?
6. Wie wird ein neuer Teilnehmer in eine Infrastruktur eingebunden?

Um eine Infrastruktur analysieren zu können, ist es zunächst erforderlich, die wesentlichen Merkmale von Infrastrukturen und technologischen Ansätze vorzustellen (siehe Kapitel 6). Da hierbei das Thema der Datenintegration von zentraler Bedeutung ist, werden in diesem Kapitel die Grundlagen der Datenintegration vermittelt.

Ein wesentliches Element von aktuellen Infrastrukturen in der Biodiversitätsinformatik sind Wrapper, welche lokale Repositorien über Schnittstellen miteinander verbinden und Anpassungen des Datenaustauschformats an die lokalen Schemata

ermöglichen. Die Distribution der Daten findet im Allgemeinen über Portale statt, über welche die Daten nach einer Websuche in verschiedenen Formaten heruntergeladen werden können. Die Analyse einer Dateninfrastruktur hat die Aufgabe, alle Einzelkomponenten dieser Infrastruktur zu identifizieren und zu klassifizieren und das Zusammenspiel dieser Komponenten zu bewerten.

Dabei ist von besonderem Interesse, wie stark sich Datenverluste von spezifischen Einzelkomponenten auf die Gesamtqualität der Infrastruktur auswirken. Bei der Gesamtanalyse der Infrastruktur ist es von Bedeutung, ob neben Datenverlusten Artefakte – wie Duplikate – oder ein Bedeutungswandel der Daten auftritt. Dies hat gravierende Folgen für die Datennutzung in Forschungsprojekten. Durch Duplikate werden Auswertungen zu Artvorkommen verfälscht und durch einen Bedeutungswandel der Daten können diese bei einer sekundären Auswertung der Daten nicht mehr richtig interpretiert werden.

Grundlage für die Lösung dieser Probleme ist die Evaluation von existierenden Infrastrukturen. Hierzu wird für die generische Evaluation das 'Infrastructure Evaluation Framework' (IEF) neu entwickelt. Dieses stellt einen Baukasten an Kriterien zur Verfügung. Von diesen Kriterien werden die relevanten Kriterien in der domänenspezifischen Anpassung von IEF ausgewählt. Dadurch wird ein domänenspezifische Evaluationsframework für Infrastrukturen entwickelt. Für die Biodiversitätsinformatik wird dieses als IEF-Biodiv bezeichnet.

Damit ergeben sich für den Bereich der Infrastrukturen folgende Aufgabenstellung:

1. Vorstellung der Grundlagen der Datenintegration (Kapitel 6)
2. Entwicklung des generischen Evaluationsframeworks IEF und des domänenspezifischen Frameworks IEF-Biodiv zur Evaluation von existierenden Infrastrukturen in der Biodiversitätsinformatik (Kapitel 7)
3. Aufbau einer Infrastruktur für die Biodiversitätsinformatik (Kapitel 8)

1.3 Beispiel

Die Mängel beim Datenaustausch in Infrastrukturen in der Biodiversitätsinformatik werden im folgenden Abschnitt an einem Beispiel verdeutlicht. Hierfür wird aufgrund der herausragenden Bedeutung für die Biodiversitätsinformatik die Infrastruktur von GBIF (siehe Abschnitt 7.4.1) in Kombination mit der Infrastruktur des IBF-Projekts verwendet.

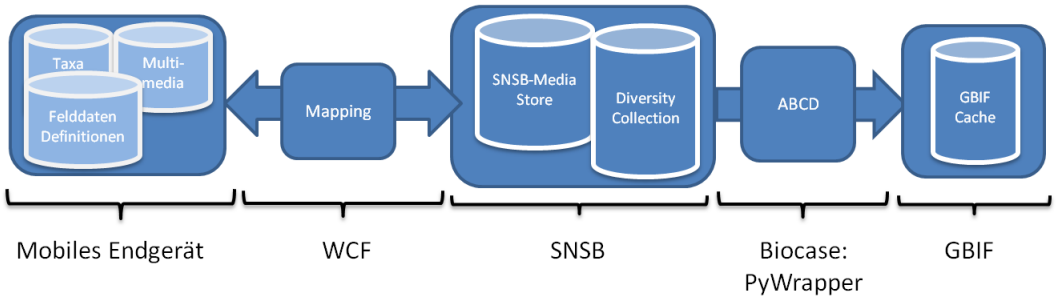

Abbildung 1.1: Datenfluss im IBF-Projekt

Das primäre Ziele des IBF-Projektes ist die Speicherung von Forschungsdaten in einem zentralen Repositorium, um diese über ein Portal allgemein verfügbar zu machen [116]. Als primärer Speicherort der Langzeitarchivierung steht das Repositorium des SNSB-IT-Centers zur Verfügung. Von dort werden die Daten in die GBIF-Infrastruktur (siehe Abschnitt 7.4.1) exportiert und über das GBIF-Portal publiziert. Dementsprechend ergibt sich von der Datenerhebung im Feld bis zum Datenabruf über das GBIF-Portals folgende Infrastruktur (siehe Abbildung 1.1):

1. Datenaufnahme im Feld mit DiversityMobile auf einem Mobiltelefon und lokale Speicherung auf einer Datenbank im Mobilgerät

2. Datenübertragung an das SNSB-IT-Center in das Datenformat von DiversityCollection

3. Datenspeicherung am SNSB-IT-Center mit der Möglichkeit der Datennachbearbeitung über DiversityCollection

4. Export der Daten an die GBIF-Datenbank im ABCD-Format über den PyWrapper

5. Distribution der Daten über das GBIF-Portal in einem wählbaren Ausgabeformat (DwC, Excel, csv)

Bei der Verwendung dieser Infrastruktur (umfangreiche Vorstellung siehe Abschnitt 7.4.2) werden die Daten in mindestens vier verschiedenen Datenformaten gespeichert und müssen somit mehrfach konvertiert werden. Jeder dieser Übergänge ist eine potentielle Quelle von Fehlern und damit maßgeblich für die Qualität der Gesamtarchitektur. Die erste Fehlerquelle findet sich bereits bei der primären Datenerhebung für ein Forschungsprojekt im Feld. Bereits an dieser Stelle besteht die Notwendigkeit, dass das Datenmodell von DiversityMobile die Speicherung der

projektrelevanten Forschungsfragen vollständig ermöglicht. Kann auch nur eine Aussage im Bezug auf das Forschungsprojekt nicht abgespeichert werden, dann tritt bereits an dieser Stelle ein Informationsverlust auf. Die nächste kritische Stelle ist die Datenübertragung vom Mobilgerät an das SNSB-IT-Center. Das auf dem Mobilgerät verwendete Datenschema ist weitgehend mit DiversityCollection kompatibel. Allerdings existiert bereits bei dieser Übertragung ein gewisser Anpassungsbedarf, da einige Datentypen, die in der Datenbank des SNSB-IT-Center verwendet werden (z.B. zur Kodierung von Geodaten), auf der Datenbank des Mobilgeräts nicht zur Verfügung stehen. Anschließend können die Daten am Repositorium des SNSB-IT-Centers dauerhaft gespeichert werden.

Eine weitere Datentransformation muss bei dem Datenexport vom SNSB-IT-Center an GBIF vorgenommen werden. Die Datenübertragung an sich unterscheidet sich zwar auf theoretischer Ebene nicht von der vorhergehenden Datenübertragung, ist aber in der Praxis ungleich komplexer, da das Datenschema von DiversityCollection vor einem anderen Hintergrund entwickelt wurde als das ABCD-Schema und DwC, welche von der GBIF-Infrastruktur verwendet werden. Deshalb sind diese Datenschemata sehr unterschiedlich. Insbesondere findet an dieser Stelle eine Technologieübergang statt, da die Daten von einem relationalen Datenbankmodell in ein XML-basiertes Schema übertragen werden müssen. Da der Nutzer über das GBIF-Portal ein Output-Format spezifizieren kann und ABCD nicht als Output angeboten wird, müssen die Daten innerhalb der GBIF Infrastruktur ein weiteres Mal transformiert werden.

Das heißt, dass bereits in diesem einfachen Anwendungsszenario die Daten bereits mindestens dreimal transformiert werden. Sobald eine Anfrage über mehrere GBIF-Knoten hinweg ausgeführt wird, entsteht zusätzlich die Schwierigkeit, dass die Daten aus unterschiedlichen Quellen stammen und aus unterschiedlichen Formaten in das ABCD-Format übertragen werden. Dabei können schemaspezifische Übertragungsfehler durch den PyWrapper entstehen. Die Daten werden aber anschließend als gleichartig behandelt.

Die Inkompatibilität zwischen den Datenmodellen von GBIF und DiversityCollection führt dabei zu erheblichen Datenverlusten, wie anhand der Abbildungen 1.2 bis 1.4 erläutert wird. Grundlage der Abbildungen ist folgender Anwendungsfall, welcher im IBF-Projekt [106] im Teilprojekt IBFLichens aufgetreten ist. Eine Studie mit ähnlichem Anwendungshintergrund wurde mit [218] für IBFFungi veröffentlicht.

Im Rahmen einer Geländebegehung soll erhoben werden, wie Pflanzen, Pilze und Flechten zusammenleben. Zu diesem Zweck sollen biologische Objekte im Gelände

kartiert und identifiziert werden. Neben dem Standort der einzelnen Objekte ist dabei von Interesse, ob bestimmte Arten gemeinsam an einem Standort auftreten und in welcher Form diese zusammenleben (z.B. Symbiose zwischen Pilzen und Pflanzen).

Abbildung 1.2 zeigt einen typischen Anwendungsfall der Dokumentation des Zusammenlebens zwischen einer Pflanze ('Quercus Robur L.'), einem Pilz ('Athelia arachnoidea (Berk.) Jülich') und einer Flechte('Xanthoria parietina L.') im Format von DiversityCollection. Die Visualisierung macht deutlich, dass Pflanze, Flechte und Pilz zueinander in Beziehung stehen. Intern ist auch die Form des Zusammenlebens (z.B. Symbiose zwischen Pilz und Pflanze) dokumentiert.

Die Datenübertragung an das SNSB-IT-Center ist problemlos möglich, da das Datenmodell von DiversityCollection im Hinblick auf diese Projekte konzipiert wurde. Die Daten können in die Infrastruktur von GBIF übertragen und über das GBIF-Portal publiziert werden (Abbildung 1.3) Die Datenübertragung an GBIF ist allerdings mit erheblichen Datenverlusten verbunden, wie ein Ausschnitt desselben Datensatzes nach Konvertierung der Originaldaten über das ABCD-Schema in DwC zeigt (Abbildung 1.4). Die Kartierungsobjekte werden unabhängig voneinander als Einzelbelege präsentiert. Die Flechten wurden erst gar nicht an GBIF übertragen.

Damit sind innerhalb der gesamten Infrastruktur nicht nur die Informationen zum Zusammenleben der Kartierungsobjekte verloren gegangen, sondern sogar zwei Objekte selbst. Die Ursache dieses Informationsverlusts kann ohne eingehende Analyse der Infrastruktur nicht identifiziert werden. Als mögliche Ursachen kommen in Frage:

- Das ABCD-Schema kann nicht alle Felder aus DiversityCollection abbilden
- Der PyWrapper kann die Daten nicht richtig zuordnen, obwohl korrespondierende Datenfelder vorhanden sind
- Der PyWrapper wurde nicht richtig konfiguriert
- Bei der Transformation von ABCD nach DwC treten innerhalb der GBIF-Infrastruktur treten Verluste auf

Der erste Punkt korreliert direkt zu der Fragestellung, was ein Datenspeichersystem inhaltlich erfassen kann, während die weiteren Punkte relevant bezüglich der Qualität der Infrastruktur sind. Die genaue Ursache ist aber ohne eingehende Evaluation der verwendeten Datenstandards und der Infrastruktur nicht möglich.

Für eine wissenschaftliche Nutzung persistent gespeicherter Daten ist eine gute Datenqualität eine unentbehrliche Voraussetzung. Es ist deshalb von entscheidender Bedeutung, die so gezeigten Schwachstellen einer Infrastruktur aufzudecken, zu

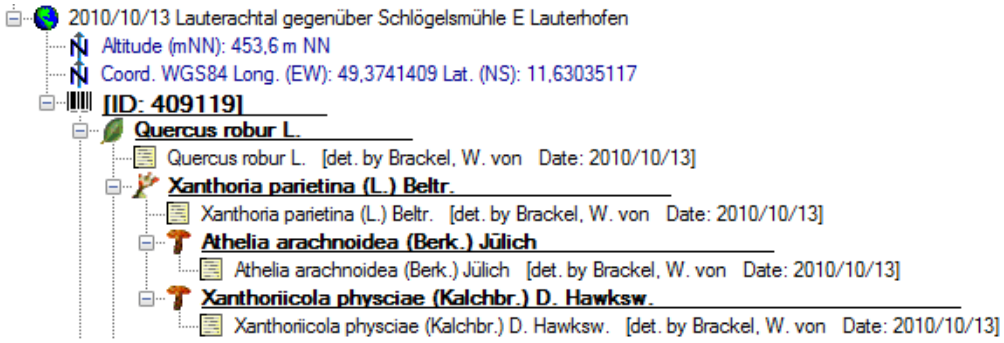

Abbildung 1.2: Daten von IBFLichens in DiversityCollection

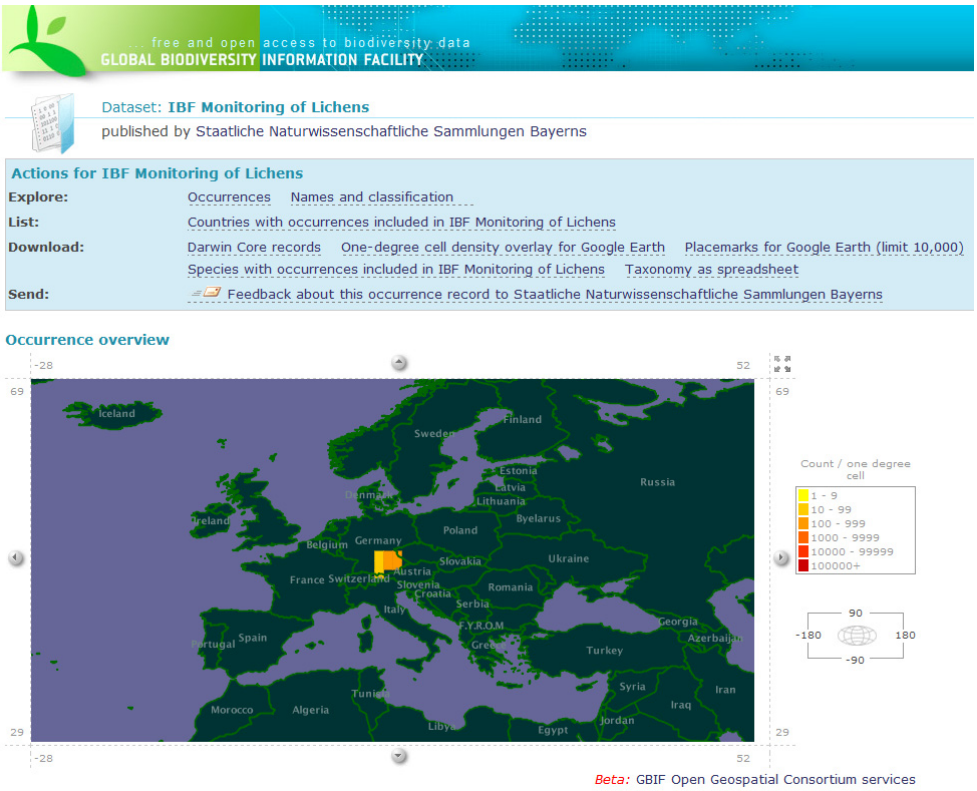

Abbildung 1.3: Sammlung IBFLichens bei GBIF

Quercus robur L. (Catalogue number: 409119-502734)

Occurrence key in GBIF portal: 367135255
Occurrence page in GBIF portal: http://data.gbif.org/occurrences/367135255
Web service request for data for occurrence: http://data.gbif.org/ws/rest/occurrence/get/367135255
Taxon key in GBIF portal: 24947321
Taxon page in GBIF portal: http://data.gbif.org/species/24947321
Web service request for taxon: http://data.gbif.org/ws/rest/taxon/get/24947321
catalogNumber: 409119-502734
collectionCode: IBFlichenscoll
collector: von Brackel, Wolfgang (Hemhofen)
country: Germany
decimalLatitude: 49.3741416931
decimalLongitude: 11.6303510666
institutionCode: REG
earliestDateCollected: 2010-10-13
latestDateCollected: 2010-10-13
Identified as: Quercus robur L.
Identification higher taxa: Offenboden
locality: Lauterachtal gegenüber Schlögelsmühle E Lauterhofen
gbifNotes: Data from GBIF data index - original values.

Xanthoria parietina (L.) Beltr. (Catalogue number: 409120-502824)

Occurrence key in GBIF portal: 367135521
Occurrence page in GBIF portal: http://data.gbif.org/occurrences/367135521
Web service request for data for occurrence: http://data.gbif.org/ws/rest/occurrence/get/367135521
Taxon key in GBIF portal: 24947363
Taxon page in GBIF portal: http://data.gbif.org/species/24947363
Web service request for taxon: http://data.gbif.org/ws/rest/taxon/get/24947363
catalogNumber: 409120-502824
collectionCode: IBFlichenscoll
collector: von Brackel, Wolfgang (Hemhofen)
country: Germany
decimalLatitude: 49.3728065491
decimalLongitude: 11.6430740356
institutionCode: REG
earliestDateCollected: 2010-10-13
latestDateCollected: 2010-10-13
Identified as: Xanthoria parietina (L.) Beltr.
Identification higher taxa: Totholz
locality: Lauterachtal zwischen Schlögelsmühle und Pattershofen W Kastl
gbifNotes: Data from GBIF data index - original values.

Abbildung 1.4: Daten von IBFLichens bei GBIF im DarwinCore Format

protokollieren und zu eliminieren. Dafür existieren aber aktuell in der Biodiversitätsinformatik noch keine befriedigenden Lösungen.

1.4 Lösungsweg und Aufbau

Die vorliegende Arbeit beginnt in Kapitel 2 mit einer umfassenden Analyse der Anwendungsdomäne. Dazu werden zunächst die Disziplinen der Biodiversitätsforschung und der Biodiversitätsinformatik vorgestellt und voneinander abgegrenzt. Neben der Einführung notwendiger Fachtermini wird auch die Notwendigkeit der Biodiversitätsinformatik dargestellt und die wichtigsten Organisationen, Projekte und Netzwerke kurz beschrieben.

Anschließend gliedert sich die Arbeit nach den beiden großen Themenkomplexen der Datenstandards und Infrastrukturen in zwei Hauptteile. Diese sind analog nach folgender Struktur aufgebaut:

1. Grundlagen, verwandte Ansätze und Arbeiten
2. Entwicklung eines Evaluationssystems und Anwendung auf existierende Strukturen
3. Vorstellung des verbesserten Systems

Der erste Hauptteil beschäftigt sich mit der Evaluation von Datenstandards, da eine geeignete Methode zur Evaluation von Datenstandards eine unverzichtbare Grundvoraussetzung bei der Evaluation von Infrastrukturen ist. Dazu werden zunächst bereits existierende Ansätze zur Bewertung von Schemata untersucht und mit Hilfe dieser die wichtigsten Kriterien zur Evaluation erarbeitet. Dabei sind relevante Ansätze nicht nur im Umfeld der Bewertung von konzeptuellen Schemata wie z.B. in [85, 123, 130, 168], sondern auch bei der Bewertung von XML Schemata [143, 248] und dem Requirementsengineering zu finden. Es wird gezeigt, dass die Implementierung der Kriterien in vorhandenen Evaluationssystemen insbesondere im Bezug auf das Kriterium der Vollständigkeit den Anforderungen eines Evaluationssystems für die Biodiversitätsinformatik nicht genügt und somit ein eigenes Evaluationssystem entwickelt werden muss (Kapitel 3).

Hierzu wird als Grundlage eine prozessorientierte Sichtweise gewählt und die 'prozessorientierte Schemaevaluation' (POSE) auf Basis der 'perspektiven-orientierten Prozessmodellierung' (POPM) als neues Evaluationssystem entwickelt. Dieses wird auf etablierte Standards der Biodiversitätsinformatik angewendet (Kapitel 4). Die Evaluation ergibt, dass die bisherigen Datenstandards den Anforderungen aus der

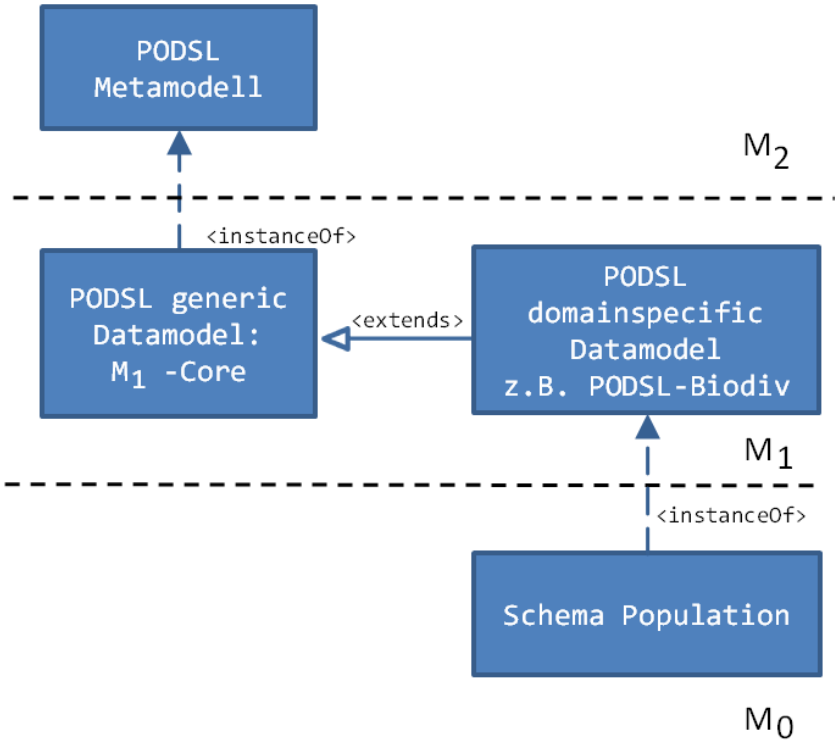

Abbildung 1.5: Metastruktur von PODSL

Praxis nicht genügen und somit ein erheblicher Bedarf an einem neuen, flexibleren Datenstandard besteht. Um die Anforderungen bezüglich der Flexibilität und der Vollständigkeit zu erfüllen, wird in Kapitel 5 mit der 'Process Oriented Data Schema Language' (PODSL) ein neues Datenschema vorgestellt, welche auf den Prinzipien der Metamodellierung mit OMME aus [250] beruht. Auf Basis des Metamodells von PODSL wird eine generisches Datenmodell entwickelt, das domänenspezifisch erweiterbar ist. Auf dieser Basis wird mit PODSL-Biodiv ein Datenstandards geschaffen, der speziell auf die Anforderungen der Biodiversitätsinformatik zugeschnitten ist. Die Metastruktur von PODSL ist in Abbildung 1.5 dargestellt.

Der zweite Hauptteil beschäftigt sich mit der Evaluation von Infrastrukturen. Dazu werden in Kapitel 6 die Grundlagen von Infrastrukturen und der Datenintegration, sowie existierende Ansätze vorgestellt. Da in einer Infrastruktur mehrere Komponenten miteinander interagieren, wird deutlich, dass innerhalb von Infrastrukturen eine prozessorientierte Sichtweise von erheblichem Nutzen ist. Somit wird mit DaltON [117, 118, 40, 201] ein System zum Datenaustausch vorgestellt, in welchem eine prozessorientierte Sichtweise vorherrscht.

Da aktuelle keine Methode zur Evaluation von Infrastrukturen existiert, wird anschließend mit dem 'Infrastructure Evaluation Framework' (IEF) ein Evaluations-

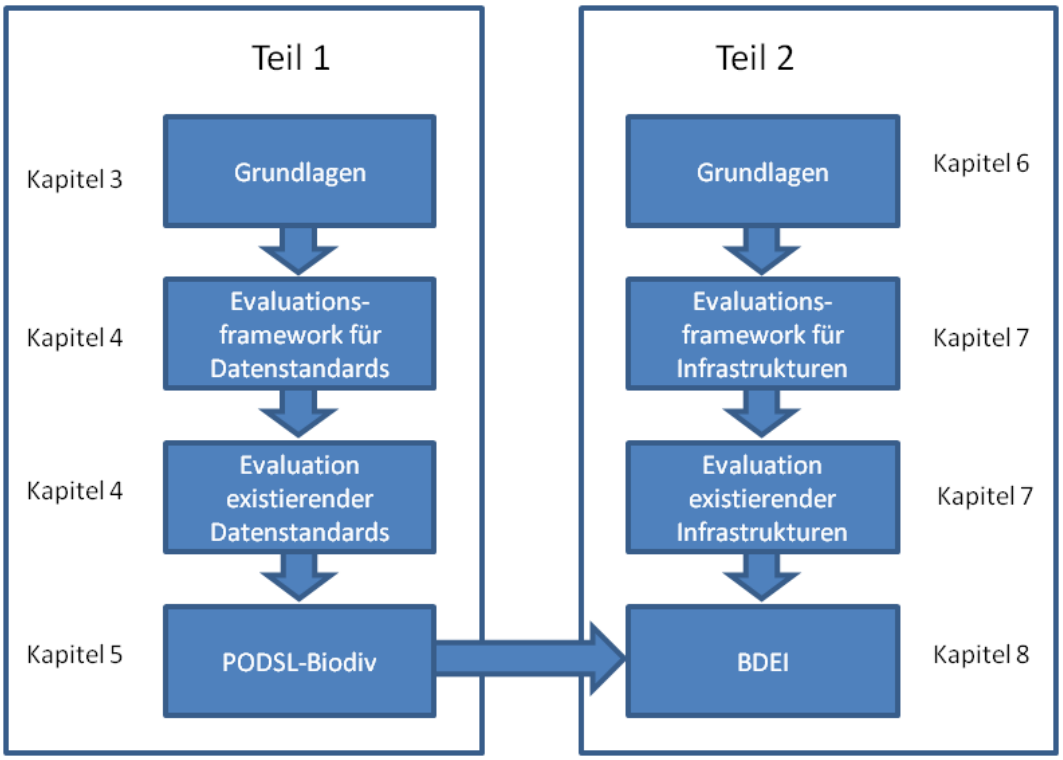

Abbildung 1.6: Gesamtkonzept der Arbeit

framework für Infrastrukturen auf Basis der Perspektiven von POPM neu entwickelt und auf ausgewählte Infrastrukturen der Biodiversitätsinformatik angewendet (Kapitel 7). Es zeigt sich, dass die Infrastruktur von GBIF in vielen relevanten Merkmalen einer Infrastruktur überzeugen kann, aber erheblicher Verbesserungen bedarf. Hierfür wird mit der 'Biodiversity Data Exchange Infrastructure' (BDEI) in Kapitel 8 eine Erweiterung der GBIF-Infrastruktur als neue Entwicklung dieser Arbeit vorgestellt.

Das Konzept dieser Arbeit mit der Unterteilung in die beiden Hauptteile Datenstandards und Infrastrukturen ist in Abbildung 1.6 dargestellt. Dabei wird in beiden Teilen mit PODSL-Biodiv und BDEI eine Lösung für die Anforderungen der Biodiversitätsinformatik vorgestellt. Beide Teile sind zunächst unabhängig voneinander strukturiert. Bei der Konzeption von BDEI wird mit PODSL-Biodiv das Ergebnis des ersten Teils als moderierendes Schema für den Datenaustausch verwendet.

Abschließend wird in Kapitel 9 mögliche Weiterentwicklungen und praktische Einsatzmöglichkeiten dieser Arbeit diskutiert. Mit einer einheitlichen Sicht auf Biodiversitätsdaten ist insbesondere die Auswertung der Daten mit Datamining möglich.

1.5 Beitrag dieser Arbeit

Diese Arbeit hat die Lösung der Probleme P1 bis P4 aus Abschnitt 1.2 zum Ziel. Die Lösung dieser Probleme wird durch folgende eigene Entwicklungen realisiert:

L1 Erstellung eines Kriterienkatalogs zur Evaluation von Datenstandards aus den bisherigen Entwürfen von wissenschaftlichen Arbeiten zur Evaluation konzeptueller Schemata

L2 Entwicklung von POSE zur Evaluation von Datenstandards auf Basis dieser Kriterien

L3 Entwicklung von PODSL-Biodiv als erweiterbares Modell zur Datenspeicherung in der Biodiversitätsinformatik

L4 Entwicklung von IEF und IEF-Biodiv zur Evaluation von Infrastrukturen

L5 Evaluation von Infrastrukturen der Biodiversitätsinformatik und verwandter Wissenschaften

L6 Entwicklung von BDEI als Infrastruktur für den Datenaustausch in der Biodiversitätsinformatik auf Grundlage von GBIF

Der Beitrag von L1 und L2 bezieht sich direkt auf das Problem P1. Die Lösung dieses Problems bildet sowohl die Grundlage bei der Entwicklung eines Standards für die Biodiversitätsinformatik, als auch ein wertvolles Werkzeug zur Evaluation von Infrastrukturen, da in dieser die Evaluation der verwendeten Datenstandards als Teilproblem auftritt.

Die Lösung von P2 liegt darin, ein flexibles Datenschema mit Hilfe der Metamodellierung für die Domäne der Biodiversitätsinformatik zu entwickeln und dieses bei Bedarf zu erweitern. Dies ist ein neuartiger Ansatz, der so im Bereich der Biodiversitätsinformatik (außer in Ansätzen bei [149]) noch nicht verwendet wurde. Gegenüber der bisherigen Praxis der Erweiterung von Standards hat diese Methode den Vorteil durch das Metamodell einen Rahmen zu schaffen, welcher die eindeutige Interpretation spezifischer Erweiterungen ermöglicht. Der eigene Beitrag hierzu ist L3 mit der Entwicklung von PODSL-Biodiv.

In der Biodiversitätsinformatik stellen die Veränderungen und die Verluste von Daten in Infrastrukturen – wie im Beispiel zu Kapitel 1.3 gezeigt werden konnte – ein erhebliches Problem dar. Aus akademischer Sicht ist die Lösung dieses Problems von besonderem Interesse, da es aktuell keine oder kaum Arbeiten gibt, welche sich mit durchaus praxisrelevanten Fragestellung der Evaluation von Infrastrukturen

auseinandersetzen. Infrastrukturen werden aus Sicht der Literatur häufig mit der Infrastruktur eines Unternehmens mit einer bekannten Teilnehmergruppe betrachtet (vgl. [194]). Die Betrachtung von Infrastrukturen mit einem offenen Teilnehmerkreis ist weitgehend unerforscht. Die Lösung von P3 durch L4 stellt damit einen erheblichen Schritt zum Verständnis von offenen Infrastrukturen dar und ist nicht nur für die Biodiversitätsinformatik von entscheidender Bedeutung, da sie auch auf ähnliche Architekturen angewendet werden kann. Durch die Anwendung von IEF-Biodiv auf etablierte Infrastrukturen wird deutlich, dass keine diese Infrastrukturen in der jetzigen Form alle Anforderungen der Biodiversitätsinformatik erfüllen kann (L5).

Abschließend werden zur Lösung von P4 alle Erkenntnisse dieser Arbeit genutzt. Zum einen wird in L4 erkannt, dass ein flexibles Datenmodell ein elementarer Bestandteil einer Infrastruktur in der Biodiversitätsinformatik ist. Zum anderen werden in L5 die Schwachpunkte existierender Infrastrukturen deutlich gemacht. Dementsprechend wird mit der 'Biodiversity Data Exchange Infrastructure' (BDEI) in L6 eine Infrastruktur vorgestellt, welche PODSL-Biodiv als Austauschformat verwendet. Durch die Flexibilität von PODSL-Biodiv lässt sich der Datenaustausch in BDEI gut an heterogene Anforderungen anpassen. Da die von der 'Global Biodiversity Information Facility' (GBIF) Netzwerk bereits wichtige Kriterien der Evaluation aus L5 erfüllt, das BDEI auf Grundlage des GBIF Netzwerks entwickelt. Allerdings kann in L5 gezeigt werden, dass das GBIF Netzwerk erhebliche Mängel im Bezug auf die Funktionalität und Datenverluste aufweist. BDEI greift auf die Infrastruktur des GBIF Netzwerks zurück und verbessert diese Infrastruktur in Bezug auf die Verlustfreiheit und Flexibilität der Infrastruktur, so dass P4 gelöst wird.

Kapitel 2

Die Anwendungsdomäne

Ziel des folgenden Kapitels ist die Vorstellung der Biodiversitätsforschung und der Biodiversitätsinformatik aus der Sicht eines Informatikers. Die Biodiversitätsforschung ist die Anwendungsdomäne der Biodiversitätsinformatik, da die Biodiversitätsforschung die zu Grunde liegenden Projekte und Probleme definiert. Aus der Sicht eines Wissenschaftlers im Biodiversitätsbereich ist zunächst seine Anwendungsdomäne interessant – die Datenspeicherung und der Datenaustausch als klassische Aufgaben der Biodiversitätsinformatik sind hingegen nur von sekundärem Interesse. Deshalb ist es als Informatiker wichtig, die grundlegenden Probleme der Biodiversitätforschung soweit zu verstehen, wie sie zur Behandlung von Problemen innerhalb der Biodiversitätsinformatik erforderlich sind. Diese wird in Abschnitt 2.1 vorgestellt. Dabei ist anzumerken, dass die vorgestellten Probleme wie z.B. das Artenproblem [48] und weitere terminologische Fragestellungen [226, 227, 265] innerhalb der Biodiversitätsforschung nicht abschließend geklärt werden konnten. Die Standpunkte dieser offenen Fragestellungen können innerhalb dieser Arbeit nicht vertieft werden, so dass der interessierte Leser auf einschlägige Fachliteratur – wie sie auch in der Bibliografie verwendet wird – verwiesen werden muss.

Anschließend wird die Biodiversitätsinformatik in Abschnitt 2.2 als eigenständige Disziplin eingeführt und es werden die wichtigsten Begriffe dieses Bereichs definiert. Das Verständnis für die Probleme dieser Domäne ist Grundvoraussetzung für die Erstellung von PODSL-Biodiv als Datenstandard für die Biodiversitätsinformatik (Kapitel 5) sowie BDEI als Infrastruktur(Kapitel 8). Um die Arbeitsweise in Projekten zu illustrieren, wird diese für den Informatiker in Form von Prozessmodellen zugänglich gemacht. Dazu werden Prozesse, wie sie im Rahmen des IBF-Projektes [116, 52] aufgetreten sind, mit Hilfe des 'Perspective Oriented Process Modellings' [114] veranschaulicht.

In Abschnitt 2.3 werden die wichtigsten Akteure und Strukturen im Bereich der Biodiversitätsinformatik vorgestellt. Dies sind internationale und nationale Organisationen, welche innerhalb der Biodiversitätsinformatik von entscheidender Bedeutung sind – wie z.B. die TDWG [232] oder GBIF [82]. Darüber hinaus wird ein kurzer Überblick über die Projektarbeit im Umfeld der Biodiversitätsinformatik gegeben (Abschnitt 2.3.2). Da auch hier die Zahl der Projekte sehr umfassend ist, wird insbesondere auf wichtige europäische und deutsche Projekte eingegangen. Von besonderer Bedeutung ist dabei das IBF-Projekt, da an Hand dessen die grundlegenden Fragestellungen dieser Arbeit identifiziert und getestet werden konnten.

Der nächste behandelte Aspekt ist die Bereitstellung von Daten über Portale (Abschnitt 2.3.3) aus Biodiversitätsdatenbanken. Bei der Auswahl der Portale ist dabei die Bedeutung dieser Portale als alleiniges Kriterium nicht ausreichend, da insbesondere auch in kleineren Portalen eine Vielzahl von Datensätzen gespeichert wird. Diese sind durch einen starken Projektbezug und die Verwendung von proprietären Schemata als Beispiele für die Schwierigkeiten im Bereich der Datenintegration und des Datenaustauschs von Bedeutung.

Abschließend wird ein Überblick über die wichtigsten Standards und ihrem jeweiligen Anwendungsbereich gegeben. Im Rahmen dessen wird auch auf mögliche Erweiterungen und die Organisation hinter der Standardisierung eingegangen. Die Kenntnis dieser Standards ist insbesondere zum Verständnis der Diskussion in Kapitel 4, in dem eine Evaluation von ausgewählten Standards ausgeführt wird.

2.1 Biodiversitätsforschung

Die Biodiversitätsforschung nimmt aktuell durch das Jahr der Biodiversität 2010 [245] und die Dekade der Biodiversität 2010-2020 [246] der Vereinten Nationen einen hohen Stellenwert ein (siehe Abbildung 2.1)[1]. Durch diese beiden Großereignisse wird die zunehmende, internationale Bedeutung der Biodiversitätsforschung, welche mit der Convention on Biological Diversity (CBD) [28] 1992 auf der 'United Nations Conference on Environment and Development' in Rio de Janeiro eingeleitet wurde, unterstrichen.

[1] Das CBD-Sekretariat hat das offizielle Logo zum Jahrzehnt der Biodiversität zur Verfügung gestellt und die Genehmigung zur Veröffentlichung in dieser Arbeit erteilt.

Abbildung 2.1: Offizielles Logo zum Jahrzehnt der Biodiversität

Die CBD wird dabei zu den bedeutendsten, internationalen Abkommen im Umweltbereich [102, 153] gezählt und wurde bisher von 193 Staaten inklusive Deutschland [29] ratifiziert. Dabei verpflichtet die CBD in Artikel 1 die ratifizierenden Staaten zur Einhaltung folgender, gleichrangiger Ziele [28]:

- Erhaltung der biologischen Vielfalt
- Nachhaltiger Nutzen ihrer Komponenten
- Gerechte Verteilung aus dem Nutzen genetischer Ressourcen und Austausch von genetischen Ressourcen und relevanten Technologien

Die Umsetzung der CBD wurde durch bislang 11 Konferenzen der Teilnehmer ('Conference of Parties') in Angriff genommen und wurde 2000 durch das Protokolle von Cartagena und 2010 durch das Protokoll von Nagoya zusätzlich reguliert [29]. Die Konvention verpflichtet die ratifizierenden Staaten zur Entwicklung einer nationalen Strategie zur Umsetzung der oben genannten Ziele, wobei diese für Deutschland erst im Jahr 2007 durch die Bundesregierung beschlossen werden konnte [24]. Zusammenfassend kann man die CBD als zentralen Ausgangspunkt für die völkerrechtliche Verankerung der Biodiversitätswissenschaft und somit als Ausgangspunkt für eine Vielzahl von Initiativen und Projekten betrachten.

Gemäß der Präambel der CBD wurde ein Mangel an Kenntnissen über die Zusammenhänge in der Biodiversitätsforschung ersichtlich, der durch den Aufbau von wissenschaftlichen, technischen und institutionellen Kapazitäten beseitigt werden soll [28]. Als ein wichtiger Schritt in dieser Richtung kann das formal von der CBD unabhängige 'Global Biodiversity Assessment' (GBA) betrachtet werden, welches von dem 'United Nations Environmental Programme' (UNEP) initiiert wurde. Das GBA hatte die Aufgabe, eine unabhängige, kritische, expertengeprüfte, wissenschaftliche Analyse der wichtigsten globalen Aspekte der Biodiversität zu erstellen [102]. Das Ergebnis des Projektes ist eine umfassende Beschreibung des Standes der Biodiversität Mitte der 90er Jahre, welches in [102] publiziert wurde.

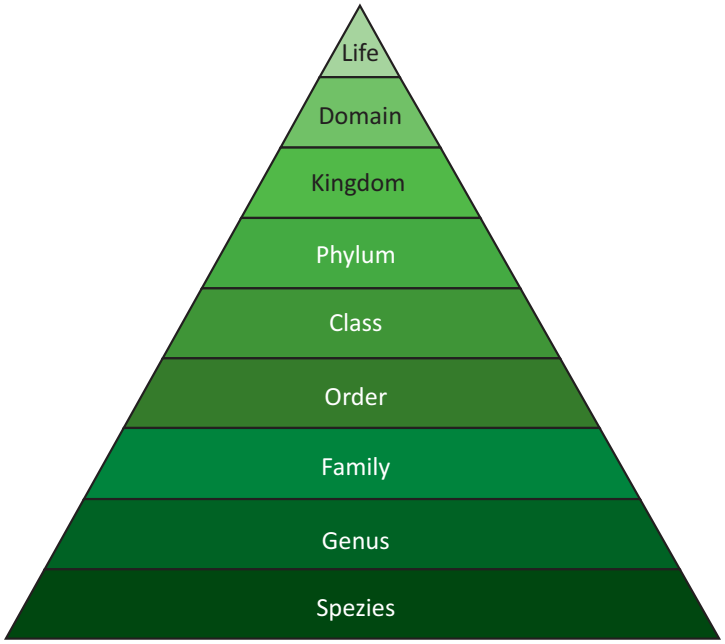

Abbildung 2.2: Hierarchie der biologischen Klassifikation mit acht taxonomischen Rängen in Einklang mit [266]

2.1.1 Definitionen

Es besteht Uneinigkeit über die genauen Definitionen von fundamentalen Begriffen wie 'Art' (engl. 'species') und 'Biodiversität' (engl. 'Biodiversity' beziehungsweise 'Biological Diversity'). Für die Definition des Biodiversitätsbegriffs ist der Begriff der Art von entscheidender Bedeutung, da dieser in den wichtigsten Definitionen der Biodiversität verwendet wird [119]. Die Definition des Artbegriffs ist umstritten. So wird aktuell von Biologen eine Vielzahl verschiedener Artkonzepte diskutiert und welche Kriterien dafür maßgeblich sind [226]. In der Wissenschaft wird je nach Zählweise aktuell zwischen bis zu 27 Konzepten unterschieden [265]. Hierbei sind verschiedene Systeme – wie zum Beispiel das in [266] vorgestellte System – gebräuchlich (Abbildung 2.2). Der taxonomische Rang der Art ist von besonderer Bedeutung, da dieser von Wissenschaftlern aus dem Bereich der Botanik [159] aber auch allgemein im Bereich der Biodiversitätsforschung [102] als taxonomische Grundeinheit betrachtet wird.

Grundlegend für die Probleme bei der Bestimmung des Artbegriffs ist das biologische Artenkonzept [226], welches auf Mayr [155, 156] zurückgeht.

Definition 2.1: Art nach Mayr [156]

Eine Art ist eine Gruppe von tatsächlich oder potentiell sich kreuzenden Populationen, welche im Bezug auf die Vermehrung von anderen solcher Gruppen isoliert sind.

Diese Definition ist im Laufe der Zeit stark kritisiert worden [54, 63, 223, 226]. Die Kritikpunkte beziehen sich insbesondere darauf, dass die Artdefinition nach Mayr [155, 156] in weiten Teilen der Biologie gar nicht anwendbar ist. So kann diese definitionsgemäß nicht auf Organismen mit asexueller Fortpflanzung angewendet werden [63]. Darüber hinaus ist die Forderung des Artkonzepts in der Praxis kaum zu überprüfen und insbesondere bei ausgestorbenen Formen nicht anwendbar [226]. Zudem existieren zahlreiche Ausnahmen [226]. So sind beispielsweise Löwen und Tiger miteinander kreuzbar aber in relevanten Eigenschaften verschieden [265]. Folglich haben sich mit der Zeit alternative Definitionen des Begriffs 'Art' je nach dem Anwendungsbereich der definierenden Wissenschaftler entwickelt. So werden in [21, 72] sieben verschiedene Definitionen zum Artbegriff gegeben, von welchen die morphologische Artdefinition in [72] in den meisten Anwendungsfällen als günstig betrachtet und deshalb hier vorgestellt wird. Für weitere Definitionen des Begriffs sei auf [72, 21] verwiesen.

Definition 2.2: Morphologischer Artbegriff

Arten sind die kleinsten natürlichen Populationen, welche durch klare Unterschiede erblicher Eigenschaften (Gestalt, Verhalten, biochemische Eigenschaften) dauerhaft voneinander unterschieden werden können.

Die so entstandene Vielzahl der möglichen Definition für 'Art' und die sich daraus ergebenden Komplikationen werden als das Artenproblem bezeichnet [48]. Entscheidend ist hierbei für den Biodiversitätsinformatiker, dass je nach Artdefinition ein Beobachtungsobjekt auch unterschiedlich taxonomisch eingeordnet werden muss. Das heißt auch, dass zwei biologische Individuen nach einer Artdefinition derselben Art angehören, wogegen sie nach Verwendung einer anderen Artdefinition verschiedenen Arten angehören. So wären nach der Definition von Mayr streng genommen Löwen und Tiger dieselbe Art, da sie sich miteinander kreuzen können. Bei der Verwendung des morphologischen Artbegriffs, würden sie allerdings verschiedenen Arten zugerechnet, da sie sich in erblichen Eigenschaften signifikant voneinander unterscheiden.

Darüber hinaus ist die Zuordnung zu einer Art über die Zeit hinweg nicht zwangsläufig konstant, da neue Entwicklungen wie die DNA-Analyse eine Neubewertung

von bestimmten Arten notwendig machen und auch taxonomische Artnamen sich im Laufe der Zeit ändern können. Zusätzlich werden taxonomische Bezeichnungen auch je nach wissenschaftlicher Gemeinschaft unterschiedlich verwendet. Dies geschieht im Allgemeinen mit einem regionalen Bezug. So wurde zu Beginn des 20. Jahrhunderts dieselbe Vogelart in den USA an der Ost- und an der Westküste unterschiedlich bezeichnet, so dass Statistiken über die Verbreitung von Vögeln in dieser Zeit verfälscht wurden[2]. Auch innerhalb Deutschlands sind aktuell verschiedene taxonomische Listen im Einsatz. So werden in Norddeutschland beispielsweise[3] andere taxonomische Listen zur Bestimmung von Blütenpflanzen verwendet als in Bayern.

Das Problem der verschiedenen taxonomischen Listen ist so schwerwiegend, dass dafür von der TDWG mit dem 'Taxonomic Concept Schema (TCS)' (vgl. Abschnitt 2.3.4) ein eigener Standard definiert wurde [232]. Für die Biodiversitätsinformatik ist dies ein erhebliches Problem, da alte Datenbestände kontinuierlich an die Entwicklungen im taxonomischen Bereich angepasst werden müssen. Als Quintessenz lässt sich für den Informatiker daraus ableiten, dass die Artbezeichnung alleine noch keine zuverlässige Auskunft über den identifizierten biologischen Organismus geben kann. Zu einer sicheren Identifizierung werden vielmehr zusätzlich noch Jahreszahl und die verwendete taxonomische Liste benötigt. Diese Problematik ist dahingehend praxisrelevant, dass bei einem Datenintegrationsprozess im Rahmen der Biodiversitätsinformatik nicht nur die Schemata der Datenstandards aufeinander abgebildet werden müssen, sondern auch die Daten je nach verwendeter Taxonomie angepasst werden müssen. Dies ist insbesondere vor dem Hintergrund problematisch, dass nicht immer die benötigten Zusatzinformationen wie die verwendete taxonomische Liste im Rahmen der Datenerhebung miterfasst werden.

Da unter dem Begriff 'Biodiversität' in den wichtigsten Definitionen (vgl. [28]) auch die Artenvielfalt inkludiert ist, tritt das Artenproblem indirekt bei der Definition des Biodiversitätsbegriffs auf. Abgesehen von dieser Problematik gibt es eine Vielzahl von Möglichkeiten, den Biodiversitätsbegriff zu verstehen und zu definieren. Dabei werden in der Praxis die Begriffe 'Biodiversity' und 'Biological Diversity' synonym verwendet [97]. Eine Auflistung von Definitionen und deren Verständnis namhafter Wissenschaftler aus dem Bereich der Biodiversitätsforschung findet sich in [227]. Im Gegensatz zur Definition des Artbegriffs lässt sich allerdings die Entwicklung des Biodiversitätsbegriffs historisch gut nachvollziehen. Darüber hinaus wird der Begriff der 'Biodiversität' in offiziellen Dokumenten wie der CBD [28] verankert. Dementsprechend liegt die Problematik bei der Definition des Biodiversitätsbegriffs

[2] Beispiel aus dem Vortrag zu Döring [203] von der TDWG 2009

[3] nach Auskunft von Wolfgang Ahlmer, Universität Regensburg

weniger in der Formulierung an sich, sondern wie diese grundlegende Definition interpretiert wird.

Der Begriff 'Biodiversität' wie er im heutigen Sinne verwendet wird, tritt erstmal in den 80er Jahren in wissenschaftlichen Publikationen auf [97, 226]. Die ursprüngliche Wortschöpfung kann allerdings auf [42] zurückgeführt werden. Im wissenschaftlichen Bereich wurde der Begriff populär durch den Tagungsband der Konferenz 'National Forum on Biodiversity', der den Titel 'Biodiversity' trug [97]. Eine weitgehend akzeptierte Definition von Biodiversität findet sich in der CBD [97, 72, 102]:

Definition 2.3: Biodiversität nach der CBD [28]

Biodiversität ist die Variabilität unter lebenden Organismen jeglicher Herkunft, darunter unter anderem Land-, Meeres- und sonstige aquatische Ökosysteme und die ökologischen Komplexe, zu denen sie gehören. Dies umfasst die Vielfalt innerhalb der Arten und zwischen den Arten und die Vielfalt der Ökosysteme.

Der Begriff 'Biodiversität' wird von Heywood in [102] folgendermaßen präzisiert:

Definition 2.4: Biodiversität nach Heywood [102]

Biodiversität ist die absolute Diversität und Veränderlichkeit aller Lebewesen und aller Systeme, denen diese angehören. Dies schließt den ganzen Bereich der Variation und Veränderlichkeit zwischen Systemen und Organismen auf der bioregionalen und landschaftlichen Ebene, sowie die Ökosystem- und Habitatebene – genauso wie die verschiedenen organismischen Ebenen bis zur Art, Population, Individuum und Genen innerhalb des Bereichs der Population mit ein. Darüber hinaus sind auch die strukturellen und funktionalen Beziehungen innerhalb und zwischen diesen Organisationsebenen inklusive der Handlungen von Menschen in der Definition enthalten.

Diese Definition wurde nach Gruppen unterteilt und so verschiedenen Ebenen zugeordnet [102] (vgl. Abbildung 2.3):

- Diversität der Ökosysteme
- Artendiversität
- Genetische Diversität

Die so gebildeten Ebenen und Gruppen sind, wie in Abbildung 2.3 zu erkennen ist, nicht voneinander unabhängig, sondern vielmehr ineinander verschachtelte Hierarchien [72].

Zusätzlich wird von [27] gefordert, die Diversität auf molekularer Ebene als zusätzliche Komponente einzuführen. Dementsprechend umfasst der Biodiversitätsbe-

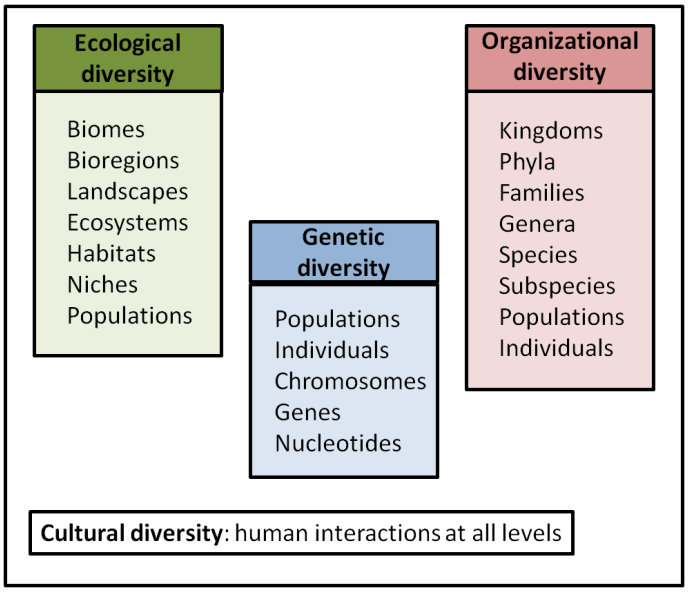

Abbildung 2.3: Ebenen der Biodiversität nach [102]

griff viel mehr als nur die Vielfalt der Arten, wobei in der Praxis dem Aspekt der Artenvielfalt immer noch besondere Aufmerksamkeit zuteil wird (vgl. [227]).

Zusammenfassend lässt sich festhalten, dass die Biodiversitätsforschung aus vielen verschiedenen Teildisziplinen besteht, die in den unterschiedlichsten Bereichen der Biologie und Geowissenschaften verankert sind. Dementsprechend ist der Charakter der Biodiversitätsforschung selbst so heterogen, dass kein einzelner Wissenschaftler den Gesamtbereich der Biodiversitätswissenschaft erforschen kann. Die in der Biodiversitätswissenschaft forschenden Wissenschaftler spezialisieren sich somit auf Teilgebiete, die vom Ansatz, von den Methoden und auch dem Umgang mit Daten sehr verschieden sind. Die Herausforderungen der Biodiversitätsinformatik haben folglich auch ihre Wurzeln darin, dass es die Biodiversitätsforschung an sich nicht gibt, sondern eine Vielzahl von Disziplinen, die unter diesem Begriff subsumiert werden.

2.1.2 Kartierung

Eine besonders wichtige Methode in der Biodiversitätsforschung ist die Beobachtung und Sammlung von biologischen Objekten im Gelände, was als Kartierung bezeichnet wird. Dabei werden Belege (engl. 'specimen' – siehe Abbildung 2.4)[4] gesammelt und bei biologischen Sammlungen eingereicht. Dieser wird mit einer Referenz auf den Sammler und das Sammelereignis archiviert.

[4]Das SNSB hat das Bild des Belegs in Abbildung 2.4 zur Verfügung gestellt und die Genehmigung zur Veröffentlichung in dieser Arbeit erteilt.

Definition 2.5: Beleg

Ein Beleg ist die während eines Sammelereignisses entnommen physische Probe einer biologischen Entität mit der entsprechenden Dokumentation des Sammelereignisses.

Wird kein physischer Teil eines biologischen Objektes zur Archivierung gesammelt, sondern lediglich die Existenz einer Art im Rahmen eines Kartierung dokumentiert, wird von einer Beobachtung (engl. 'observation') gesprochen.

Definition 2.6: Beobachtung

Eine Beobachtung ist jede Dokumentation einer Art im Gelände, bei dem keine Probe entnommen wird.

Die gleichzeitige Aufnahme einer Beobachtung und eines Beleges schließt sich nicht aus. Entscheidend ist, dass bei einem Beleg das biologische Objekt oder ein Teil dessen zur Archivierung entnommen wird. Die Existenz dieses Objektes kann separat durch eine Beobachtung beschrieben werden. Ein wichtiger Nachweis von Beobachtungen sind dabei Multimediadokumente wie Bilder oder Videoaufnahmen.

2.1.3 Messung von Biodiversität

Biodiversität ist auch als eine messbare Größe bestimmbar. Dies ist erforderlich, um Fragestellungen in Bezug auf die Veränderungen der Biodiversität über die Zeit zu formulieren [72]. Dazu bezieht sich die Messung der Biodiversität stets auf eine bestimmte Teilkomponente der Biodiversität und auf ein bestimmtes Areal. Im folgenden wird exemplarisch das Problem der Messung der Artenvielfalt an einem bestimmten Ort betrachtet. Auch für die anderen Teilbereiche der Biodiversität können Kennzahlen bestimmt werden. Für einen umfassenden Überblick über den Bereich der Messung von Biodiversität wird auf [97, 104, 243] verwiesen.

Ein häufig verwendetes Maß für die Artenvielfalt in einem Bereich ist dabei der Simpson-Index λ, wie er in [243] beschrieben wird: Dazu wird zunächst eine Gemeinschaft von s Arten betrachtet, wobei die Art i jeweils mit der Wahrscheinlichkeit p_i auftritt:

$$\lambda := \sum_{i=1}^{s} p_i^2 \tag{2.1}$$

Ist eine Art dominant, nähert sich der Simpson-Index dem Wert 1 an, wogegen ein niedriger Simpson-Index für eine ausgewogene Artverteilung spricht. In der Praxis

Abbildung 2.4: Beleg aus der botanischen Staatssammlung München

wird deshalb der Kehrwert $1/\lambda$ als Diversitätsmaß verwendet.

Die Messung der Biodiversität basierend auf der Anzahl von Arten an einem bestimmten Ort ist nicht unproblematisch. Dies wird in [97] dahingehend kritisiert, dass nicht alle Arten zur Biodiversität in gleicher Weise beitragen. Dementsprechend sollte jede Kennzahl für die Biodiversität eines Areals stets auch ausdrücken, wie verschieden die betrachteten Arten voneinander sind [97]. Eine Lösung dieses Problems sieht [97] in einer unterschiedlichen Gewichtung von Arten.

Zusammenfassend sind folgende Informationen für den Informatiker von Interesse:

- Biodiversität ist eine messbare Größe, die sich auf ein Gebiet bezieht
- Es gibt eine Vielzahl von Methoden, um die verschiedenen Bereiche der Biodiversität zu messen
- Diese sind aber nicht universal anwendbar
- Über die Anwendbarkeit einer Methode auf einen bestimmten Bereich besteht innerhalb der wissenschaftlichen Gemeinschaften Uneinigkeit

Gebiete, in denen die gemessene Biodiversität besonders hoch ist, obwohl diese Gebiete an sich nicht sonderlich groß sind, werden dabei als 'Hotspots' [175] bezeichnet. Sie dienen dazu Prioritäten bei der Wahl von Gebieten zum Schutz der Biodiversität zu setzen.

2.1.4 Ziele

Die Ziele der Biodiversitätsforschung ergeben sich wie in Kapitel 2.1 beschrieben aus der CBD [28]. Dabei ist für die Wissenschaft insbesondere das Ziel der Erhaltung der biologischen Vielfalt von Interesse. Diese wird nach [226, 133] hauptsächlich durch die folgenden Entwicklungen bedroht:

- Biotopzerstörung und Biotopveränderung
- unkontrollierte Bejagung und Befischung
- chemische und physikalische Umweltbelastung
- Verdrängung durch invasive Arten
- Klimawandel

Örtlich kann eine der Ursachen vorherrschen, wobei allerdings meistens Ursachenkombinationen vorliegen, in denen der menschliche Faktor nicht isolierbar ist [226].

Der Verlust biologischer Vielfalt fand in den letzten Jahrzehnten in beängstigtem Ausmaß statt, wie wissenschaftlich umfassend dokumentiert ist (vgl. [64, 133, 199, 226]). Eine besonders kritische Entwicklung ist, dass die Auslöschung einer Spezies häufig auch die Auslöschung einer weiteren Spezies – zum Beispiel aufgrund einer Art als entscheidender Nahrungsbestandteil einer anderen Art – bedingt [129]. Auf diese Weise entsteht ein Dominoeffekt. Die Ausrottung einer Art muss auch nicht zwangsläufig sofort auftreten. Eine Art kann auch langsam über mehrere Generationen hinweg verschwinden (Aussterbeschuld [240]), so dass die Einschätzung, ob eine Art bedroht ist oder nicht, häufig fehlerbehaftet ist [133].

Im Rahmen der CBD [28] haben sich die unterzeichnenden Staaten zu einem Rückgang des Biodiversitätsverlusts bis zum Jahr 2010 verpflichtet. Obwohl viele Maßnahmen zur Erhaltung der Biodiversität getroffen wurden, konnte dieses Ziel bis auf wenige Ausnahmen nicht erreicht werden [199]. Dabei ist nach [199] die Erweiterung der wissenschaftlichen Erkenntnisse und die Investition in erfolgreiche Biodiversitätserhaltungsprogramme von entscheidender Bedeutung, jedoch alleine nicht ausreichend. Dementsprechend wurden in [199] für die Erhaltung der Biodiversität über das Jahr 2010 hinaus folgende politische Ziele formuliert:

- Management der Biodiversität als öffentliches Gut
- Integration der Biodiversität in die öffentliche und private Entscheidungsfindung
- Durchsetzung der Implementierung von Verträgen durch geeignete Institutionen, Überwachung und Verhalten

Um das Ziel der Erhaltung der Biodiversität überprüfen zu können, muss aber zunächst die Beobachtung der Biodiversität möglich sein und ist dementsprechend auch eines der entscheidenden Ziele der Biodiversitätsforschung (vgl. [197]).

Als Fazit lässt sich festhalten, dass das große Ziel in der Biodiversitätswissenschaft die Erhaltung der Artenvielfalt ist. Diesbezüglich wird in den verschiedensten Programmen entweder direkt oder indirekt durch die Verbesserung der Informationslage auf diese Ziel hingearbeitet, wobei die bisherigen Anstrengung in diese Richtung von bescheidenem Erfolg gekrönt waren. Deshalb ist es von entscheidender Bedeutung für die Zukunft die Anstrengungen diesbezüglich zu verstärken und besser zu strukturieren.

2.1.5 Wert der Biodiversität für die Menschheit

Um die Vielfalt der Biodiversität zu erhalten wird wie im vorangegangen Abschnitt beschrieben von den ratifizierenden Staaten der CBD ein enormer Aufwand betrieben. Dieser lässt sich nur rechtfertigen, wenn es sich bei der Erhaltung der Biodiversität um ein lohnenswertes Ziel handelt. Dabei liegt zuallererst in der Erhaltung der Vielfalt der Biodiversität ein besonders hoher Wert [62]. So kann beispielsweise das Leben des Menschen in einer intakten Umwelt als eigenständiger Wert betrachtet werden [226].

Tatsächlich gibt es neben der ethischen Komponente und dem Nutzen der Vielfalt an sich ökonomische, medizinische und soziale Gründe, welche den Nutzen der Vielfalt der Biodiversität belegen [196, 152]. Diese werden als Ökosystemserviceleistungen bezeichnet [22]. Zudem werden Berechnungen angestellt, um den monetären Wert der Biodiversität zu messen. Dieser wird in [37] mit einen Wert von über 30 Billionen Dollar jährlich beziffert. Folgende kurze Beispiele zeigen, worin der Wert der Biodiversität für den Menschen liegt.

Als erstes ist zu erwähnen, dass die Pharmaforschung viele ihrer Erkenntnisse und Rohstoffe aus der Natur bezieht und dabei auch auf eine reichhaltige Biodiversität angewiesen ist [152]. Bisher konnte dabei nur ein sehr kleiner Teil der Artenvielfalt auf ihre medizinische Nutzbarkeit getestet werden [152]. Dementsprechend kann in jeder verschwundenen Art die Chance verloren gehen, entscheidende Fortschritte im medizinischen Bereich zu erzielen [152]. Da insbesondere in den Hotspots der Biodiversität eine hohe Vielfalt auf sehr engen Raum gefunden werden kann, sind diese für den medizinischen Nutzen der Biodiversität besonders wichtig.

Auch in der Landwirtschaft kann die Biodiversität zur Züchtung oder genetischen Erzeugung neuer Nutzpflanzen verwendet werden [152]. Ausgangspunkt dafür ist, dass nur in wenigen Arten einer Nutzpflanze wichtige Eigenschaften vorhanden sind. Dies ist beispielsweise bei der Suche nach Resistenzen von Krankheiten oder Schädlingen von entscheidender Bedeutung. So wurden bei der Suche einer Reissorte, die gegen das RGS-Virus – einem Pflanzenvirus der in den 70er Jahren in Asien insbesondere beim Reisanbau enorme Schäden verursacht hat – resistent ist, eine Vielzahl von Reisarten untersucht, wobei sich lediglich eine Reissorte als resistent erwies [192]. In der Vielfalt der Arten liegt somit eine wichtige Quelle von Ressourcen, die in der Landwirtschaft nutzbar sind. Während in der Vergangenheit die Biodiversität fremder Länder allgemein nutzbar waren, werden diese nun durch die CBD [28] den Staaten Eigentumsrechte an der genetischen Vielfalt der eigenen Biodiversitätsreesourcen eingeräumt. Dies soll die Ursprungsländer an der eigenen Biodiversität

profitieren lassen und als Motivation zum Erhalt der Biodiversität dienen [28].

Biodiversität spielt auch im Umweltschutz eine entscheidende Rolle. So trägt die Biodiversität zur Verbesserung der Luft- und Wasserqualität bei[37] . Dies geschieht beispielsweise durch die Verhinderung von Erosion und einer Verlangsamung des Klimawandels [37]. Darüber hinaus kann zusätzlich angeführt werden, dass die Biodiversität in vielen Ländern die Grundlage des Tourismus ist und somit einen wirtschaftlichen Wert erhält [152].

Unabhängig von der Bezifferung des Wertes, welche in [62] kritisiert wird, bestehen für die Erhaltung der Biodiversität eine Vielzahl von triftigen Gründen, so dass diese als eines der Schlüsselziele der jetzigen Generation aber auch der zukünftigen Generationen betrachtet werden muss. Dabei dient die Vielfalt auch stets als Versicherung für die Erhaltung des Lebens an sich. Auf der Ebene der Nutzbarkeit durch den Menschen wird dies dadurch erreicht, dass in der Vielfalt der Biodiversität immer wieder eine biologische Struktur identifiziert werden kann, die bei der Lösung eines aktuellen Problems hilfreich ist[226]. Auf der Ebene des Lebens an sich ist die Vielfalt der Arten eine Versicherung, dass Leben auf der Erde weiterbestehen kann. Die Erde hat im Lauf der Jahrmillionen für Lebewesen die verschiedensten Lebensräume bereitgestellt. Viele Arten sind durch den Verlust ihrer Lebensräume ausgestorben und auch für den Menschen besteht keine Garantie, dass sich dieser dauerhaft dem Wandel der Lebensräume anpassen kann. Da durch die Vielfalt der Biodiversität auch immer Arten existiert haben, die durch einen Wandel profitieren, konnte die Existenz des Lebens an sich gewährleistet werden.

2.2 Biodiversitätsinformatik

Als Ausgangspunkt für die Bedeutung der Informatik kann dabei folgendes in [47] veröffentlichte Beispiel dienen, welches auch von der TDWG als Ausgangspunkt ihrer Tätigkeit herangezogen wird [232]: Da vor den 70er Jahren das Ausmaß der Klimaerwärmung und damit das Schmelzen des Eises an den Polen noch nicht über Satellitendaten erfasst werden konnte, gab es diesbezüglich bis zur Veröffentlichung von [47] keine Erkenntnisse. In [47] wurden hingegen die Aufzeichnungen von Walfangschiffen in der Antarktis als Datenbasis verwendet. Darin war der Ort jedes Walfangs seit 1931 dokumentiert und es konnte gezeigt werden, dass die Eisfläche in der Antarktis zwischen 1950 und 1970 um 25% zurückgegangen ist. Sicherlich hatte die Erfassung der Walfänge nicht den Zweck, den Klimawandel zu dokumentieren. Die Erkenntnisse in [47] wurden somit aus Daten gewonnen, die eigentlich aus einem anderen Grund aufgenommen wurden.

Dieses Beispiel zeigt eine der zentralen Herausforderungen der Biodiversitätsinformatik: Im Rahmen der Biodiversitätsforschung wurde bereits eine Vielzahl von Daten erhoben. Diese sind jedoch verschiedenartig, physikalisch verteilt und nicht organisiert [212]. Der Anspruch der Biodiversitätsinformatik muss es aber sein, neue Erkenntnisse aus diesen Daten zu gewinnen [221]. Die Biodiversitätsinformatik steht in diesem Bereich noch am Anfang.

2.2.1 Definition

Der Begriff der Biodiversitätsinformatik ist zunächst vom Begriff der Bioinformatik zu unterschieden. Dieser könnte theoretische als Oberbegriff der Biodiversitätsinformatik dienen, wird aber in der Praxis für die Informatik der molekularen Biologie verwendet [90, 221]. Unter dem Begriff der Biodiversitätsinformatik ist hingegen die Anwendung der Informatik auf die Domäne der Biodiversität wie sie in Abschnitt 2.1 beschrieben wurde zu verstehen. Der Begriff erhielt laut einer Korrespondenz zur Begriffsbestimmung von Berendsohn mit Kollegen [13] auf dem 'OECD Megascience Forum Working Group on Biological Informatics' 1996 in Paris Einzug in die heutige Gemeinschaft der Biodiversitätsinformatik[5].

Der Begriff Biodiversitätsinformatik wird dabei in [120, 221] in folgender Weise definiert und in dieser Arbeit so verwendet:

Definition 2.7: Biodiversitätsinformatik

Der Begriff Biodiversitätsinformatik umfasst die Anwendung von Informationstechnologie auf die Verwaltung, algorithmische Erkundung, Analyse und Interpretation von Primärdaten im Bezug auf das Leben. Einen besonderen Stellenwert dabei hat die Organisationsebene der Art.

Diese Definition ist auch mit dem Biodiversitätsverständnis von [14] kompatibel. In [120] wird dabei insbesondere Augenmerk auf die Anwendung der Informatik zur Verwaltung des Auftretens von Arten, der taxonomischen Art und Multimediadaten gelegt.

2.2.2 Aufgaben

Kennzeichnend für die Aufgaben der Biodiversitätsinformatik ist der Umgang mit enormen Datenmengen, welche in einer dramatischen Geschwindigkeit wachsen und durch das Internet begünstigt werden [221]. Dabei konnten beispielsweise bereits erste

[5] Erste Verwendungen des Begriffs konnten auf 1993 datiert werden

Erfolge durch die Vergrößerung der auswertbaren Gebiete und eine höhere Auflösung in diesen erzielt werden (vgl.[221]).

In [14] werden die Ziele der Biodiversitätsinformatik als die Erfassung, Speicherung, Bereitstellung und Analyse von Informationen über Organismen, Populationen, Taxa und ihre Interaktionen bezeichnet. Für GBIF [82] steht hingegen der Austausch von Daten im Vordergrund. Im Rahmen dieser Arbeit werden die folgenden Aufgabenbereiche als die Ziele der Biodiversitätsinformatik betrachtet:

1. Datenerhebung: Bei der Datenerhebung wird Biodiversitätsinformatik primär für die Digitalisierung der Daten verwendet. Dabei kann mit unterstützender Software beispielsweise mit der automatischen Aufnahme mit Standardwerten, Georeferenzierung, Verknüpfung von Daten eine deutlich höhere Datenqualität erreicht werden als mit einer analogen Datenaufnahme.

2. Datenspeicherung: Eine besonders wichtige Anforderung an die Biodiversitätsinformatik ist die langfristige Speicherung von Daten. Diese soll in zentralen Repositorien organisiert sein und so die Abhängigkeit der Daten von den physischen Ressourcen eines Projektes lösen.

3. Datenaustausch: Für den Datenaustausch werden Strukturen, Protokolle und Standards benötigt (vgl. Abschnitt 2.2.3). Dabei müssen Datenverluste vermieden und die Datenqualität gesichert werden.

4. Data Provenance: Um den Ursprung und die ursprüngliche Bedeutung von Daten erkennen zu können, müssen jegliche Veränderungen und Transformationen an Datensätzen sowie deren Herkunft dokumentiert werden.

5. Datenauswertung: Mit statistischen Methoden und Methoden des Datamining sollen letztendlich Schlüsse aus den erhobenen Daten gezogen werden.

Dabei ist die Beziehung dieser Ziele hierarchischer Natur. Ohne die Speicherung der Daten in zentralen Speichern ist kein Datenaustausch möglich. Bei GBIF [82, 232] wird aktuell die zentrale Problemstellung im Datenaustausch gesehen. Allerdings ist für eine sinnvolle Auswertung der Daten nicht nur der Austausch der Daten an sich erforderlich. Es ist vielmehr zwingend, dass die Daten an sich miteinander verglichen werden können. Dies ist nicht unbedingt der Fall, wenn Daten vom Schema her kompatibel sind, da hier die Daten trotz der kompatiblen Schemata noch eine unterschiedliche Bedeutung tragen können. So kann bei Messung der Länge eines biologischen Objekts, trotz der Betrachtung der gleichen Art der Referenzpunkt der

Messung verschieden sein. Auch wenn die Messmethode innerhalb der Datensätze hinreichend dokumentiert ist, wird in diesem Fall noch zusätzlich eine Methode zur Transformation der Daten benötigt. Darüber hinaus muss diese Transformation dokumentiert werden. Data Provenance kann deshalb als eine zusätzliche Herausforderung der Biodiversitätsinformatik betrachtet werden.

Für Auswertungen innerhalb der Biodiversitätsinformatik werden Daten von besonders hoher Qualität benötigt. Die Qualität der Daten wird insbesondere durch folgende Fehlerquellen [221] beeinträchtigt:

- Falsche Identifikation von Arten
- Veraltete Taxonomie (vgl. Abschnitt 2.1.1)
- Fehlerhafte Georeferenzierung

Die vorangegangenen Punkte beziehen sich dabei auf fehlerhafte Einträge in Datenbanken in der Biodiversitätsinformatik. Diese Fehlerquellen sind für einen Informatiker besonders schwer zu identifizieren, da diese Expertenwissen voraussetzen. Dabei ist für den zweiten Punkt die mangelnde Abdeckung mit aktueller Information der großen taxonomischen Dienstanbieter wie Species2000 die Hauptursache [221]. Da diese Aufgaben in der Biodiversitätsinformatik nur ansatzweise gelöst sind, werden immer wieder globale, zentralisierte Lösungen propagiert [212].

2.2.3 Infrastrukturen für Daten in der Biodiversitätsinformatik

Der Datenaustausch in der Biodiversitätsinformatik erfolgt über speziell dafür geschaffene Infrastrukturen. Dabei werden an dieser Stelle nur kurz allgemeine Merkmale von Infrastrukturen wiedergegeben. Eine umfassende Vorstellung von Infrastrukturen in der Biodiversitätsinformatik findet sich in Kapitel 7. Eine Infrastruktur besteht aus verschiedenen Komponenten, welche den Datenaustausch zwischen dem Empfänger und den Quellen ermöglicht. Diese sind im Folgenden:

- Primärer Datenspeicher: Darunter ist der Datenspeicher zu verstehen, in welchem die Primärdaten zuerst in digitalisierter Form abgelegt werden. Dies kann eine Excel-Tabelle auf dem PC oder die Datenaufnahme mit einem Mobilgerät sein. Kennzeichnend für diese Art von Datenspeicher ist, dass dieser im Rahmen eines Projektes verwendet und somit nicht dauerhaft betrieben wird.
- Repositorium: Unter einem Repositorium ist ein Datenspeicher zu verstehen, der auf die langfristige Datenhaltung ausgerichtet ist. Bei diesen handelt es sich im Allgemeinen um eine relationale Datenbank.

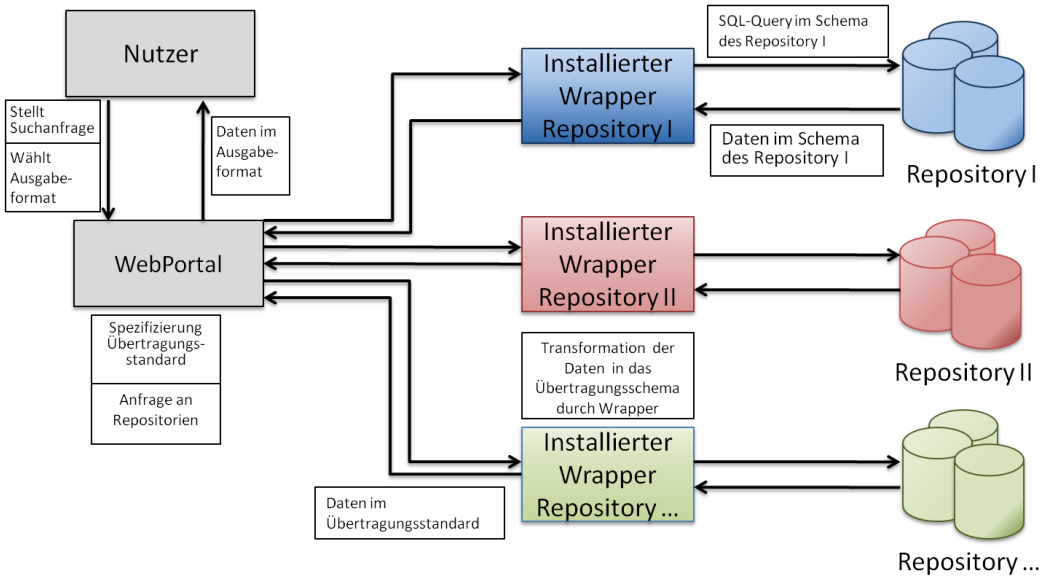

Abbildung 2.5: Nutzeranfrage an ein Datenportal

- Wrapper: Unter einem Wrapper ist ein Programm zur Transformation von Daten zu verstehen, welches mit Hilfe eines Protokolls den Datenaustausch zwischen zwei Datenspeichern moderiert. Dazu werden die Mapping-Informationen bei der Konfiguration des Wrappers erstellt.

- Protokoll: Das Protokoll dient als Schnittstelle für den Datenaustausch zwischen zwei Datenspeichern. Es enthält die Formatierung für Anfragen und die Rückgabe der Daten.

- Portale: Unter einem Portal ist eine Struktur zu verstehen, die es einem Endnutzer ermöglicht, Daten zu erhalten. Im Allgemeinen erlauben diese die Auswahl von Daten über eine Weboberfläche und stellen diese in verschiedenen Outputformaten zur Verfügung.

Eine Übersicht über den Datenfluss bei einer Anfrage an ein Portal mit heterogenen Quellen kann in Abbildung 2.5 betrachtet werden. Dabei ist charakteristisch, dass Daten aus verschiedenen Quellen in ein globales Schema transformiert werden und anschließend über das Portal verbreitet werden (siehe Kapitel 6.1). Dabei können weitere Schematransformationen stattfinden.

Die Abbildung 2.6 zeigt die Arbeitsweise des GBIF-Portals (siehe auch Abschnitt 2.3.3 und Abschnitt 7.4.1) mit verschiedenen GBIF-Knoten, wie diese am SNSB-IT-Center implementiert ist. Die Daten werden von GBIF einmal wöchentlich vom SNSB-IT-Center im ABCD-Format (siehe Abschnitt 2.3.4) in eine zentrale Daten-

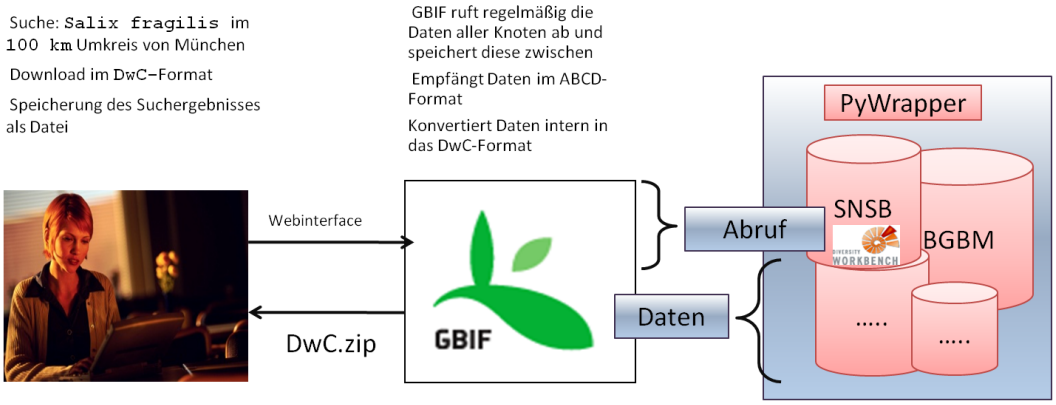

Abbildung 2.6: Nutzeranfrage an das GBIF-Portal

bank exportiert. Dazu muss der PyWrapper am SNSB-IT-Center installiert sein, welcher die Informationen zur Schemaübertragung vom SNSB in das ABCD-Schema enthält. Wie in Abschnitt 1.3 gezeigt wurde, liegt hier eine mögliche Quelle für Datenverluste. Von dort werden die Daten bei Nutzeranfragen im GBIF-Portal in das gewünschte Outputformat transformiert und über ein .zip-File zum Download angeboten (siehe [80]). Aktuell kann als Output-Format zwischen DarwinCore (DwC siehe Abschnitt 2.3.4), .kml, die taxonomische Artenliste als .txt ohne Geodaten oder ein Excel gewählt werden.

2.2.4 Prozesse in der Biodiversitätsinformatik

Die Aufgabe der Erhaltung der Biodiversität ist häufig in Projekten organisiert. Innerhalb dieser Projekte werden die Projektziele in konkrete Handlungsabläufe umgesetzt. Im Rahmen dieser tritt eine Vielzahl von Prozessen auf. Im Folgenden werden exemplarisch Prozesse des IBF-Projektes [116, 52, 260] beschrieben. Dazu ist die Modellierungsmethode des 'Perspective Oriented Process Modelling' [114] besonders gut geeignet, welches in Abschnitt 4.1 vorgestellt wird.

Der typische Prozess einer Begehung mit nichtdigitaler Datenerfassung ist in Abbildung 2.7 modelliert. Dabei ist zu erkennen, dass innerhalb des Ablaufs ein gewisses Maß an Flexibilität benötigt wird. Bei den verwendeten Werkzeugen ist zu erkennen, dass neben den Geräten zur Datenaufnahme auch die Artbestimmung und eine Karte des Gebiets benötigt wird. Diese Abläufe und Hilfsmittel mögen für einen Biologen selbstverständlich sein, sind aber bei der Erstellung von unterstützenden Programme in der Informatik unverzichtbar, da sich in diesen die Anforderungen der Nutzer widerspiegeln. So konnte aufgrund solcher Prozessmodelle mit DiversityMobile ein Programm zur digitalisierten Datenerfassung [116, 52] erstellt werden, welches ne-

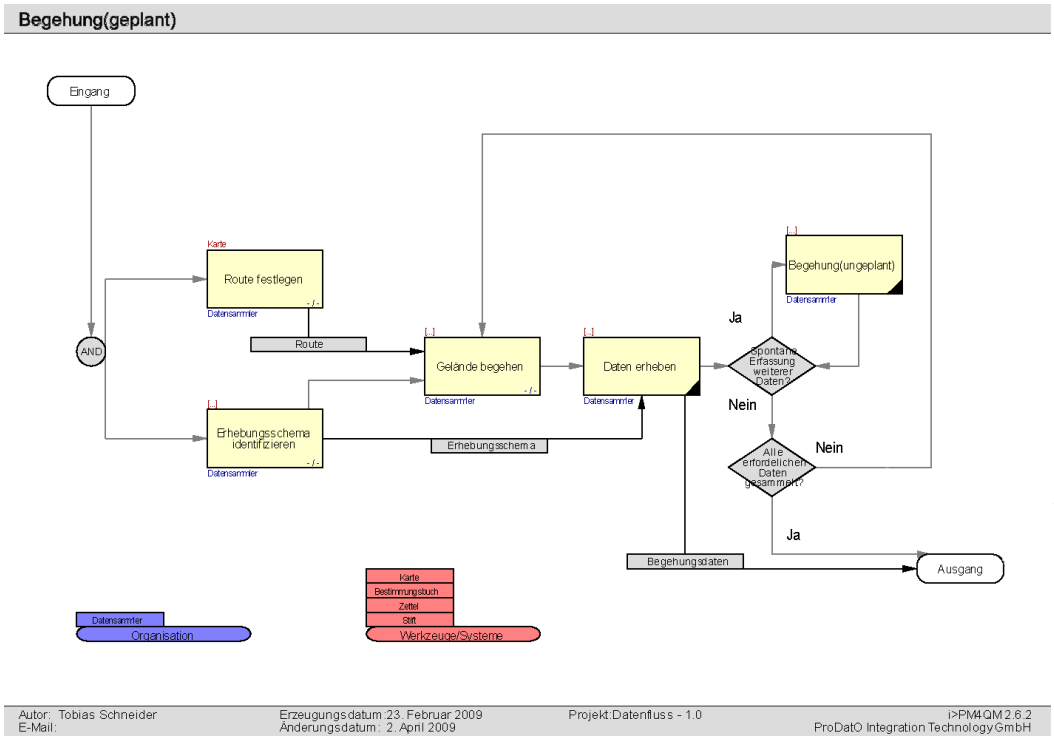

Abbildung 2.7: Prozess einer Begehung in IBFPlants

ben der Datenaufnahme für die identifizierten Arten gleichzeitig eine taxonomische Unterstützung bietet und die Arbeit mit Karten ermöglicht.

2.3 Strukturen in der Biodiversitätsinformatik

Zum Abschluss werden wichtige Organisationen, Projekte, Portale und Standards der Biodiversitätsinformatik vorgestellt. Selbstverständlich kann im Rahmen dieser Übersicht nur ein kleiner Ausblick gegeben werden. Für einen Überblick über 600 Projekte und Datenquellen in der Biodiversitätsinformatik, wird der Leser auf [232] verwiesen.

2.3.1 Organisationen

Der dauerhafte Zusammenschluss von Wissenschaftlern und Institutionen ist bei der Umsetzung von Projekten und Standards sowie der Umsetzung von Abkommen von entscheidender Bedeutung [199]. Im folgenden Abschnitt wird eine Reihe von Organisationen vorgestellt, welche hierfür von Bedeutung sind.

TDWG

Die TDWG traf sich als 'Taxonomic Database Working Group' erstmals 1985 auf Initiative der 'International Union of Biological Sciences (IUBS)' in Genf und ist heute eine ständige Einrichtung, welche sich nach mehreren Namensänderungen als 'Biodiversity Information Standards' bezeichnet. Die Informationen des folgenden Abschnitts sind dabei sofern nicht anders angegeben [232] entnommen. Ihren Hintergrund hat die TDWG im Bereich der Standardisierung von Pflanzensammlungen, wobei der Anspruch heute in der allgemeinen Standardisierung von Objekten zum Datenaustausch in der Biodiversitätsinformatik liegt.

Dabei betrachtet die TDWG folgende Ziele als ihre Hauptaufgaben:

- Entwicklung und Verbreitung von Standards und Richtlinien für die Aufnahme und den Austausch von Daten über biologische Organismen
- Förderung der Nutzung von Standards durch möglichst geeignete und effektive Maßnahmen
- Als Service zum Austausch über Publikationen und Meetings

Die TDWG arbeitet zum Erreichen dieser Ziele eng mit der 'Global Biological Information Facility (GBIF)' und dem 'Open Geospatial Consortium (OGC)' zusammen. Die TDWG untergliedert sich dabei in über 15 Arbeitsgruppen, in welchen Standards für die Biodiversitätsinformatik entwickelt werden. Für diese Arbeit sind insbesondere die folgenden von der TDWG publizierten Standards relevant, welche in Abschnitt 2.3.4 kurz vorgestellt werden:

- DarwinCore (DwC)
- Access to Biological Collection Data (ABCD)
- Structured Descriptive Date (SDD)
- TDWG Access Protocol for Information Retrieval (TAPIR)

Darüber hinaus entwickelt die TDWG Standards zur eindeutigen Kennzeichnung von Pflanzen die 'Life Sciences Identifiers (LSID)'. Diese können analog zu 'Globally Unique Identifier (GUID)' dazu verwendet werden, um biologische Organismen zu kennzeichnen. Die TDWG publiziert außerdem den Delta Standard, der durch den SDD Standard ersetzt werden soll. Delta ist aber in der Praxis noch weit verbreitet. Um die in Abschnitt 2.1.1 beschriebenen Probleme in Bezug auf verschiedene Taxonomien zu behandeln, wurde von der TDWG das 'Taxonomic Concept Transfer Schema (TCS)' entwickelt und wird als Standard von der TDWG empfohlen.

Abbildung 2.8: Offizielles GBIF Logo

GBIF

Die 'Global Biodiversity Information Facility (GBIF)' ist eine internationale aus öffentlichem Mitteln finanzierte Organisation mit Sitz in Kopenhagen [82, 135]. GBIF[6] sieht es dabei als seine Aufgabe an, Daten aus dem Bereich der Biodiversitätsforschung allgemein verfügbar zu machen [82]. Dazu stellt GBIF folgende Dienste bereit:

- Eine Infrastruktur zur Verteilung der Daten aus Sammlungen und Feldbeobachtungen (siehe Abschnitt 7.4.1).
- Schnittstellen, Protokolle und Standards in Zusammenarbeit mit der TDWG (siehe Abschnitt 2.3.1)
- Zugang zu internationalen Mentor- und Trainigsprogrammen

Um das erste Ziel zu verwirklichen, hat GBIF ein Netzwerk von Datenanbietern geschaffen, welches biologische Primärdaten über das GBIF-Portal (siehe auch Abschnitt 2.3.3) zur Verfügung stellt. Datenanbieter müssen dazu die Werkzeuge der Infrastruktur und die Protokolle implementieren. Aktuell werden über ca. 400 Datenanbieter ca. 340 Millionen georeferenzierte Primärdatensätze zur Verfügung gestellt [82]. Besondere Bedeutung innerhalb dieser Infrastruktur kommt den 'Participant Biodiversity Information Facility' (BIF) oder auch GBIF-Knoten zu, welches die einzelnen Datenanbieter koordiniert und zum Datenaustausch neben der technischen Infrastruktur auch Personal zur Verfügung stellt [75] (z.B. durch ein Trainingsangebot). Aufgaben eines GBIF-Knotens ist die Förderung, Koordination und Unterstützung des Datenaustauschs innerhalb der Domäne seiner Teilnehmer [75]. Dabei stellt der GBIF-Knoten die primäre Anlaufstelle eines Teilnehmers dar und ist auf der anderen Seite auch Ansprechpartner des GBIF-Sekretariats [74]. GBIF-

[6] Das GBIF-Sekretariat hat das offizielle GBIF-Logo in Abbildung 2.8 zur Verfügung gestellt und die Genehmigung zur Veröffentlichung in dieser Arbeit erteilt.

Knoten sind dabei häufig auf nationaler Ebene organisiert, welche die Datenanbieter eines Landes koordinieren.

Weitere Organisationen

Auf nationaler Ebene sind die 'Staatlichen Naturwissenschaftlichen Sammlungen Bayerns (SNSB)' in München [220] und der 'Botanischer Garten und Botanisches Museum Berlin-Dahlem (BGBM)' [13] von besonderem Interesse. Beide Organisationen erfüllen die GBIF-Standards als GBIF-Knoten und sind Ausgangspunkt für Projekte im Bereich der Biodiversitätsforschung [59, 116, 52]. Dabei hatte das BGBM maßgeblichen Anteil an der Entwicklung und Verbreitung des ABCD-Standards und hat als Institution an dem EDIT [59] und der Entwicklung des BioCASE-Protokolls, welches eng mit ABCD verknüpft ist, teilgenommen.

International ist noch das vornehmlich in den USA operierenden 'Long Term Ecological Relationship' (LTER)-Netzwerk [146] erwähnenswert. Dieses verbindet über 1800 Wissenschaftler aus 26 Institutionen zum Studium ökologischer Prozesse. Dabei verwendet LTER zur Datenspeicherung im Allgemeinen keine TDWG-Standards sondern die 'Ecological Metadata Language' (EML). Eine aktuelle Entwicklung aus dem LTER-Netzwerk heraus ist dataONE, welches auch EML als Standard für Metadaten verwendet [44]. Auch das 'National Biodiversity Network (NBN)' aus Großbritannien verwendet keine TDWG-Standards sondern hat zur Datenhaltung und -austausch eigene Standards definiert [177].

Für die Bereitstellung taxonomischer Dienste sind Species2000 [224] und das 'Integrated Taxonomic Information System (ITIS)' von Bedeutung. Diese beiden Organisationen betreiben gemeinsam das Datenportal 'Catalogue of Life' (COL) [35] (siehe Abschnitt 2.3.3), welches weltweit eine der wichtigsten Quellen für taxonomische Daten ist.

2.3.2 Projekte

Innerhalb der Biodiversitätsinformatik werden insbesondere Aufgaben in Bezug auf die Infrastruktur in Projekten organisiert, die zur Aufgabe haben, dauerhafte Strukturen zu schaffen. So wurde im Rahmen des BioCASE-Projektes [17] von 2001 bis 2004, ein Netzwerk und ein Datenportal geschaffen, welches bis heute Bestand hat. Auch die ursprüngliche Entwicklung des PyWrappers geht auf das BioCASE-Projekt zurück.

Das Projekt zum 'Setting up an Information Network on Biological Research Data gained in the Field up to the Sustainable Storage in a Primary Data Repository

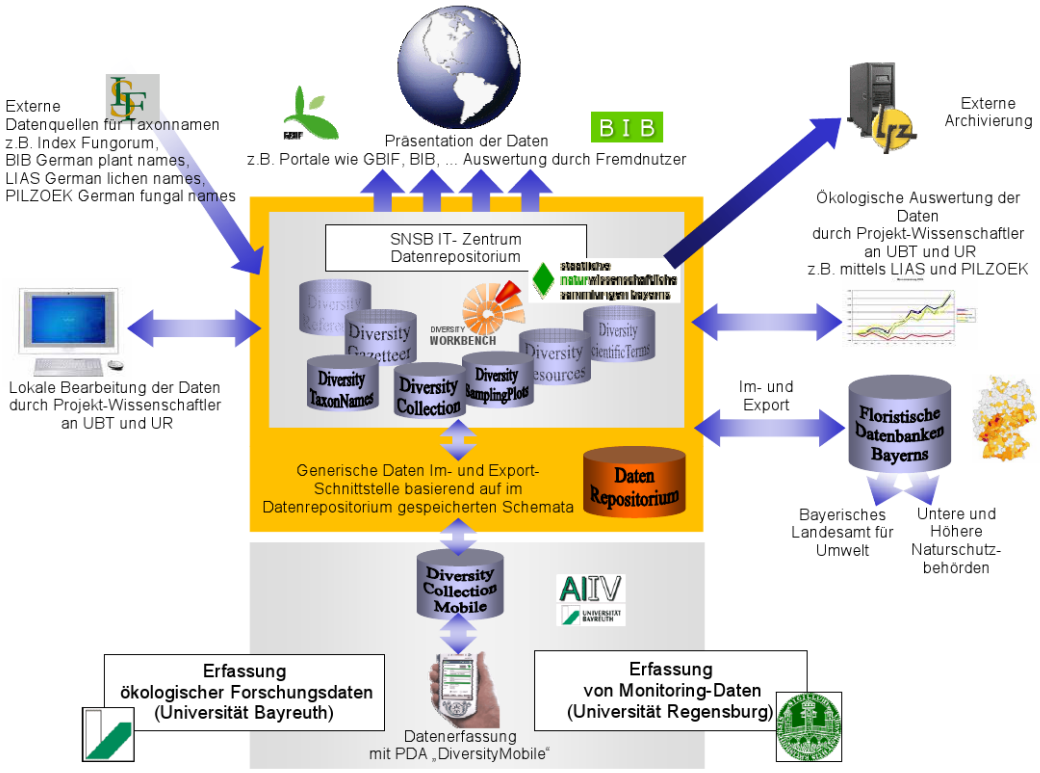

Abbildung 2.9: Datenfluss in IBF nach [106]

(IBF)' hat die Erstellung einer kontinuierlichen Dateninfrastruktur von der Datenerhebung im Feld über ein zentrales Repositorium bis zur Datenverbreitung über ein Portal zum Ziel [52]. Im Rahmen dieses Projektes wurde für die Datenaufnahme eine Anwendung für WindowsMobile und WindowsPhone geschaffen, mit der Daten mit Hilfe einer Synchronisationssoftware an das SNSB-IT-Center übertragen werden. Anschließend können der Allgemeinheit diese Daten über verschiedene Portale zugänglich gemacht werden. Die Infrastruktur des IBF-Projektes wird in Abbildung 2.9 gezeigt und in Abschnitt 7.4.2 evaluiert.

Das 'European Distributed Institute of Taxonomy (EDIT)'-Projekt ist ein internationales Projekt mit dem Ziel zur Schaffung eines Netzwerks für die taxonomische Forschung im Bereich der Ökologie- und Biodiversitätsforschung [57], das im März 2011 abgeschlossen wurde. Im Rahmen dieses Projektes wurde eine Reihe von Werkzeugen entwickelt, welche den Umgang mit taxonomischen Daten erleichtern sollen. Dazu zählt das 'Common Data Model' (CDM) – eine Java-Klassenbibliothek für taxonomische Entitäten – sowie das 'CDM Data Portal', welche die Veröffentlichung von in CDM verwalteten Daten ermöglichen soll.

Portal	Anwendungshintergrund	URL
Arctos	Sammlungsdaten	arctos.database.museum
Barcode of Life	Gensequenzen	www.boldsystems.org
BIB	Pflanzen (allgemein)	www.bayernflora.de
BioCASE	Artvorkommen (weltweit)	search.biocase.org
Biodiversity Heritage Library	Taxonomie	www.biodiversitylibrary.org
Catalogue of Life	Taxonomie	www.catalogueoflife.org
DiscoverLife	Artvorkommen (weltweit)	www.discoverlife.org
Encyclopedia of Life	Quellenverknüpfung	www.eol.org
Fishbase	Backgroundinformationen Fische	www.fishbase.org
GBIF	Artvorkommen (weltweit)	data.gbif.org
GenBank	Genetik	www.ncbi.nlm.nih.gov
IABIN	Artvorkommen (Amerika)	www.iabin.net
IUCN Redlist	Bedrohte Arten	www.iucnredlist.org
LIAS	Ascomyceten	www.lias.net
LTER	Primärdatenquellen	metacat.lternet.edu
NBN	Artvorkommen (regional)	data.nbn.org.uk
OBIS	marine Biodiversität	iobis.org

Tabelle 2.1: Portale in der Biodiversitätsinformatik

2.3.3 Portale

Portale stellen Daten im Allgemeinen vor einem bestimmten Anwendungshintergrund zur Verfügung. Dieser kann sich sowohl auf eine spezielle Art von Daten, als auch auf eine spezielle Domäne oder aber auch Region beziehen. So werden über das Portal des 'Catalogue of Life' (COL) ausschließlich taxonomische Daten verteilt [35], während über LIAS sowohl die Taxonomie als auch die Verbreitung von Schleimpilzen in Deutschland erhältlich ist [140]. Eine Sonderstellung nimmt dabei das GBIF-Portal ein, welches es sich zur Aufgabe gemacht hat, Aufzeichnungen über das weltweite Auftreten von Arten zur Verfügung zu stellen und dazu über ein großes Netzwerk an Quellen verfügt. Auch das LTER-Netzwerk verfügt über ein Datenportal [146]. Im diesem können aber keine Primärdatensätze geladen werden, sondern nur Informationen in EML über die Projekte, in denen sie entstanden sind. Für die Primärdatensätze an sich muss ein Projektverantworlicher kontaktiert werden.

Eine Übersicht über Portale ist in der Tabelle 2.1 mit Nennung des Anwendungshintergrunds angegeben.

2.3.4 Standards

Im folgenden Abschnitt wird ein kurzer Überblick über Standards in der Biodiversitätsinformatik gegeben. Eine umfassende Evaluation von Standards dieser Domäne wird in Kapitel 4 ausgeführt. Dabei ist die Spezifikation über ein XML-Schema die bevorzugte Methode, um einen Standard festzulegen.

DarwinCore (DwC) und Access to Biological Collection Data (ABCD)

'DarwinCore' (DwC) und 'Access to Biological Collection Data' (ABCD) sind Standards zum Datenaustausch, die von der TDWG als XML-Schema definiert werden. Beide Standards haben die Erfassung und Beschreibung von biologischen Primärdaten aus Sammlungen und Feldbeobachtungen, sowie die Aufnahme von assoziierten Daten (wie z.B. des Wissenschaftlers, der ein Objekt kartiert) als Aufgabe. DwC hat dabei seine Ursprünge im DublinCore – einem Standard der Bibliotheksverwaltung, wogegen ABCD mit dem konkreten Ziel eines Datenaustauschstandards geschaffen wurde. Da beide Standards vor einem vergleichbaren Anwendungshintergrund definiert sind, stellt sich die Frage, ob tatsächlich beide Standards benötigt werden. ABCD ist mit über 1000 Feldern der umfassendere Standard. DwC berücksichtigt nur ca. 100 Felder und es wird mit dem 'SimpleDarwinCore' sogar eine vereinfachte Variante angeboten.

ABCD erhebt den Anspruch, die Anforderungen der Biodiversitätsforschung möglichst vollständig zu beschreiben und so mit möglichst vielen existierenden Datenstandards kompatibel zu sein. Dabei wurde auf rekursive Strukturen verzichtet, welche in DwC gebräuchlich sind. Als Ergebnis dieser Entwicklung stellt sich ABCD als eine lange, gruppierte Liste mit über 1200 Konzepten dar, deren Vokabular verschlagwortet ist (bzw. sein sollte). Innerhalb der Spezifikation von ABCD werden eine Vielzahl von Feldern (z.B. Konzept 1314: Breed) nur unzureichend beschrieben (z.B. Konzept 1241: MeasurementsOrFacts). Darüber hinaus wird in beiden Standards die Möglichkeit gegeben, diese zu erweitern. Dementsprechend existieren in der Praxis eine Reihe von Erweiterungen von ABCD und DwC für spezifische Anwendungsfelder (vgl. [1, 67, 232]).

Neben dem unterschiedlichen Architekturansatz hat ABCD somit den Anspruch an sich, DwC inhaltlich zu überdecken und potentiell ersetzen zu können. Allerdings beziehen sich die Mapping-Informationen von ABCD auf DwC und umgekehrt auf die Version des DwC 1.4 aus dem Jahr 2005 (vgl. [15]). Dieser wurde seitdem weiterentwickelt und beinhaltet eine Anzahl an Konzepten, welche aktuell nicht von ABCD unterstützt werden. Auf der anderen Seite beinhaltet ABCD (allein aufgrund

```
<?xml version="1.0" encoding="UTF-8" ?>
- <DarwinRecord xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dwcore="http://rs.tdwg.org/dwc/dwcore/"
xmlns:dwgeo="http://rs.tdwg.org/dwc/geospatial/"
xmlns:dwcur="http://rs.tdwg.org/dwc/curatorial/">
 <dwcore:GlobalUniqueIdentifier>LD:General:45352</dwcore:GlobalUniqueIdentifier>
 <dwcore:DateLastModified>2007-01-19T00:00:00Z</dwcore:DateLastModified>
 <dwcore:Remarks>Nom. Kavalas, Ep. Pangeou. Mt Pangeo. 6 km SW of Nikisiani. Stony
meadows. Altitude 1750 m.</dwcore:Remarks>
 <dwcore:ScientificName>Luzula luzuloides ssp. cuprina</dwcore:ScientificName>
 <dwcore:Kingdom>Plantae</dwcore:Kingdom>
 <dwcore:Family>Juncaceae</dwcore:Family>
 <dwcore:Genus>Luzula</dwcore:Genus>
 <dwcore:SpecificEpithet>luzuloides</dwcore:SpecificEpithet>
 <dwcore:InfraspecificRank xsi:nil="true" />
 <dwcore:InfraspecificEpithet>ssp. cuprina</dwcore:InfraspecificEpithet>
 <dwcore:AuthorYearOfScientificName xsi:nil="true" />
 <dwcore:NomenclaturalCode xsi:nil="true" />
 <dwcore:Continent>Europe</dwcore:Continent>
 <dwcore:Country>Greece</dwcore:Country>
 <dwcore:StateProvince>Nom. Kavalas</dwcore:StateProvince>
 <dwcore:County>Ep. Pangeou</dwcore:County>
 <dwcore:Locality>Ep. Pangeou</dwcore:Locality>
 <dwcore:MinimumElevationInMeters>1750</dwcore:MinimumElevationInMeters>
 <dwcore:EarliestDateCollected>1979-07-25</dwcore:EarliestDateCollected>
 nil="true">
 <dwgeo:DecimalLatitude>40.9</dwgeo:DecimalLatitude>
 <dwgeo:DecimalLongitude>24.1167</dwgeo:DecimalLongitude>
 <dwcur:Preparations>Herbarium specimen</dwcur:Preparations>
 <dwcur:IndividualCount>1</dwcur:IndividualCount>
 </DarwinRecord>
```

Abbildung 2.10: Auszug eines DwC-Datensatzes aus dem BioCASE-Portal

der größeren Anzahl) eine Vielzahl von Konzepten, die in DwC nicht berücksichtigt werden. Die Datenübertragung von DwC nach ABCD und umgekehrt ist aktuell nur unter Informationsverlust möglich.

Die Abbildung 2.10 zeigt einen Auszug aus einem DwC-Datensatz, an welchem eine Vielzahl von Nullwerten nicht aufgenommen wurden. Das heißt, DwC-Datensätze sind in der Praxis meist dünn besetzt. Bezeichnend sind hierbei die Nullwerte in den Feldern 'NomenClaturalCode' und 'AuthorYearOfScientificName'. Diese Felder beschreiben die verwendete Taxonomie und sind notwendig, um eine veraltete Taxonomie zu identifizieren (vgl. Abschnitt 2.2.2). Diese Felder sind überdies nur bis zur DwC-Version 1.4 und nicht in der aktuellen DwC-Version enthalten. Der Datensatz müsste dementsprechend noch der neuen Version des Standards angepasst werden.

Der Umstand dieser beiden konkurrierenden Standards zeigt Uneinigkeit innerhalb der Biodiversitätsinformatik bezüglich der Gestaltung von Standards. Aktuell wird von der TDWG der DwC als aktueller Standard empfohlen wogegen ABCD nur eine Empfehlung als 'TDWG 2005 Standard'[7] hat und als aktueller Standard noch ratifiziert werden muss [232]. Welcher Standard sich letzten Endes durchsetzen wird, beziehungsweise ob beide Standards parallel existieren können wird die Zukunft zeigen. Aktuell wird aber gerade durch diese Unstimmigkeiten bei fundamentalen Standards die Arbeit innerhalb der Biodiversitätsinformatik erschwert.

Structured Descriptive Data (SDD), DiversityDescriptions und DEscription Language for TAxonomy (DELTA)

'Structured Descriptive Data' (SDD) hat als Anwendungsfeld die Erfassung sogenannter beschreibender Daten und geht maßgeblich auf das Informationsmodell von 'DiversityDescriptions' [90] zurück. Ziel der Arbeit war es eine Weiterentwicklung des bis dato gebräuchlichen 'Description Language for Taxonomy (DELTA)' Standards. Ursprünglich wurde DELTA zur strukturierten Beschreibung taxonomischer Daten entwickelt [232], wird aber auch als Austauschstandard eingesetzt [90].

Der Begriff beschreibenden Daten wird in SDD in einem weiteren Sinne als in DELTA verstanden: Beschreibende Daten sind Daten über intrinsische Eigenschaften von Organismen, an Hand welcher Individuen, Populationen oder Taxa kontextunabhängig identifiziert werden können [90]. Dabei ist es eines der Ziele von SDD Beschreibungen aus verschiedenen Formaten und insbesondere auch natürlicher Sprache verwalten zu können [232]. Dies ist in DELTA nicht möglich. Darüber hinaus enthält DiversityDescriptions den vollen Funktionsumfang von DELTA und entwickelt DELTA zusätzlich weiter – z.B. durch die Einführung von 'Modifikatoren' als neue Terminologieklasse und der Möglichkeit statistische Kennzahlen zu verwalten. SDD wird genauso wie DELTA von der TDWG als 'TDWG 2005 Standard' zur Erfassung beschreibender Daten empfohlen [232].

TDWG Access Protocol for Information Retrieval (TAPIR)

Der aktuelle Ansatz zur Organisation des Datenaustauschs der TDWG ist das 'TDWG Access Protocol for Information Retrieval' (TAPIR)-Protokoll und wird von dieser als Standard geführt [232]. Ausgangspunkt für TAPIR ist die parallele Nutzung

[7] Diese Standards wurden auf der TDWG Konferenz 2005 in St. Peterburg ratifiziert, erfüllen aber noch nicht die Richtlinien zur Entwicklung von Standards, die auf der TDWG-Konferenz 2006 in St. Louis beschlossen wurden [232].

des ABCD- und DwC-Standards. Dabei wurde ursprünglich das BioCASE-Protokoll für den Datenaustausch mit ABCD und das DiGIR-Protokoll für den Datenaustausch mit DwC entwickelt. Mit TAPIR werden diese beiden Protokolle vereinigt und erweitert, so dass auf die ursprünglichen Protokolle nicht mehr zurückgegriffen werden muss. In der Praxis sind aber noch Installation von BioCASE und DiGIR zu finden.

Weitere Standards

Ein Standard zur Speicherung von Hintergrundinformationen ist der 'Ecological Metadata Standard (EML)' [128]. Der Begriff Metadata wird dabei von den Autoren des Standards synonym für Hintergrundinformationen verwendet. In diesem werden Informationen über Projekte, die Verantwortlichen und die geografische Lage des Projektes beschrieben. Es besteht die Möglichkeit, die verwendeten Methoden zu dokumentieren und ein kurzes Abstract über das Projekt zu schreiben. Der EML-Standard wird beispielsweise vom LTER-Netzwerk zur Beschreibung von Projekten verwendet.

Ein besonders interessanter Ansatz zur Speicherung von Beobachtungsdaten wird mit der 'Extensible Observation Ontology (OBOE)' in [149] eingeführt. Ziel dieser Ontologie ist es, die Bedeutung der Daten bei der Beobachtung und Messung von biologischen Objekten zu erfassen. Wie in Abbildung 2.11 zu erkennen ist, ist die Grundstruktur von OBOE dazu abstrakter definiert als ABCD oder DwC. Darüber hinaus werden Messungen explizit berücksichtigt und können innerhalb des Datenformats genau beschrieben werden.

Ein entscheidender Vorteil von OBOE liegt auf der Ebene der Datenintegration: Es können verschiedene OBOE-Datensätze darauf überprüft werden, ob diese miteinander kompatibel sind. Dazu und für die Vereinigung von Datensätzen wird eine algorithmische Vorgehensweise beschrieben. Obwohl OBOE kein Datenstandard an sich ist, da OBOE nicht von einem standardisierenden Gremium entwickelt wurde, ist OBOE in diesem Zusammenhang von Bedeutung, da OBOE in einer Vielzahl von Projekten verwendet wird und somit die Grundstruktur für eine relevante Anzahl an Datensätzen liefert.

Ein weiteres nicht als Standard ratifiziertes Modell zur Datenspeicherung ist das im EDIT-Projekt entstandene 'Common Data Model (CDM)' [59]. In diesem wird ein umfassendes Framework von Java-Klassen gegeben, welches Entitäten aus der Biodiversitätsforschung beschreibt. Darüber hinaus stellen viele Portale (z.B. GBIF, BioCASE, OBIS) ihre Daten über .csv-Dateien oder, wenn diese georeferenziert sind,

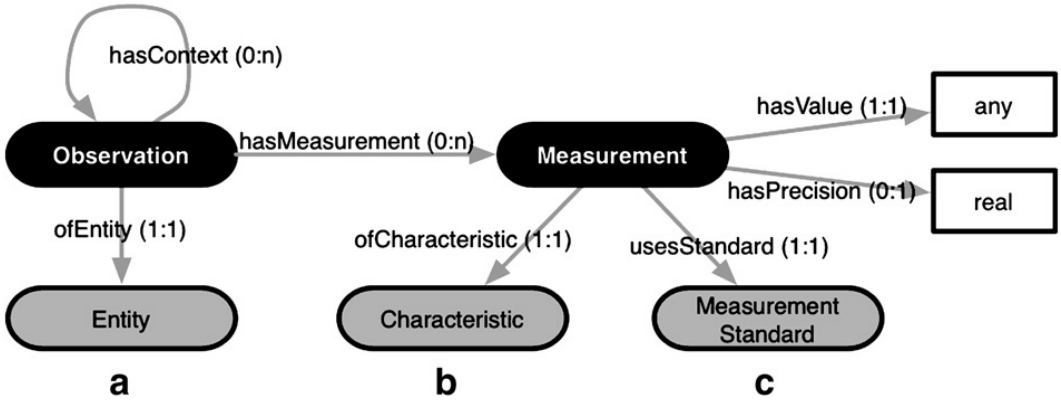

Abbildung 2.11: Grundstruktur von OBOE nach [149]

über das .kml-Format von Google-Earth zur Verfügung.

Fazit

Die TDWG ist die wichtigste Organisation für die Spezifikation von Standards in der Biodiversitätsinformatik. Dabei sind die wichtigsten Standards für die Kartierung und die Sammlungsverwaltung ABCD und DwC und müssen praktisch von allen Anwendungen im Bereich der Biodiversitätsinformatik unterstützt werden, da diese dazu in der Lage sind Primärdaten von Kartierungen zu erfassen. Dabei stellt ABCD die neuere Entwicklung und den umfangreicheren Standard dar. Trotzdem konnte DwC durch ABCD noch nicht verdrängt werden, da dieser strukturelle Vorteile gegenüber ABCD aufweist und breite Anwendung findet. Dementsprechend werden von bedeutenden Organisationen wie GBIF beide Standards in ihren Anwendungen verwendet und mit TAPIR ein Protokoll geschaffen, das beide Standards unterstützt. Aktuell ist fraglich, welcher dieser konkurrierenden Standards sich durchsetzten wird. Für die Verwaltung von Metadaten ist EML weit verbreitet. EML wird primär von US-amerikanischen Projekten und Organisationen wie LTER und DataONE eingesetzt. Vor diesem Hintergrund empfiehlt es sich, in eigenen Projekten entweder DwC oder ABCD zu verwenden. Dabei scheint aufgrund der besseren Struktur für die Datenübertragung DwC zur Anwendung in Kartierungsprojekten besser geeignet zu sein. Eine genaue Analyse dieser und weiterer Standards findet sich in Kapitel 4.

Kapitel 3

Grundlagen der Evaluation von Datenstandards

Im folgenden Kapitel werden bisherige Arbeiten aus dem Themenkomplex zur Evaluation von Datenstandards untersucht. Diese beziehen sich im Allgemeinen auf die Evaluation von Schemata von relationalen Datenbanken. Die so identifizierten Kriterien werden am Ende des Kapitels auf deren Anwendbarkeit in einem Framework zur Evaluation von Datenspeichersystemen geprüft.

Zunächst werden in Abschnitt 3.1 grundlegenden Definitionen dieses Themenbereichs eingeführt. Diese bewusst allgemein gehaltenen Begriffe sind notwendig, da in der Biodiversitätsinformatik neben der klassischen Speicherung in Datenbanken die Speicherung in XML-Formaten oder anderen Formaten wie Ontologien weit verbreitet ist. Es werden auch in Datenspeichern, die keine relationalen Datenbanken sind, Daten nach einem Schema zumindest teilweise strukturiert. Das heißt, auch Daten in XML-Form liegt im Allgemeinen ein Schema zu Grunde, welches für die Beurteilung der Qualität eines Datenmodells herangezogen werden kann. In der Literatur hat sich zur Bewertung dieser Schemata eine Theorie um die Evaluation konzeptueller Schemata entwickelt. Diese Theorie wird mit seinen wichtigsten Arbeiten in diesem Kapitel eingehend betrachtet, da es den Ausgangspunkt für die Entwicklung einer eigenen Theorie zur Evaluation von Schemata zur Datenspeicherung darstellt.

Der Abschnitt 3.2 beschäftigt sich mit etablierten Systemen zur Evaluation konzeptueller Schemata in der Literatur. Diese finden sich in der Literatur bei folgenden Teildisziplinen der Informatik:

- Requirements Engineering
- Software Engineering

Je nach Definition dieser Begriffe in der Literatur ist diese Aufteilung nicht disjunkt und die Begriffe können sogar einander umfassen. So kann das Requirements Engineering genauso als Teildisziplin des Software Engineering aufgefasst werden wie die Modellevaluation als Teildisziplin des Requirements Engineering. Die Problematik der Qualitätsbewertung eines Datenspeichersystems wird in diesen verschiedenen Gebieten aber aus unterschiedlichen Blickwinkeln und in einer anderen Granularität betrachtet. Dementsprechend ist es nahe liegend neben den einschlägigen Werken zur Evaluation von konzeptuellen Modellen auch Literatur zum Requirementengineering wie auch zum Softwareengineering zu betrachten.

Die Evaluationssysteme der Literatur werden dabei auf ihre Anwendbarkeit zur Evaluation von Standards und Datenspeichersystemen in der Biodiversitätsinformatik untersucht. Dabei werden zunächst die Kriterien der Frameworks an sich betrachtet und ob diese in der Domäne der Biodiversitätsinformatik geeignete Kriterien sind. Anschließend wird auf die Generizität der Kriterien eingegangen, da im Bereich der Biodiversitätsinformatik mit verschiedenen Techniken gearbeitet wird. Als letztes Bewertungskriterium für eine Framework zur Qualität der Evaluation der Datenmodellierung wird die Implementierung der Kriterien an sich herangezogen.

In Abschnitt 3.3 werden die betrachteten Frameworks einander gegenübergestellt und die Ergebnisse der Literaturrecherche zusammengefasst. Im Rahmen dieser Diskussion können bisher von der Literatur unbeachtete Kriterien identifiziert werden, welche als Grundlagen für die Entwicklung des Evaluationsframeworks 'Process Oriented Schema Evaluation' (POSE) (siehe Kapitel 4) dienen.

3.1 Grundlagen

Im folgenden Abschnitt werden die Grundlagen zur Evaluation von Datenstandards eingeführt. Diese beinhaltet neben der Einführung grundlegender Definitionen die Vorstellung von Technologien, die zur Speicherung von Daten verwendet werden. Darüber hinaus wird auf Standardisierung und den Kontext eingegangen, in welchem ein Datenstandard verwendet wird.

3.1.1 Konzeptuelle Modelle

Nach [96] ist eine Datenbank ein Modell eines tatsächlich existierenden Systems. Der Inhalt einer Datenbank stellt demnach jederzeit eine Momentaufnahme eines Zustands einer Anwendungsumgebung dar und eine Veränderung der Daten im Speicher korreliert stets mit einem Ereignis in der Anwendungsumgebung. Dementsprechend

spiegelt die Struktur einer Datenbank auch stets die Struktur eines Systems aus der Realität wider. Dabei kann die Bedeutung eines Datenschemas diesem nach [96] nicht direkt entnommen werden. Diese wird nämlich durch den Designer der Datenbank festgelegt. Diese Sichtweise war grundlegend für das 'Semantic Data Modeling' (SDM) für Datenbanken, welches je nach Blickwinkel eine spezielle Form oder einen alternativen Ansatz zur Entity-Relationship-Modellierung (ER) nach [31] darstellt. Da in der Praxis mittlerweile auch alternative Datenspeichersysteme wie Ontologien oder XML-Dokumente zu finden sind, soll im Rahmen dieser Arbeit sich nicht nur auf die Analyse von Datenbanken beschränkt werden. Dementsprechend wird an Stelle des Begriffs der Datenbank mit dem weiter gefassten Begriff des Datenspeichers gearbeitet, welcher folgendermaßen definiert ist:

Definition 3.1: Datenspeicher

Jedes System, welches zur dauerhaften und strukturierten Speicherung von Daten geeignet ist, wird als Datenspeicher bezeichnet. Dabei ist die konkreten technischen Realisierung unerheblich.

Entscheidend bei der vorangehenden Definition ist die Ausrichtung auf den Zweck des Systems – nicht die technische Realisierung. Dies ist konsistent mit der Nomenklatur von Infrastrukturen (siehe Kapitel 6), in welcher unter einer Datenquelle ein Datenspeicher unabhängig von seiner technischen Realisierung verstanden wird [139]). Der Begriff des Datenspeichers ist damit auf Ontologien und XML-Dokumente anwendbar. Die in [96] getroffene Aussage über Datenbanken lässt sich problemlos auf beliebige Systeme zur Datenspeicherung erweitern. Entscheidend ist der Gedanke, dass eine Menge von Daten nicht isoliert betrachtet werden kann, sondern dass diese stets einen Ausschnitt der Realität repräsentiert und somit eine bestimmte Bedeutung hat. Dementsprechend werden im Folgenden – wenn nicht explizit anders erwähnt – Erkenntnisse der Evaluation von relationalen Datenbanken analog auf die Evaluation von Datenspeichern im Allgemeinen angewendet.

Unter einem Schema wird in [96] ein Modell der Realität verstanden. In der Modelltheorie wird dabei nach [225] ein Modell als die Abbildung eines Originals in Form einer künstlichen Entität betrachtet. Diese Abbildung enthält nur die für den Anwendungszweck notwendigen Merkmale des realen Objektes. Für eine ausführliche Diskussion des Modellbegriffs wird der interessierte Leser auf [250] verwiesen. Eine besondere Rolle bei der Modellierung der Struktur eines Datenspeicher spielen dabei konzeptuelle Modelle, welche die Grundlage der Implementierung von Schemata in Datenspeichern darstellen. Eine Definition dieses Begriffs findet sich in [142].

Definition 3.2: Konzeptuelles Modell nach Lindland [142]

Ein konzeptuelles Modell ist eine Menge von Aussagen zur Spezifikation, welche zur Lösung eines Anwendungsproblems benötigt werden.

Diese Definition ist generisch gehalten und versteht Aussagen im Sinne der Logik und befindet sich im Einklang mit der Modelldefinition nach [225]. Dementsprechend ist es möglich mit dem Begriff des konzeptuellen Modells nach Lindland nicht nur relationale Datenbanken im Speziellen, sondern Datenspeicher in Allgemeinen zu betrachten. Zur Erstellung konzeptueller Schemata ist die Modellierung als Entity-Relationship-Diagramm nach [31] gängige Praxis, wobei auch alternative Modellierungstechniken wie z.B. UML oder SDM nach [96] zum Einsatz kommen. Mit Hilfe eines konzeptuellen Modells wird ein Abbild des Anwendungsproblems in einen Datenspeicher übertragen. Alle relevanten Entitäten und Beziehungen des Anwendungsproblems müssen bereits in dem konzeptuellen Modell hinterlegt sein. Im Modell nicht erfasste Entitäten verfügen über keine Repräsentation im physikalischen Speicher. Folglich ist die Analyse des zugrunde liegenden konzeptuellen Schemas ein wesentlicher Aspekt der Evaluation von Datenstandards.

3.1.2 Datenspeicher

Im folgenden Abschnitt wird ein kurzer Überblick über Technologien gegeben, welche als Datenspeicher zum Einsatz kommen können. Datenbanken zählen hierbei zu den wichtigsten Vertretern und werden in der Praxis primär eingesetzt. Darüber hinaus werden Daten aber auch in Form von XML oder Ontologien gespeichert. Die objektorientierte Programmierung (OOP) wird bei der Softwareentwicklung eingesetzt. In diesem Kontext treten auch Fragestellung im Bezug auf die Datenverwaltung und die Persistenz von Daten auf. Sie wird deshalb an dieser Stelle der Vollständigkeit wegen besprochen.

Datenbanken

Datenbanken kommen immer dann zum Einsatz, wenn besondere Anforderungen hinsichtlich des zu speichernden Volumens, der Ausfallsicherheit oder der Datenqualität gestellt werden [206]. Dabei ist insbesondere der Einsatz von 'relationalen Datenbankmanagemensystemen' (RDBMS) weit verbreitet. Die Modellierung von Schemata in Datenbanken wird dabei im Allgemeinen als ER-Modellierung nach [31] ausgeführt. Datenbanken sollen hier nicht intensiv besprochen werden, da diese Technologie hinlänglich bekannt ist. Der interessierte Leser sei zu Fragestellungen

bezüglich der Modellierung auf [122] und zu Fragen bezüglich RDBMS auf [206] verwiesen.

Neben den relationalen Datenbanken etablieren sich in den letzten Jahren zunehmend alternative Datenbanktechnologien wie objektorientierte Datenbanken oder NOSQL-Datenbanken. Die objektorientierten Datenbanken bieten eine Reihe von Vorteilen in komplexeren Anwendungsbereichen und unterstützen das Konzept der Vererbung aus der objektorientierten Programmierung [122]. NOSQL-Datenbanken werden nach ihrer internen Struktur in 'Key-Value-Stores', 'Dokumentdatenbanken', 'Column-Family-Stores' und 'Graphdatenbanken' unterteilt und stellen damit eine Abkehr vom klassischen relationalen Modell dar [207]. Einige NoSQL-Datenbanken wurden dabei für spezifische Anwendungsfälle implementiert und wurden aufgrund ihrer einfacheren Struktur primär für die Verwaltung von großen Datenmengen eingesetzt.

Objektorientierte Programmierung

Die objektorientierte Programmierung (OOP) stellt aktuell de facto den Standard für die Verwaltung von Daten in Computerprogrammen dar. Grundlegend für die OOP ist die Unterscheidung von 'Klasse' und 'Objekt'[134]. Objekte stellen die Abbildung eines konkreten Gegenstandes aus einer Realität dar und repräsentieren diese innerhalb eines Programmablaufs [250]. Klassen beschreiben den Typ eines Objekts und damit über welche Eigenschaften und Methoden ein Objekt verfügt [250]. Durch die Spezifikation der Eigenschaften im Typ wird ein Schema der Eigenschaften eines Objektes angelegt. Dementsprechend lässt sich aus der Vorlage einer Klasse eine Vielzahl an Objekten erzeugen. Dieser Vorgang wird als Instanziierung bezeichnet [250].

Die OOP soll hier nur kurz behandelt werden, da die Kenntnis dieser Technologie weit verbreitet ist. Für eine umfassende Einführung wird auf [134] verwiesen. Es soll aber hier kurz auf das Konzept der Vererbung von Spezifikationen eingegangen werden. Mit Hilfe der Vererbung kann eine Klasse um zusätzliche Eigenschaften erweitert werden und dadurch eine neue Klasse entstehen [134]. Diese wird als 'Unterklasse' der ursprünglichen Klasse bezeichnet und verfügt über alle Eigenschaften der ursprünglichen Klasse, welche als 'Oberklasse' bezeichnet wird. Dieser Vorgang wird Ableitung genannt. Ein Objekt der Unterklasse verfügt über alle Eigenschaften eines Objekts der Oberklasse und ist damit praktisch ein Objekt seiner Oberklasse. Es kann in der Programmierung in Methoden und Schnittstellen verwendet werden, welche für Objekte der Oberklasse geschrieben wurden. Dies wird als 'Prinzip der

Ersetzbarkeit' bezeichnet [134].

Strukturen in der OOP werden häufig in der 'Unified Modeling Language' modelliert [189]. Eine Einführung in die Modellierung mit UML findet sich in [23].

XML

Als zweite Grundlage zur Bewertung von Standards in der Biodiversitätsinformatik dient die Theorie zur Evaluation von Schemata zur Strukturierung von XML-Schemata. Die Struktur eines XML Dokuments kann dabei über eine 'XML Schema Definition' (XSD) oder eine 'Document Type Definition' (DTD) erfolgen [174] und auch gegenüber dieser Struktur validiert werden. Da XSD über eine Vielzahl von Vorzügen – wie z.B. dass ein XSD Schema selbst in validem XML geschrieben wird und deutlich flexibler ist – verfügt (vgl. [174]) und auch die W3C-Empfehlung [256] zur Strukturierung von XML-Dokumenten darstellt, hat sich XSD in den letzten Jahren gegenüber DTD durchgesetzt. So sind auch die wichtigsten Standards der Biodiversitätsinformatik DwC und ABCD im XSD-Format spezifiziert. Darüber hinaus ist die Datenspeicherung- und Übertragung in XML-Dokumenten in der Community stark verbreitet, so dass auch proprietäre Modelle im XSD-Format definiert sind und die Dokumentstrukturierung über XSD de facto als Standard in der Biodiversitätsinformatik betrachtet werden kann.

Die Erstellung von XSD-Schemata wird aktuell häufig noch über Modellierungssoftware wie dem Altova SchemaAgent von [5] ohne vorherige konzeptuelle Modellierung ausgeführt. Ein Überblick über die Evaluation von Werkzeugen zur Modellierung von XSD, wie X-Evolution (siehe [88]) findet sich in [151]. Die konzeptuelle Modellierung im Kontext von XML ist aktuell Forschungsgegenstand in Projekten wie eXolutio [126], welches in [127] vorgestellt wird.

Somit kann die in einem XSD gespeicherte Struktur Ergebnis einer konzeptuellen Modellierung und deren Umsetzung sein. Für die konkrete Wahl einer Modellierungssprache, wäre die Modellierung in Form eines ER-Diagramms ein nahe liegender Ansatz, da sich dieser in der Anwendung im Datenbankbereich bewährt hat. Die Modellierung von XSD in Form von ER-Diagrammen ist aber nach [179] insbesondere deswegen ungeeignet, da die hierarchische Struktur vom XML-Dokumenten in ER-Diagrammen nicht dargestellt werden kann. So ist es durchaus üblich, dass innerhalb einer XML-Struktur andere Strukturen eingebettet werden. Die Wahl des Wurzel-Elements bestimmt aber die Zugänglichkeit der untergeordneten Konzepte bereits zum Zeitpunkt der Modellierung. Dies ist ein entscheidender Unterschied zu Datenbanken, da in diesem Kontext über die Mächtigkeit der Abfragesprache SQL

Selektionen sehr variabel ausgeführt werden können. Weitere Unterschiede die einer ER-Modellierung im Wege stehen, ist die Tatsache, dass XML-Dokumente über eine unregelmäßige Struktur verfügen, strukturierte mit unstrukturierten Daten vermischt werden können und die Ordnung von Elementen beachtet werden muss (vgl. [180]).

Dementsprechend hat sich in der Literatur eine Vielzahl von Erweiterungen des klassischen ER-Modells gebildet (wie z.B. in [198, 180, 65]), mit dem Ziel die oben genannten Herausforderungen zu meistern. Ein anderer Ansatz ist die konzeptuelle Modellierung an Anlehnung an die Unified-Modeling-Language (UML) wie z.B. in [144] mit UXS (UML and XML Schema).

Ontologien

Eine Ontologie in der Informationstechnologie ist eine explizite Spezifikation einer Konzeptualisierung, wobei unter einer Konzeptualisierung eine abstrakte, vereinfachte Sicht der Anwendungsdomäne zu verstehen ist [87]. Die Erstellung einer Ontologie ist demnach die Suche nach einer Menge von Begriffen und Beziehungen, welche eine Domäne repräsentieren [87]. Die Erstellung von Ontologien erfolgt in der Praxis über formale Sprachen wie z.B. die 'Web Ontology Language (OWL)', welche eine W3C Empfehlung ist [255]. In der Biodiversitätsinformatik ist insbesondere OBOE [149] (siehe Abschnitt 4.4.8) und die 'TDWG Ontology' [230] von Bedeutung. Die TDWG Ontology versucht die Kernbereiche der Biodiversitätsforschung zu erfassen und Exportmöglichkeiten nach DwC, ABCD und SDD zu spezifizieren.

Formale Sprachen wie OWL, die in Ontologien verwendet werden, basieren im Allgemeinen auf dem 'Resource Description Framework (RDF)'[255], welches von der W3C als Framework zur Repräsentation von Informationen im Internet empfohlen wird [253]. Die Grundstruktur von RDF ist die Organisation von Aussagen in Tripeln von Subjekt, Prädikat und Objekt. Bei der Anwendung von RDF wird eine Domäne durch eine Menge von gerichteten Subjekt-, Prädikat- und Objektbeziehungen modelliert. Dementsprechend kann RDF auch zur Spezifikation von konzeptuellen Schemata im Sinne von [142] verwendet werden. Ein wesentlicher Aspekt bei der Verwendung von RDF ist die Arbeit mit URI´s zur eindeutigen Identifikation von Elementen von Tripeln. Die Identifikation über URI´s ist dabei für Subjekte und Prädikate zwingend vorgeschrieben wohingegen für Objekte die Verwendung von bestimmten elementaren Datentypen über sogenannte 'Literale' gestattet ist. Die eindeutige Referenzierung der Elemente einer Menge von Aussagen ist dabei von entscheidender Bedeutung. Durch diese Referenz ist jeder Begriff eindeutig defi-

niert und kann semantisch von Begriffen, die identisch benannt sind, unterschieden werden.

3.1.3 Standardisierung

Ein besonderes Merkmal eines Schemas zur Speicherung von Daten ist es, wenn dieses von einer Organisation veröffentlicht wird, welche über die Autorität verfügt, Standards beziehungsweise Normen zu definieren. Allgemein betrachtet ist ein Standard ein Format oder eine Struktur, auf die sich eine Nutzergruppe geeinigt hat [120]. Eine Norm im Sinne von Wirtschaft, Industrie und Wissenschaft ist gemäß [56] eine 'Vorschrift, Regel, Richtlinien o. Ä. für die Herstellung von Produkten, die Durchführung von Verfahren, die Anwendung von Fachtermini o. Ä.'. Dabei ist ein Standard innerhalb der Mitglieder des Anwendungsbereichs bindend, was im Allgemeinen dadurch erreicht wird, dass dieser entweder durch den Gesetzgeber oder durch ein Gremium mit einflussreichen Mitglieder aus dem Fachbereich ratifiziert wird. Streng genommen ist somit die Verwendung des Begriff 'Standard' erst dann gerechtfertigt, wenn diese Voraussetzungen erfüllt sind. Ein Beispiel für ratifizierte Standards sind die in Abschnitt 2.3.4 vorgestellten Datenstandards ABCD und DwC, welche von der TDWD (siehe Abschnitt 2.3.1) für die Domäne der Biodiversitätsinformatik herausgegeben werden. Im weiteren Verlauf dieser Arbeit soll aber auch aus Gründen der Einfachheit für das Schema eines Datenspeichers der Begriff 'Datenstandard' verwendet werden, wenn dieser noch nicht von einer Organisation ratifiziert wurde. Sofern der Umstand der Ratifizierung in einem Kontext relevant ist, wird aber explizit darauf eingegangen.

3.1.4 Umgebung eines Datenstandards

Ein besonderes Merkmal der Qualität eines Datenstandards ist auch die Umgebung, in welcher er eingesetzt wird. Dabei wird zwischen 'geschlossenen' (Abbildung 3.1) und 'offenen' (Abbildung 3.2) Umgebungen unterschieden. In einer geschlossen Umgebung sind alle Teilnehmer dieser Umgebung bekannt und es ist auch nicht vorgesehen, dass regelmäßig neue Teilnehmer in die Umgebung aufgenommen werden müssen. Diese Situation ist kennzeichnend für ein bestimmtes Projekt mit fest vorgegebenen Auftrag. In eine offene Umgebung müssen hingegen stets neue Mitglieder integriert werden können und somit auch stets neue Anforderungen berücksichtigt werden. Die Anforderungen an die Flexibilität des verwendeten Datenschema sind damit deutlich höher. Dies ist kennzeichnend für einen Datenstandard einer großen Domäne wie z.B. der Biodiversitätsinformatik.

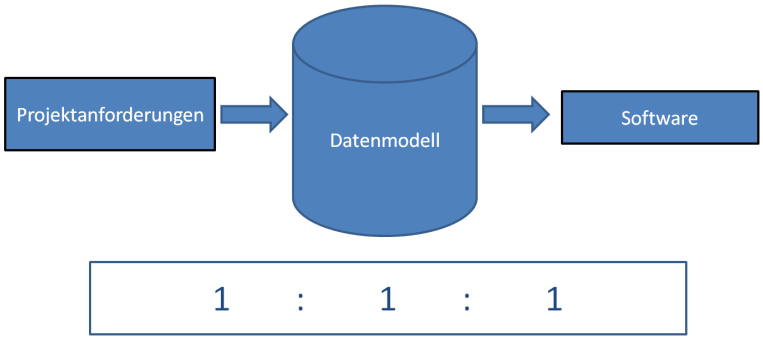

Abbildung 3.1: Datenmodell in einer geschlossenen Umgebung

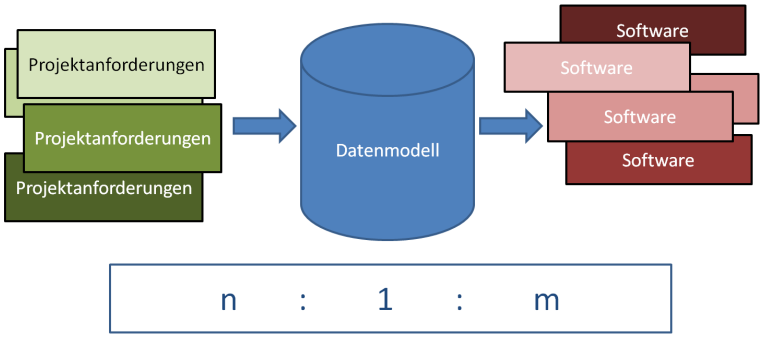

Abbildung 3.2: Datenmodell in einer offenen Umgebung

Die Besonderheit bei der Verwendung von Datenstandards in offenen Umgebungen liegt darin, dass diese – auf Basis der Ratifizierung einer relevanten Institution – für die verschiedensten Anwendungsfälle und auch für die verschiedenartige Softwareprodukte verwendet werden sollen. Dies ist ein Unterschied zur Anwendung der Evaluation von konzeptuellen Schemata im Requirementengineering, in welchem von der Anwendung des Systems innerhalb einer Organisation ausgegangen wird (vgl. [98]). Dies wird in den Abbildungen 3.1 und 3.2 illustriert.

In der Biodiversitätsinformatik sind z.B. die Standards ABCD und DwC von so großer Bedeutung, dass die Situation aus Abbildung 3.2 gegeben ist. Ein Gegenbeispiel ist LIAS (siehe Abschnitt 2.3.3), welches ein Datenschema verwendet, das auf Beobachtungsdaten von Schleimpilzen spezialisiert ist. Dieses konnte vor dem Hintergrund der exakten Anforderungen von LIAS definiert werden und es mussten keine weiteren Anforderungen berücksichtigt werden. Die Analyse eines weit verwendeten Standards wie ABCD oder DwC ist hingegen ungleich schwieriger, da hier die Interessen einer Vielzahl an Projekten berücksichtigt werden müssen.

3.1.5 Data Provenance

Eine weiterer wichtiger Bereich ist 'Data Provenance'. Die Bedeutung dieses Begriffes wurde im letzten Jahrzehnt deutlich erweitert. Diese bezieht sich aber stets auf die Herkunft der Daten einer Datenquelle. Ursprünglich wurde unter 'Data Provenance' die Herkunft eines Datensatzes und der Prozess verstanden, über den ein Datensatz in eine Datenbank integriert wurde [25]. Mit der 'Why-Provenance' und der 'Where-Provenance' werden zwei Arten von Data Provenance unterschieden. Dabei geht die 'Why-Provenance' der Frage nach, warum ein Datensatz in der Datenbank enthalten ist, und die 'Where-Provenance' woher der Datensatz ursprünglich stammt [25]. Die 'Where'-Provenance muss bei der Anwendung in einer Domäne wie der Biodiversitätsinformatik durch den Datenstandard unterstützt werden und ist bei den Prozessen in einer Infrastruktur zu dokumentieren.

Diese Unterscheidung ist heute noch gebräuchlich (vgl. [157] [1]). Eine umfassende Diskussion des Begriffs hierzu findet sich in [219]. Dabei kommt [219] zu dem Schluss, dass unter 'Data Provenance' alle Informationen zu verstehen sind, welche die Historie eines Datensatzes (beliebiger Technologie) beginnend bei der Originalquelle erfassen. Demnach soll mit 'Data Provenance' nicht nur die Herkunft eines Datensatzes erfasst werden, sondern auch alle Transformationen, die ein Datensatz durchläuft. Dabei wird unter Provenance eine Menge von Annotationen an die überwachten Datensätze verstanden [157]. Damit ist 'Data Provenance' ein wichtiges Werkzeug für die Identifikation von Datensätzen und beim Aufspüren von Artefakten durch Datentransformationen.

3.2 Evaluationssyteme für konzeptuelle Modelle

Die Evaluation von Datenstandards beruht primär auf den von diesen verwendeten Modellen. Deshalb werden im folgenden Abschnitt verschiedene Evaluationssysteme für konzeptuelle Modelle vorgestellt. Diese sind im Allgemeinen dem Requirements- und Softwareengineering entlehnt und für geschlossene Umgebungen konzipiert. Es konnten in der bisher veröffentlichten Literatur auch keine Systeme gefunden werden, welche offene Umgebungen explizit berücksichtigen. Viele der Kriterien aus diesen Evaluationssysteme sind jedoch auf die Verwendung von offenen Umgebungen übertragbar.

Die Evaluation von konzeptuellen Modellen geht hauptsächlich auf die Arbeiten

[1] Die spezialisierte Verwendung des Begriffs im Bezug auf Datenbanken darin ist aber zu restriktiv.

von [142], [171] und [258] zurück. Dabei wurden die ersten Ansätze zur metrischen Evaluation in [171] publiziert. Die Ansätze dieser Arbeiten wurden im Laufe der Zeit weiterentwickelt – wie z.B. der Ansatz aus [142] in [132] und [131]. Ein Überblick über verschiedene Systeme zur Schemaevaluation findet sich in [173]. In letzter Zeit hat die Evaluation von konzeptuellen Schemata durch Veröffentlichungen der Gruppe um M.Genero wie mit [85] an Aktualität gewonnen. So wurde mit [66] ein umfassendes Lehrwerk zum Bereich der konzeptuellen Modellierung publiziert und in [181] mit dem 'Conceptual Modeling Quality Framework' (CMQF) (siehe Abschnitt 3.2.3) eine Synthese der Theorien von [142] und [258] veröffentlicht.

Die Notwendigkeit einer expliziten Schemabewertung ergibt sich nach [171] aus der großen Anzahl an alternativen Möglichkeiten zu einem Anwendungsfeld ein Datenmodell zu definieren. Demnach wird in der Praxis zumeist ohne objektive Richtlinien und auf Basis von Einzelmeinungen entschieden, mit welchen Datenmodell gearbeitet wird. Objektive Kriterien zur Bewertung eines Datenmodells sind somit bei dem Design und der Wahl eines guten Datenmodells von entscheidender Bedeutung. Projekterfahrungen im Bereich der Biodiversitätsinformatik legen nahe, dass sich an der Aktualität dieser Einschätzung nichts geändert hat.

Selbstverständlich haben sich auch die Methoden, die in der Praxis verwendet werden, seit der Veröffentlichung von [171] weiterentwickelt. So aktualisiert Moody sein Framework in einer Vielzahl von Arbeiten (siehe [168, 169, 170, 172, 173]). Da nicht alle Frameworks mit allen Aktualisierungen vorgestellt werden können, wurde sich auf eine Auswahl beschränkt. Dies sind:

- Das Modelle von Lindland [142], als wichtiger, früher Ansatz
- Das Framework von Moody [168], da in diesem Framework erstmals Metriken zur Qualitätsmessung angeben wurden
- CMQF [181] als vereinheitlichende Theorie

Darüber hinaus wird ein kurzer Überblick zur Evaluation von XML-Dokumenten gegeben. Diese Frameworks werden auf die Anwendbarkeit in der Biodiversitätsin-

formatik untersucht. Dabei werden folgenden Kriterien angewendet:

1. geeignete Definition der Kriterien des Evaluationssystems
2. Generizität der Kriterien
3. Implementierung der Kriterien
4. Generizität der Implementierung
5. Anwendbarkeit in der Biodiversitätsinformatik

Unter Punkt 1 werden die Kriterien eines Frameworks an sich betrachtet und es wird der Frage nachgegangen, was unter diesen zu verstehen ist und ob diese als Kriterien geeignet sind. Da die meisten Frameworks für eine bestimmte Technologie und Domäne konzipiert wurden, wird in Punkt 2 erfasst, ob diese Kriterien sich auch auf andere Bereiche übertragen lassen. In Punkt 3 wird geprüft, ob ein System für die Ermittlung oder Messung eines Kriteriums eine Methode implementiert hat und ob diese zur Messung geeignet ist. Punkt 4 befasst sich mit der Frage, ob die Implementierung der Kriterien nur für spezielle Systeme geeignet und von der Art der Modellierung abhängig ist, sofern überhaupt eine Methode zur Implementierung angegeben wurde. In Punkt 5 wird letztlich subsumierend analysiert, ob das Framework zur Evaluation von Datenstrukturen in der Biodiversitätsinformatik geeignet ist.

3.2.1 Das Evaluationsframework nach Lindland

Vorstellung

Lindland et al. [142] entwarfen einen ersten Ansatz zur Evaluation von konzeptuellen Modellen (CM). In [142] wurde an den vorhergehenden Arbeiten kritisiert, dass die Definitionen zu vage und kompliziert und teilweise ungeeignet sind. Darüber hinaus wurde bemängelt, dass die Kriterien nur unstrukturiert aufgeführt sind und nicht sauber zwischen Sprach-, Anforderungs- und methodischer Ebene unterschieden wird. Um diesen Missständen entgegenzutreten wurden als Grundlage des Frameworks von Lindland folgende Forderungen aufgestellt [142]:

- Trennung von Anforderungen, Sprache und Methoden
- Trennung von Qualitätskriterien und Wegen, diese zu erreichen
- Mathematische Fundierung

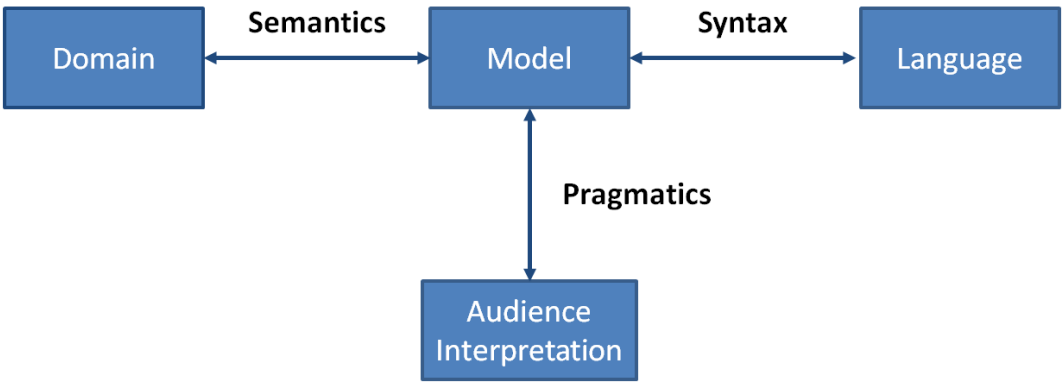

Abbildung 3.3: Grundlagen der CM-Bewertung nach Lindland aus [142]

- Direkter Bezug der Evaluation auf das gebildete Modell ohne dieses notwendigerweise implementieren zu müssen

Um diese zu erfüllen, dienen folgende Ebenen als Grundlage des Evaluationssystems nach [142] (siehe Abbildung 3.3):

- Syntax – Korrekte Modellierung in der Modellierungssprache: Verknüpft das Model mit der Modellierungssprache ohne die Bedeutung von Sprachkonstrukten zu interpretieren
- Semantik – Korrekte Modellierung der Domäne: Interpretation der im Model getroffenen Aussagen im Hinblick auf die Anwendungsdomäne
- Pragmatik – Interpretation des Modells durch die Nutzer: Stellt einen Bezug zwischen der Zielgruppe und dem Modell her, wobei zusätzlich zu Syntax und Semantik berücksichtigt wird, in welcher Weise die Zielgruppe die im Modell formulierten Aussagen interpretieren wird

Das Modell selbst besteht hierbei aus Aussagen, die in einer Modellierungssprache formuliert werden und die Anwendungsdomäne beschreiben. Die Modellierungssprache besteht aus allen Aussagen, die syntaktisch korrekt gebildet werden können. Hierzu verfügt die Sprache über eine Grammatik und ein Alphabet. Darüber hinaus verfügt die Sprache über eine spezielle Semantik, welche den in der Sprache geschaffenen Konstrukten eine Bedeutung verleiht. Mit der pragmatischen Ebene wird in dieser Arbeit auch sehr bald die Zielgruppe für die Beurteilung eines CM herangezogen. Die Interpretation der Zielgruppe besteht aus den Aussagen, von denen die Zielgruppe eines Modells ausgeht, dass sie im Modell enthalten sind. Der Begriff der Zielgruppe ist hierbei von den Autoren bewusst weit gefasst. So werden neben den

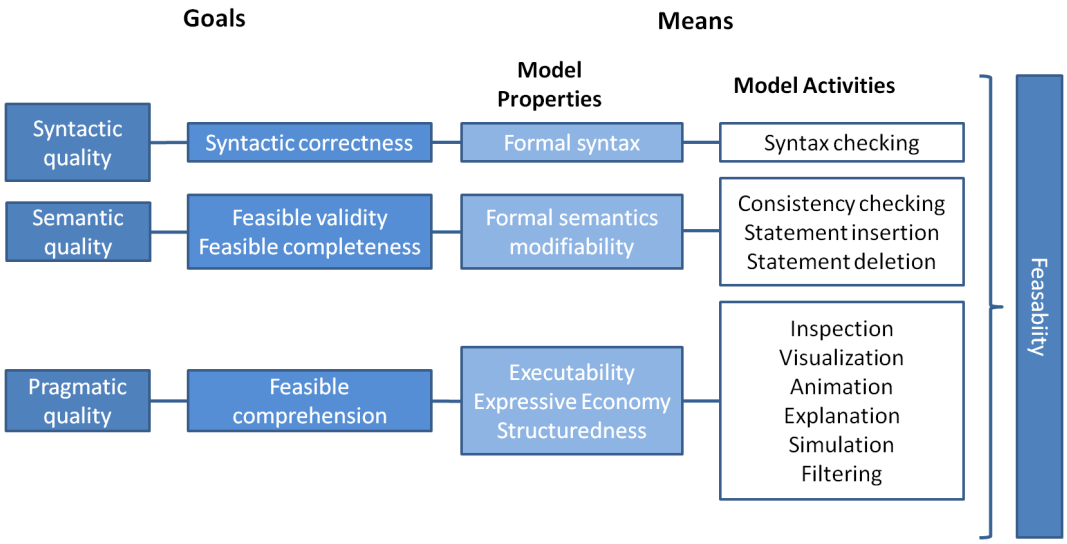

Abbildung 3.4: CM-Framework nach Lindland aus [142]

Nutzern eines Modells auch potentielle Nutzer, Kunden oder sogar andere Computersysteme inkludiert. Für eine gute Qualität eines CM ist es nicht ausreichend, dass dieses syntaktisch und semantisch korrekt gebildet wurde. Es muss auch von den Nutzern verstanden werden.

Das Modell steht mit jeder Ebene in einer bestimmten Beziehung. So steht das Modell mit der Modellierungssprache auf der Syntaxebene, mit der Anwendungsdomäne auf der Semantikebene und mit den Nutzern auf der Pragmatikebene in Beziehung. Die Anwendungsdomäne besteht dabei aus allen Aussagen, welche korrekt gebildet wurden und für die Problemlösung von Bedeutung sind. In der Nomenklatur von Lindland ist Domäne gleichbedeutend mit der idealen Kenntnis eines speziellen Problembereichs [142].

Das Framework nach Lindland trennt vor diesem Hintergrund, zwischen den Mitteln der Zielerreichung ('means') und den Zielen ('goals') selbst, welche für die drei Ebenen des Frameworks formuliert werden (vgl. Abbildung 3.4). Dabei ist zu beachten, dass streng zwischen Zielen und Maßnahmen, um ein Qualitätskriterium zu verbessern, unterschieden wird. Die vorgeschlagenen Methoden sind so aufgeführt, dass diese mit dem Ziel verbunden sind, das von ihnen direkt beeinflusst wird. Indirekt können auch andere Ziele beeinflusst werden. Darüber hinaus wird im Framework berücksichtigt, dass häufig eine vollständige Zielerreichung nicht oder nur mit einem unvertretbar hohem Aufwand möglich ist.

Auf syntaktischer Ebene wird als einziges Ziel 'syntaktische Korrektheit' verfolgt, was bedeutet, dass alle Aussagen des Modells bezüglich der Syntax wohlgeformt

sein müssen. Mittel um diese Ziele zu erreichen sind z.B. Fehlervermeidung und Fehlerkorrektur.

Auf semantischer Ebene werden die Ziele 'Validität' und 'Vollständigkeit' gefordert. Unter Validität wird verstanden, dass alle Aussagen im Model korrekt und relevant für die Problemstellung sind. Unter Vollständigkeit wird verstanden, dass alle korrekten und relevanten Aussagen der Anwendungsdomäne im Modell enthalten sind. Allerdings kann im Allgemeinen weder Vollständigkeit noch Validität erreicht werden, da die Anwendungsdomänen für eine vollständige und valide Erfassung zu komplex sind. Dementsprechend weicht Lindland diese beiden Kriterien auf und spricht in diesem Zusammenhang von 'erreichbarer Vollständigkeit' und 'erreichbarer Validität'. Diese Ziele werden durch Überprüfung der Aussagen im Modell und deren Korrektur erreicht. Dieser Vorgang ist mit Ausnahme der Konsistenzprüfung innerhalb einer formalen Sprache nicht automatisierbar und ist dementsprechend leichter zu identifizieren als Unvollständigkeit oder Ungültigkeit des Modells.

Das Hauptziel auf pragmatischer Ebene ist Verständlichkeit ('comprehension'). Ein wichtiger Punkt dabei ist, dass *alle* beteiligten Parteien und nicht nur einzelne Gruppen dazu in der Lage sind, das Modell zu verstehen. Dabei ist entscheidend, dass jede Partei den Teil des Modells versteht, der für sie selbst relevant ist. Jeder Systemteilnehmer muss also nicht das vollständige Modell, sondern nur den für ihn relevanten Teilbereich verstehen. Die Zielerreichung auf pragmatischer Ebene ist bei Lindland sehr vage formuliert. Zur Zielerreichung wird insbesondere Visualisierung und Erklärung vorgeschlagen, sowie die Unterstützung durch technische Hilfsmittel.

Bewertung

Das Framework von Lindland gilt als eines der grundlegenden Systeme, um die Qualität eines CM zu bewerten. Interessant ist hierbei die Erkenntnis, dass neben Syntax und Semantik die Verständlichkeit eines Systems in der Zielgruppe ein entscheidendes Qualitätskriterium ist. Darüber wird hinaus klargestellt, dass es sich bei einem Modell um Aussagen in einer Modellierungssprache handelt und diese Aussagen zur Qualitätsanalyse eines Modells betrachtet werden müssen. Es fehlt aber eine formale Vorgehensweise, die es ermöglicht den Inhalt der speicherbaren Aussagen eines Informationssystems oder Modells zu erfassen und mit einem anderen Modell zu vergleichen.

Die Qualitätskriterien des Frameworks sind in den Zielen formuliert und den einzelnen Ebenen zugeordnet. Insgesamt enthält dabei das Framework von Lindland mit der Beschränkung auf die vier Kriterien syntaktische Korrektheit, Vollständig-

keit, Validität und Verständlichkeit eine ausgesprochen geringe Anzahl an Kriterien. Diese bilden zwar tatsächlich wichtige Eigenschaften eines Datenstandards ab, sind aber unzureichend um diesen in geeignetem Maße zu bewerten. So werden Kriterien wie Flexibilität und Erweiterbarkeit auf semantischer Ebene und ein Kriterium für Einfachheit auf pragmatischer Ebene nicht betrachtet.

Die Kriterien sind nicht auf eine bestimmte Domäne oder Technologie beschränkt und somit generisch anwendbar. Allerdings sind die Kriterien nur benannt. Eine konkrete Möglichkeit der Messung und Implementierung der Kriterien wird nicht formuliert. So werden z.B. die Kriterien Vollständigkeit und Validität zwar beschrieben und definiert, trotzdem bleibt diese Definition so vage, dass diese in der Praxis nicht als Grundlage für ein Framework zur Evaluation von CM verwendet werden kann. Dies macht die Anwendung des Frameworks in der Praxis in Bezug auf konkrete Fragestellungen und somit auch auf den Bereich der Biodiversitätsinformatik unmöglich.

Damit lässt sich zusammenfassend sagen, dass das Framework von Lindland das grundlegende Vokabular für die Evaluation von konzeptuellen Modellen beschreibt und für die Theorie der Evaluation von Datenstandards einen wichtigen Beitrag leistet, aber keine konkreten Implementierungen für diese angeben kann.

3.2.2 Evaluation nach Moody

Vorstellung

Das Framework von Moody ist eines der wenigen und ersten Frameworks, welche nicht nur Kriterien benennt, sondern diese auch mit Kennzahlen messbar macht. Dabei bezieht sich Moody häufig direkt auf die Messung von Kennzahlen in ER-Diagrammen, wobei seine Methodik auch auf andere konzeptuelle Modelle übertragbar ist. Die Grundlagen seines Frameworks finden sich in [171], wobei seine vollständige Theorie erstmals in [168] gefunden werden kann. Diese Theorie wurde von Moody kontinuierlich weiterentwickelt und wurde auch in der Praxis überprüft (vgl. [169, 170, 172, 173]).

Grundlegend für das Verständnis von Moody´s Theorie ist das Problem der alternativen Datenmodelle [171]. Danach kann ein gegebenes Modellierungsproblem auf verschiedene Weise gelöst werden. Er begreift folglich Datenmodellierung nicht als die Aufgabe der Identifizierung des einen, korrekten Datenmodells, sondern als ein Prozess auf der Suche nach dem besten Ergebnis aus einer Reihe von möglichen Lösungen. Um das beste Ergebnis identifizieren zu können, werden objektive, exakte und verständliche Kriterien benötigt [171]. Andernfalls besteht die Gefahr, dass sich

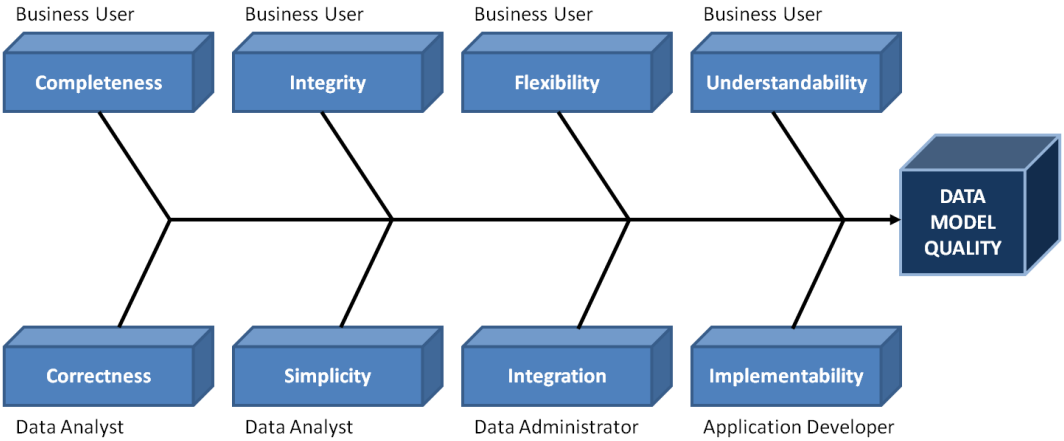

Abbildung 3.5: Kriterien nach Moody aus [168]

'ad hoc' nach Einzelmeinungen und gesundem Menschenverstand für ein bestimmtes Modell aus einer Reihe von Modellen entschieden wird. Um das beste Modell zu identifizieren und einen objektiven Entscheidungsprozess zu unterstützen, werden Kennzahlen ermittelt, welche aus dem Modell abgeleitet werden.

Das Framework nach [168] unterscheidet zwischen Kriterien, Metriken, Gewichtung und Verbesserungsstrategien. Kriterien stehen dabei für wünschenswerte Eigenschaften, die ein Datenmodell aufweisen sollte. Metriken dienen als konkrete und objektiv messbare Kennzahlen, um die Qualität eines Kriteriums zu messen. Die Gewichtung ist ein Maß für die Bedeutung eines Kriteriums in Vergleich zu den anderen Kriterien. Als Verbesserungsstrategien werden Methoden bezeichnet, deren Aufgabe es ist, die Qualität eines Datenmodells im Hinblick auf ein Kriterium oder mehrere Kriterien zu verbessern. Die Bedeutung der identifizierten Kriterien und die Qualität eines Datenmodells ist je nach Blickwinkel verschieden. Für den Anwender eines Systems ist beispielsweise das Kriterium der Implementierbarkeit eines Datenmodells von geringerer Bedeutung als für einen Entwickler. Um dem Rechnung zu tragen, integriert Moody die verschiedenen Perspektiven der Nutzer, Datenanalysten, Entwickler und Datenadministratoren in sein Framework (vgl. Abbildung 3.5). Moody verwendet in seinem Framework acht verschiedene Kriterien und gibt für jedes dieser Kriterien verschiedene Metriken (insgesamt 25) an, um die Qualität dieser zu messen. Somit kann einem Kriterium über verschiedene Wege eine Kennzahl zugeordnet werden. Eine Empfehlung zur Gewichtung der Kriterien oder Empfehlungen bezüglich der Anwendung der Metriken wird allerdings nicht gegeben.

Die Kriterien nach Moody sind im Einzelnen (vgl. Abbildung 3.5):

- Vollständigkeit: Existenz der Nutzeranforderungen im Modell
- Integrität: Durchsetzung von Geschäftsregeln
- Flexibilität: Anpassungsfähigkeit des Modells and Veränderungen
- Verständlichkeit: Aufwand das Modell zu Verstehen
- Korrektheit: Syntaktische Korrektheit in der Modellierungssprache und Einhaltung von Konventionen
- Einfachheit: Minimale Anzahl an Elementen zur Konstruktion
- Integration: Konsistenz des Modells mit anderen Daten der Organisation
- Implementierbarkeit: Technische Umsetzbarkeit

Im Folgenden werden die Kriterien Vollständigkeit, Flexibilität und Verständlichkeit eingehend besprochen, da diese im Kontext der Biodiversitätsinformatik besonders interessant sind. Prinzipiell sind zwar auch alle Kriterien im Kontext der Biodiversitätsinformatik relevant, können aber aus Platzgründen nicht ausführlich besprochen werden. Der interessierte Leser sei hierzu auf [168] verwiesen.

Vollständigkeit wird als das wichtigste aller Kriterien betrachtet, da bei seiner ungenauen oder unvollständigen Erfüllung das gesamte Datenmodell unabhängig davon wie gut die anderen Kriterien implementiert sind für die Nutzer nicht anwendbar ist. Das Vollständigkeitskriterium beschreibt, ob alle Nutzeranforderungen im Datenmodell enthalten sind. Dies könnte theoretisch durch den direkten Abgleich der Anforderungen der Nutzer mit dem Datenmodell erfolgen – wie es in [10] vorgeschlagen wird. Allerdings ist diese Vorgehensweise in der Praxis nur bedingt möglich, da die Anforderungen der Nutzer nicht direkt betrachtet werden können. Deshalb kann Vollständigkeit nach Moody nur in enger Zusammenarbeit mit den Nutzern bewertet werden. Es ist zu bemerken, dass im Vollständigkeitsbegriff nach Moody nicht nur die Vollständigkeit im Sinne von Lindland enthalten ist, sondern auch die Validität (vgl. [142] und Abbildung 3.6). Das Verständnis von Vollständigkeit nach Moody wird in Abbildung 3.6 dargestellt. In dieser Abbildung wird das Datenmodell mit den Nutzeranforderungen abgeglichen und es können verschieden Fehlerquellen identifiziert werden.

Hierbei wird nach Fehlern der 1.-4. Art unterschieden (vgl. Abbildung 3.6) unterschieden. Die Elemente, die im Datenmodell vorhanden sind, aber keiner Anforderung entsprechen, werden als Fehler 1. Art bezeichnet. Diese Elemente befinden

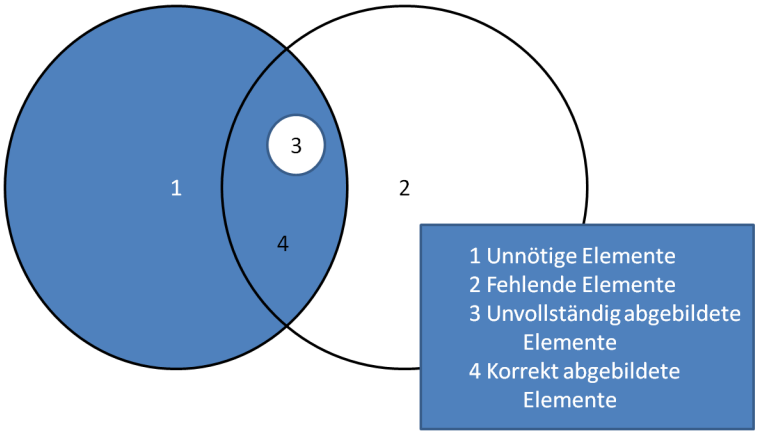

Abbildung 3.6: Vollständigkeitskriterium nach Moody aus [168]

sich außerhalb des Anwendungsbereichs des Systems und verursachen zusätzlichen Entwicklungs- und Wartungsaufwand. Auf der anderen Seite gibt es Nutzeranforderungen die bei der Modellierung des Datenmodells nicht berücksichtigt wurden. Diese werden als Fehler 2. Art bezeichnet und können später – falls überhaupt möglich – nur mit hohem Aufwand in das System integriert werden. Schlimmstenfalls werden diese überhaupt nicht identifiziert, was zu einer Ablehnung des Systems durch die Nutzer führt. Als Fehler dritter Art werden Elemente bezeichnet, die nur unvollständig einer Nutzeranforderung entsprechen oder aber unpräzise formuliert sind. Fehler 3. Art führen analog zu Fehlern 2. Art zu Anpassungen im laufenden System oder zu mangelnder Akzeptanz bei den Nutzern. Der in Abbildung 3.6 mit '4' gekennzeichnete Bereich stellt letztlich den Bereich der korrekt im Datenmodell umgesetzten Nutzeranforderungen dar. Ziel einer Analyse der Vollständigkeit ist es stets Fehler erster zweiter und dritter Art zu eliminieren. Zur Messung der Qualität eines Datenmodells schlägt Moody deshalb vor die Anzahl der Fehler der 1., 2. und 3. Art direkt als Kennzahlen zu verwenden. Die Fehler 4. Art sind durch die Anzahl der Inkonsistenzen mit dem Prozessmodell definiert. Dazu wird das Datenmodell auf die korrespondierenden Geschäftsprozesse abgebildet und das Ergebnis in Form einer CRUD (create, read, update, delete) Matrix dargestellt. Die Analyse dieser Matrix kann dazu verwendet werden, um Lücken und überflüssige Elemente zu identifizieren.

Flexibilität stellt die Robustheit und Anpassungsfähigkeit des Modells gegenüber zukünftigen Veränderungen dar. Die Zielvorgabe bei diesem Kriterium ist, dass zukünftige Veränderungen mit einem Minimum an Aufwand in das Modell integriert werden können. Es stellt damit einen wichtigen Faktor für die Flexibilität des Gesamtsystems dar. Mängel in diesem Bereich wirken sich negativ auf die Gesamt-

organisation aus. Flexibilität ist dabei recht schwierig zu bewerten, da eine sichere Evaluation dieses Kriteriums Kenntnisse über die Zukunft voraussetzt. Dementsprechend sind die Metriken bei Moody recht vage und unpräzise gehalten. Die erste Metrik wird mit der Anzahl der Elemente, die sich in der Zukunft ändern werden, angegeben. Die zweite Metrik beziffert die geschätzten Kosten der Anpassung als Kennzahl. Eine weitere Metrik wird durch das Rating von Anwendern gegeben, von welcher strategischer Bedeutung eine Veränderung in dem Modell ist [168].

Die Verständlichkeit eines Modells ist aus zwei verschiedenen Sichten von Bedeutung. Zum einen müssen Anwender das Modell verstehen, um zu verifizieren, dass es ihren Anforderungen genügt. Auf der anderen Seite müssen Entwickler das Modell zur korrekten Implementierung verstehen. Moody schlägt zur Messung der Verständlichkeit Ratings durch Anwender und das Ausführen von Testszenarien mit Nutzern und Anwendern vor und führt dies als Metrik ein.

Auch für die anderen Kriterien gibt Moody verschiedene Metriken an. Diese beziehen sich dabei teilweise direkt auf relationale Datenbanken wie die Anzahl der Entitäten und Relationen in einem ER-Diagramm oder der Normalisierungsgrad einer Datenbank.

Bewertung

Moody identifiziert in seinem Framework verschiedene Nutzerperspektiven. Die Kriterien an sich sind gut gewählt und lassen sich den Qualitätsdimensionen von Lindland zuordnen. Moody gibt dabei für seine acht Kriterien 25 verschiedenen Metriken vor und schlägt Methoden zur Bestimmung dieser vor, die allerdings recht vage formuliert sind.

Dabei liefert das Vollständigkeitskriterium einen entscheidenden Beitrag zur Theorie der Evaluation konzeptueller Schemata. Die Kriterien sowie die Fehler erster, zweiter und dritter Art sind dabei für die Messung der Qualität eines Schemas sachgemäß und relevant. Mit der Einführung der Flexibilität eines Schemas wird der Umstand, dass Schemata einem steten Wandel unterworfen sind, mit in das Modell aufgenommen. Mit dem Kriterium der Verständlichkeit wird auch auf die Bedürfnisse der Nutzer geachtet. Allerdings scheinen einige Kriterien zumindest auf sprachlicher Ebene sich gegenseitig zu umfassen. So entsteht durch die Verwendung der Begriffe Integrität und Korrektheit, sowie Einfachheit und Verständlichkeit zumindest ein gewisses Verwechslungspotential.

Die Kriterien an sich sind nicht auf relationale Datenbanken an sich beschränkt und können damit generisch verwendet werden. Dies trifft aber nicht auf die Imple-

mentierung aller Kriterien zu. Diese sind im Allgemeinen zu vage formuliert, um von praktischen Nutzen zu sein und beziehen sich häufig direkt auf relationale Datenbanken (wie z.B. der Normalisierungsgrad einer Datenbank bei Korrektheit). Moody wendet bei der Implementierung vieler seiner Kriterien die Interaktion mit den verschiedenen Nutzern des modellierten Systems in Form von Befragungen und Tests mit diesen an. So ist eine Kennzahl zur Messung der Verständlichkeit die Anzahl der Fehler durch die Anwender beim Befüllen eines Modells mit Daten. Eine entsprechende Kennzahl ist damit von der Anzahl der ausgeführten Tests und auch der Erfahrung der Anwender abhängig und kann nur bedingt zum Vergleich der Qualität von Datenmodellen herangezogen werden. Die so gebildeten Kennzahlen sind selbstverständlich auch auf andere Arten der Modellierung übertragbar. Ein Gegenbeispiel stellt die Implementierung der Korrektheit dar, in welchem der Normalisierungsgrad der Datenbank als Kriterium herangezogen wird. Bei der Analyse der Metriken lässt sich als Ergebnis festhalten, dass die Metriken zur Vollständigkeit, Flexibilität, Verständlichkeit und Implementierbarkeit generisch implementiert sind und auch in anderen Bereichen verwendet werden können. Die konkrete Implementierung der Einfachheit gibt als Metriken die Anzahl von Entitäten und Relationen im ER-Modell an. Diese ist nicht direkt auf XSD übertragbar, es konnten aber in Abschnitt 3.1.2 analoge Konzepte identifiziert werden.

Darüber hinaus sind auch die Metriken zur Messung der Kriterien nur bedingt geeignet. So ist es fraglich, ob die Anzahl der Entitäten in einem Schema tatsächlich ein adäquates Maß für die Einfachheit ist. Für die Vollständigkeit gibt Moody als Metrik die Fehler der ersten bis vierten Art an, allerdings keine Möglichkeit diese in der Praxis tatsächlich zu messen. Darüber hinaus lassen die so gebildeten Kennzahlen auch nur unzureichende Schlüsse auf die Qualität eines Datenmodells zu. So repräsentiert ein Datenmodell, bei dem eine zentrale Anforderung nicht modelliert wurde mit Sicherheit die Nutzeranforderungen schlechter als ein Datenmodell in dem 15 weniger wichtige Anforderungen nicht abgebildet wurden. Trotzdem würde die Metrik, welche die Anzahl der Fehler der 1. Art als Kennzahl ausweist, das erste System deutlich besser bewerten. Zusammenfassend lässt sich also im Bezug auf das Vollständigkeitskriterium sagen, dass von Moody zwar die Bedeutung des Kriteriums an sich korrekt erkannt wurde, er allerdings in seinem Framework keine Methode findet, um die Qualität des Kriteriums in zufriedenstellender Weise zu messen. Dies gilt auch für die anderen Kriterien in der Weise, dass das gefundene Kriterium an sich sinnvoll gewählt ist, die Methode der Messung aber zumindest fragwürdig erscheint und die Übertragung der Metriken auf die Biodiversitätsinformatik nicht möglich

oder sinnvoll ist.

In Bezug auf die Biodiversitätsinformatik sind die Kriterien der Vollständigkeit und Flexibilität von entscheidender Bedeutung. Die direkte Anwendung des Frameworks nach Moody auf die Biodiversitätsinformatik ist allerdings praktisch unmöglich. So müsste zur Messung des Vollständigkeitskriteriums die Domäne der Biodiversitätsforschung exakt definiert und vollständig verstanden sein – was nicht der Fall ist. Auch die Flexibilität eines Schemas kann mit einer Metrik wie dem Erwartungswert der Änderungen (Metrik 7 aus [168]) oder den erwarteten Kosten der Anpassung des Datenmodells (Metrik 8 aus [168]) in einer offenen Umgebung nicht in geeigneter Form bestimmt werden. Gleiches gilt auch für die Evaluation der Verständlichkeit, bei welcher Moody auf die Evaluation der Nutzer oder aber direkte Tests des Nutzerverständnisses zurückgreift.

Damit ist die konkrete Implementierung der Kennzahlen für die Biodiversitätsinformatik ungeeignet, da diese zu oberflächlich beschrieben oder nicht anwendbar sind. Das Framework nach Moody leistet aber einen wichtigen Beitrag zur Evaluation konzeptueller Schemata, da in diesem Methoden zur Messung von Kriterien angegeben wurden und wichtige Kriterien erstmals formuliert wurden.

3.2.3 Das Conceptual Modeling Quality Framework (CMQF)

Vorstellung

Mit dem 'Conceptual Modeling Quality Framework' (CMQF) wird in [181] der Versuch gemacht, die beiden grundlegenden Ansätze zur Bewertung von konzeptuellen Modellen aus [142] (in der Erweiterungsform aus [131]) und [258] zu vereinen. Während das Framework nach Lindland (LSS) [142] das Hauptaugenmerk auf die Qualität des Datenmodells legt, wird in Wand (BWW) [258] der Prozess der Modellerstellung in den Mittelpunkt gestellt. Nach [181] tragen beide Ansätze entscheidend zur Qualität eines konzeptuellen Modells bei. Somit muss der Prozess der Modellerstellung auch nach seiner positiven Auswirkungen auf die Modellqualität bewertet werden und umgekehrt [181]. CMQF begreift sich eher als eine theoretische Grundlage für die weitere Erforschung der Qualität von konzeptuellen Modellen bzw. als Grundlage für die Erstellung von Evaluationssystemen in der Praxis. Eine konkrete Implementierung des Frameworks wurde noch nicht publiziert.

Aus LSS wird der zentrale Gedanke übernommen, dass in einem Modell Aussagen aus der Realität abgebildet werden müssen und die Qualität eines Modells maßgeblich davon abhängig ist, wie gut diese Aussagen im Modell repräsentiert sind. In der Erweiterung von LSS [131] wurden die ursprünglichen Kriterien um Kriterien wie die

'Geeignetheit der Modellierungssprache' und die 'Kenntnis des Modellierers von der Anwendungsdomäne' hinzugefügt. Diese werden für die Evaluation eines Modells auf sechs verschiedenen Ebenen (sozial, pragmatisch, semantisch, syntaktisch, empirisch, physikalisch) betrachtet.

Auf der anderen Seite steht der Ansatz aus [258] in dem die Bildung eines konzeptuellen Modells als Prozess begriffen wird. Dieser wurde ebenfalls erweitert (siehe [257, 259]). In diesem wird der Prozess der konzeptuellen Modellierung ausgehend von der realen Welt als eine Reihe von Transformationen betrachtet, welche in einem Informationssystem seinen Abschluss findet. Das Framework nach Wand beginnt dabei in der Anwendungsdomäne, die von den Analysten wahrgenommen wird. Diese Wahrnehmung dient wiederum als Grundlage für den Prozess zur Erstellung eines Informationssystems. Dieses wird dann von den Anwendern wiederum wahrgenommen und mit der Wahrnehmung des Anwenders von der Domäne abgeglichen. Diese spezielle Wahrnehmung ist durch die individuellen Bedürfnisse des Anwenders gekennzeichnet [257]. Damit wird in dem Framework nach Wand die Ausdrucksstärke der Modellierungssprache und deren Grammatik, sowie die Unterschiede zwischen dem erstellten Modell und der Sicht der Nutzer herangezogen.

Die Synthese aus dem LSS und BWW Framework findet auf zwei Ebenen statt. Zum einen vertikal durch die Trennung von physischer und sozialer Realität und zum anderen horizontal durch Trennung von Anwendungsdomäne, Bezugssystem, Modellierungssprache und konzeptuelle Repräsentation. Daraus ergeben sich acht Eckpfeiler, die wie in Abbildung 3.7 dargestellt werden. Die Konstrukte auf der linken Seite korrespondieren dabei mit der physischen Realität wohingegen die Konstrukte auf der rechten Seite der sozialen Realität zugeordnet sind. Die Eckpfeiler umfassen dabei alle Elemente aus dem LSS und dem BWW Framework, wobei diese teilweise implizit enthalten sind vgl. [181].

Die Eckpfeiler stehen dabei in verschiedener Weise miteinander in Beziehung, welche verschiedenen Ebenen zugeordnet sind. So wird zwischen der physikalischen, der Wissens-, der Lern- und der Entwicklungsebene unterschieden. Die Unterteilung in diese Ebenen soll dem Prozessgedanken der konzeptuellen Modellierung Rechnung tragen und diesem folgen (vgl Abbildung 3.8).

Die Qualität der Beziehungen wird dabei als Maß für die Qualität des Modells herangezogen. So definieren die Beziehungen auf der physikalischen Ebene sieben verschiedenen Qualitätskriterien, die in Abbildung 3.8 mit P1-P7 dargestellt sind. Diese Kennzeichnung wird auf den anderen Ebenen analog verwendet. Zu beachten ist, dass man, wenn man die Ebenen aus Abbildung 3.8 übereinander legt, genau die

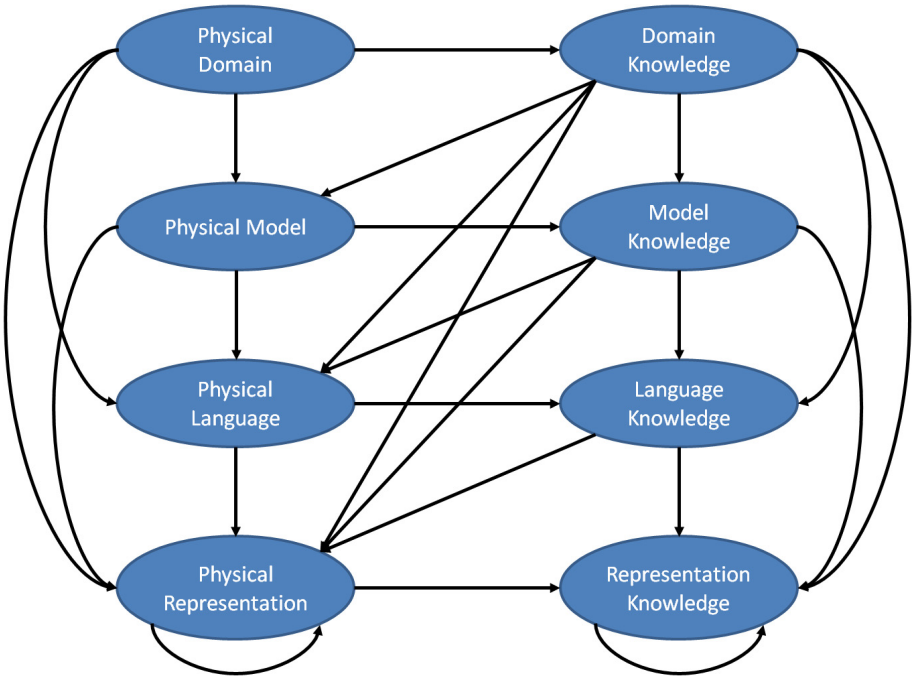

Abbildung 3.7: Eckpfeiler des CMQF nach [181]

Beziehungen erlangt, wie diese in Abbildung 3.7 dargestellt werden. Eine vollständige Liste dieser Qualitätsmerkmale mit den assoziierten Eckpfeilern findet sich in Tabelle 3.1.

Das Framework nach [181] definiert dabei nur diese Kriterien, gibt aber keine Implementierung oder Möglichkeit der konkreten Messung an. Exemplarisch sei hier aufgrund der Bedeutung das Kriterium 'P4-Semantische Qualität' herangezogen. Dieses wird in [181] als die Anforderung definiert, dass die endgültige Repräsentation die Bedeutung der Anwendungsdomäne exakt und vollständig erfassen muss und die Beschränkungen der Modellierungsaufgabe beachtet werden. Diese Definition des Kriteriums ist sachgemäß für die Qualitätsbewertung der semantischen Qualität des Modells. Die Definition allein ist aber ohne eine konkrete Möglichkeit der Messung wenig hilfreich.

Bewertung

Mit CMQF wird ein umfassender Kriterienkatalog spezifiziert, der neben dem Modell an sich auch den Prozess der Modellerstellung und Evolution berücksichtigt. Dadurch werden der Bewertung von Modellen weitere entscheidende Aspekte hinzugefügt. Die Qualität der Beziehungen zwischen den Eckpfeilern des Frameworks wird dabei als Kriterium herangezogen. Dies ist ein interessanter Ansatz, der eine umfassende Ana-

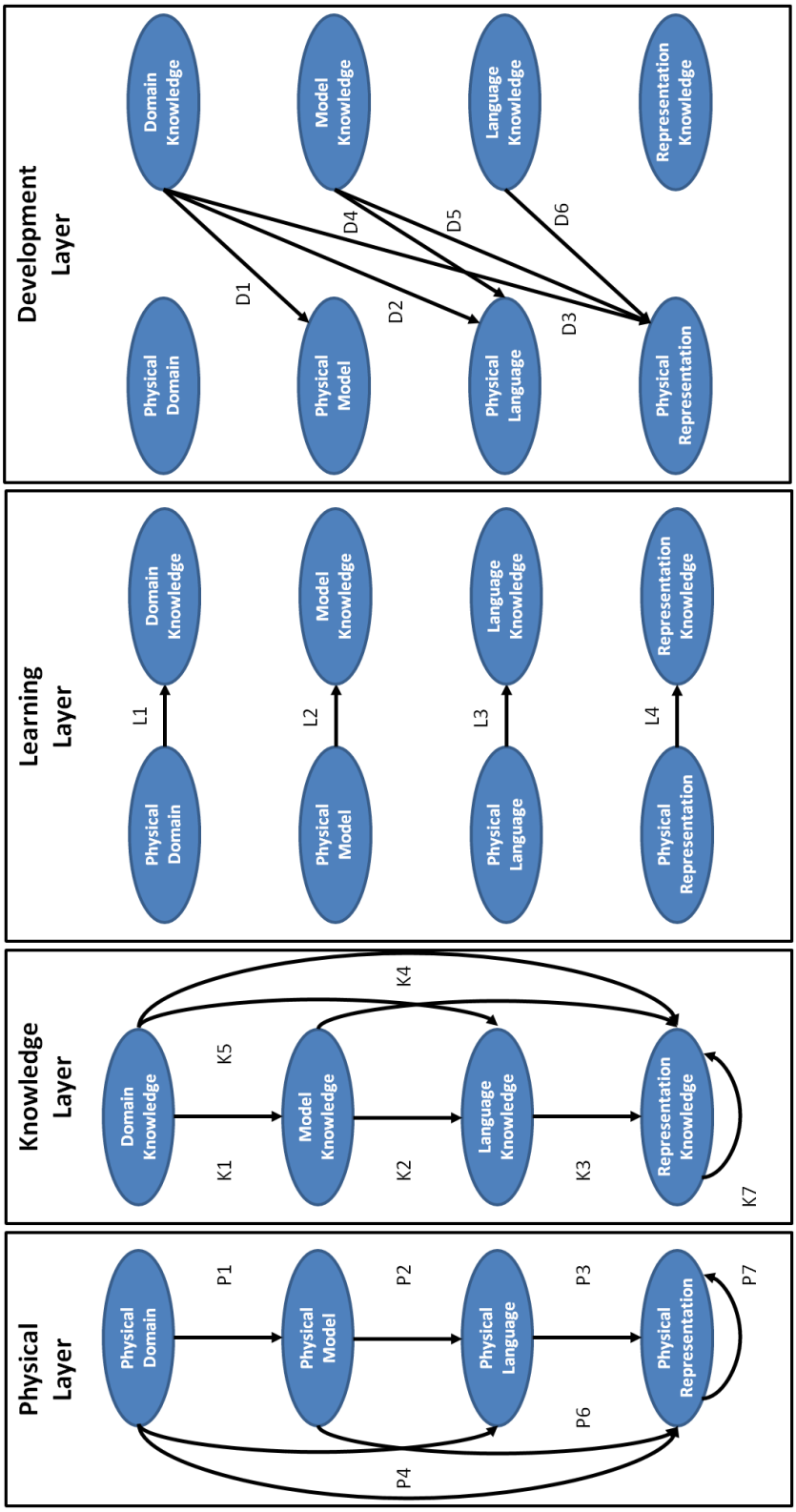

Abbildung 3.8: Ebenen von Beziehungen des CMQF nach [181]

Label	Quality type	Quality reference	Object of interest
P1	Model-domain appropriateness	Physical domain	Physical model
P2	Ontological quality	Physical model	Physical language
P3	Syntactic quality	Physical language	Physical representation
P4	Semantic quality	Physical domain	Physical representation
P5	Language-domain appropriateness	Physical domain	Physical language
P6	Intensional quality	Physical model	Physical representation
P7	Empirical quality	Physical representation	Physical representation
K1	Perceived model-domain appropriateness	Domain knowledge	Model knowledge
K2	Perceived ontological quality	Model knowledge	Language knowledge
K3	Perceived syntactic quality	Language knowledge	Representation knowledge
K4	Perceived semantic quality	Domain knowledge	Representation knowledge
K5	Perceived language-domain appropriateness	Domain knowledge	Language knowledge
K6	Perceived intensional quality	Model knowledge	Representation knowledge
K7	Perceived empirical quality	Representation knowledge	Representation knowledge
L1	View quality	Physical domain	Domain knowledge
L2	Pedagogical quality	Physical model	Model knowledge
L3	Linguistic quality	Physical language	Language knowledge
L4	Pragmatic quality	Physical representation	Representation knowledge
D1	Applied domain-model appropriateness	Domain knowledge	Physical model
D2	Applied domain-language appropriateness	Domain knowledge	Physical language
D3	Applied domain knowledge quality	Domain knowledge	Physical representation
D4	Applied model-language appropriateness	Model knowledge	Physical language
D5	Applied model knowlege quality	Model knowledge	Physical language
D6	Applied language knowledge quality	Language knowledge	Physical language

Tabelle 3.1: Qualitätsmerkmale mit assoziierten Eckpfeiler des CMQF aus [181]

lyse eines Modells erlaubt. Das Framework ist dabei generisch formuliert, so dass die Kriterien in allen Bereichen der konzeptuellen Schemaevaluation angewendet werden können. Allerdings sehen die Autoren selbst CMQF aktuell als eine theoretische Arbeit und führen keine Implementierung der Kriterien aus. So wird für keines der insgesamt 24 Qualitätsmerkmale eine Methode zur Messung der Qualität oder gar der Bestimmung einer Kennzahl angegeben. Eine direkte Anwendbarkeit des Frameworks in der Biodiversitätsinformatik ist deshalb nicht möglich.

3.2.4 Evaluation von XML-Schemata

Vorstellung

Auch wenn das W3C bereits seit 2001 XSD zur Strukturierung von XML empfiehlt [251], wurde noch keine einheitliche Theorie zur Evaluation von XML-Schemata entwickelt. Deshalb wird im Folgenden ein kurzer Überblick über Maße gegeben, die sich speziell mit der Evaluation der Struktur von XML-Dokumenten befassen. Einer der ersten Ansätze ist dabei in [158] zu finden, in welchem Grundlagen aus [125] für DTD aufgegriffen wurden. In diesen Arbeiten werden wie auch in [168] Kennzahlen definiert, mit Hilfe derer die Qualität eines XSD-Schemas bestimmt werden soll. In [125] werden insbesondere Maße zur Komplexitätsmessung eingeführt. Dazu gehört die Zeilenanzahl des Schemas (LOC) und Maße welche die Komplexität der Graphstruktur eines XML-Dokuments messen (z.B. Strukturtiefe, Fan-In, Fan-Out). Dieser Ansatz wird in [158] dahingehend fortgeführt, dass die Anzahl von verschiedenen Strukturen eines XML-Schemas als Kennzahl herangezogen wird, wie z.B. die Anzahl der Deklarationen von komplexen und einfachen Typen und die Anzahl der abgeleiteten Typen. Die so insgesamt 11 identifizierten Kennzahlen werden gewichtet und zu einer Kennzahlzahl – dem Komplexitätsindex– zusammengefasst.

Auch weitere Arbeiten wie [143] und [248] beschäftigen sich vorrangig mit der Komplexität der Graphstruktur des XML-Schemas oder äußeren Kennzahlen wie LOC. Fortführungen dieser Arbeiten wie [8] und [9] zielen primär darauf ab, die Komplexität eines XML-Schemas zu erfassen. Eine inhaltliche Auseinandersetzung mit der Semantik der zu speichernden Daten findet in diesen Arbeiten nicht statt.

Allerdings wurden in Arbeiten [38, 150] 'Best Practices' für die Erstellung von XML-Schemata publiziert. Dabei werden typische Architekturen für XML-Schemata vorgestellt und evaluiert. Die Architekturtypen heißen 'Russian Doll', 'Salami Slice', 'Venetian Blind' und 'Garden of Eden'. Für eine genaue Definition der Typen sei der interessierte Leser auf [150] verwiesen. Im Allgemeinen werden 'Garden of Eden' und 'Venetian Blind' den anderen Strukturen als überlegen betrachtet (so empfiehlt

[150] die Verwendung von 'Garden of Eden').

Bewertung

Die speziellen Theorien zur Evaluation von XML-Schemata sind zur Evaluation von Datenstandards alleine wenig geeignet, da sich diese ausschließlich auf die Struktur des Schemas beziehen und lediglich Maße für die Komplexität des Schemas angeben. Damit finden die Kriterien aus dem XML-Umfeld primär Entsprechungen in der Einfachheit in den Theorien der Bewertung von konzeptuellen Modellen. Kriterien wie Vollständigkeit oder Flexibilität. werden nicht untersucht. Darüber hinaus sind die Frameworks zur XML-Evaluation sind nicht generisch formuliert. Sie lassen sich deshalb am besten innerhalb eines der Frameworks, die in den Abschnitten 3.2.1 bis 3.2.3 vorgestellt wurden, z.B. als eine Metrik zur Bestimmung der Einfachheit eines XML-Dokuments anwenden. Eine direkte Anwendung in der Biodiversitätsinformatik ist nicht möglich.

3.3 Zusammenfassung

Im diesem Kapitel wurde gezeigt, dass in der Literatur zur Bewertung konzeptueller Schemata insbesondere eine Vielzahl an Kriterien zu finden ist, die auch zur Bewertung von Standards in der Biodiversitätsinformatik herangezogen werden können. Allerdings bleiben die wichtigsten Ansätze wie das CMQF [181] oder LSS [142] eine Implementierung dieser Kriterien schuldig. Lediglich das Framework von Moody [168] unternimmt einen Versuch, den einzelnen Qualitätsmerkmalen auch eine Kennzahl zuzuordnen. Allerdings ist auch bei diesem Ansatz die Methode der Kennzahlbestimmung so vage formuliert, dass diese in der Praxis nicht angewendet werden können. Somit konnte keines der vorgestellten Frameworks die Kriterien an ein Evaluationsframework für Datenstandards in der Biodiversitätsinformatik erfüllen (siehe Tabelle 3.2).

Insgesamt hat die Literaturrecherche ergeben, dass keines der betrachteten Frameworks zu einer Analyse von Datenstandards in der Biodiversitätsinformatik geeignet ist. Es konnten allerdings im Bereich der Evaluation von konzeptuellen Modellen Kriterien wie Vollständigkeit, Flexibilität und Verständlichkeit identifiziert werden, welche für die Evaluation eine Datenstandards entscheidende Merkmale darstellen. Insbesondere das Kriterium der Vollständigkeit konnte als besonders wichtig identifiziert werden, da dieses letztlich über die Anwendbarkeit eines Datenstandards entscheidet.

	LSS	Moody	CMQF	XML
Kriterienauswahl	gut	sehr gut	gut	schlecht
Generizität der Kriterien	ja	ja	ja	nein
Implementierung	nein	ja, aber unzureichend	nein	ja, aber nur Spezialfälle
Generizität der Implementierung	N/A	teilweise	N/A	nein

Tabelle 3.2: Ergebnis der Evaluation der Frameworks zur Qualitätsmessung von konzeptuellen Modellen

In den untersuchten Frameworks fehlen bestimmte Kriterien, die zur Bewertung von Datenspeichersystemen herangezogen werden sollten. Dies liegt daran, dass Datenstandards in der Biodiversitätsinformatik primär zum Datenaustausch eingesetzt werden. Daraus ergeben sich Kriterien wie die Garantie der Eindeutigkeit und Aktualisierbarkeit von Datensätzen, sowie die Vermeidung von Redundanzen. Letzteres tritt in der Praxis insbesondere in XML-Datenstandards wie ABCD (siehe Abschnitt 4.4.1) auf. Hierbei werden die Felder zur Wahrung von geistigen Eigentum in ABCD (Intelectual Property Rights - IPR) bei jedem Datensatz mehrfach mitgeführt, obwohl die Informationen im Allgemeinen identisch sind. Dies erzeugt zusätzliches Datenaufkommen und sollte deshalb vermieden werden. Darüber hinaus müssen einmal übertragene Daten aktuell gehalten werden. Ein Beispiel für das Problem der Aktualität von Daten ist z.B. der Wandel von taxonomischen Bezeichnungen im Laufe der Zeit oder ganz plastisch die Aktualität von Telefonnummern. Ein weiteres Kriterium für die Verwendung einer Infrastruktur ist die Unterstützung von Data Provenance (siehe Abschnitt 6.1). Über diese wird die Herkunft von Datensätzen und Veränderungen an diesen gespeichert. Auch die Unterstützung von Data Provenance wird von den untersuchten Frameworks nicht berücksichtigt. Dementsprechend ist es erforderlich eine neues Frameworks zur Evaluation von Datenstandards zu entwickeln. Ein neues Framework, welches auf Basis von Prozessen arbeitet, wird in Kapitel 4 vorgestellt. Dieses verwendet mit der 'Process Oriented Schema Evaluation' (POSE) eine neue Methode zur Messung der Vollständigkeit.

Kapitel 4

Evaluation von Datenstandards

Im folgenden Kapitel werden wichtige Datenstandards der Biodiversitätsinformatik nach einem selbst entwickelten Framework zur Messung der Qualität von Datenstandards evaluiert. Kern dieses Frameworks ist die Messung der Vollständigkeit eines Datenstandards mit dem 'Process Oriented Schema Evaluation' (POSE), welches eine Evaluation auf Basis von Prozessen in der Anwendungsdomäne ausführt. Dazu ist zunächst ein Exkurs in die Prozessmodellierung notwendig (Abschnitt 4.1), in welchem die 'perspektivenorientierte Prozessmodellierung' (POPM) eingeführt wird. Anschließend wird mit der 'prozessorientierte Schemaevaluation' (POSE) in Abschnitt 4.2 ein Framework zur Evaluation der Vollständigkeit von konzeptuellen Schemata als auf der Basis von Prozessen vorgestellt. POSE stellt eine eigene Entwicklung dar und löst die Probleme der in Kapitel 3 vorgestellten Frameworks in Bezug auf die Implementierung von Kriterien zur Messung der Vollständigkeit. POSE wird in Abschnitt 4.3 mit weiteren Kriterien dazu verwendet, um ein Evaluationsframework für Datenstandards in der Biodiversitätsinformatik zu formulieren. Dieses wird in Abschnitt 4.4 auf ausgewählte Datenstandards der Biodiversitätsinformatik angewendet.

4.1 Perspektivenorientierte Prozessmodellierung

Die Erfassung von Daten ist stets das Ergebnis eines Prozesses der Datenaufnahme. So entstehen im Rahmen der Biodiversitätsinformatik die Daten nicht einfach durch das Befüllen von Datenstrukturen, sondern werden beispielsweise in Begehungen erhoben. In Rahmen einer Begehung analysiert ein Biologe die Vegetation einer Wiese und erstellt dadurch eine Artenliste, welche aus den taxonomischen Bezeichnungen der identifizierten Fundobjekte besteht (vgl. Abschnitt 2.2.4).

Für die Modellierung von Prozessen ist die 'perspektivenorientierte Prozessmo-

dellierung' (POPM) eine etablierte und bewährte Methode. Im Rahmen des POPM nach [113, 114] wird ein Prozess durch verschiedene Perspektiven definiert. Soll ein Prozess modelliert werden, werden insbesondere Antworten auf die Fragen Was?, Wer?, Womit?, Wie? und Wann? gesucht.

Die Antworten auf diese Fragen werden in Form von Perspektiven in das Modell eingebracht. Die Perspektiven stehen dabei orthogonal zueinander, das heißt, dass sich die Informationen der Perspektiven nicht überlappen. Im Rahmen des POPM nach [114] wurden dabei folgende Basisperspektiven definiert (vgl. Abbildung 4.1):

- Funktionale Perspektive (Was?): Beschreibt die funktionalen Einheiten eines Prozesses, die ausgeführt werden sollen. Diese Perspektive bildet den Kern eines Prozessmodells und stellt damit die Struktur in einem Prozessmodell, welche sich aus der Prozesshierarchie ergibt.

- Datenorientierte Perspektive (Womit?): Beschreibt, wo innerhalb eines Prozesses Daten erzeugt oder aber konsumiert werden. Im Rahmen dieser Perspektive können neben Daten, die in Dokumenten erfasst werden, auch physische Erzeugnisse verstanden werden. So kann im Rahmen einer Prozessmodellierung in einem Krankenhaus auch eine Blutprobe als Datum aufgefasst werden.

- Organisatorische Perspektive (Wer?): Beschreibt, wer für die Ausführung des Prozesses verantwortlich ist. Dies muss nicht zwangsläufig eine natürliche Person sein, sondern kann auch eine Organisation oder aber eine Maschine sein [26].

- Operationale Perspektive (Wie?): Beschreibt die Werkzeuge, die bei der Ausführung eines Prozesses verwendet werden. Dies können bei der Felddatenerfassung Programme wie DiversityMobile oder aber auch traditionelle Werkzeuge wie Zettel und Stift sein.

- Verhaltensorientierte Perspektive (Wann?): Diese Perspektive legt fest, in welcher Reihenfolge die Prozesse innerhalb eines Prozessmodells ausgeführt werden sollen.

Die Auflistung der Perspektiven ist nicht abschließend. So können je nach den Modellierungsanforderungen der Domäne weitere Perspektiven wie z.B. die Kausalitätsperspektive (Warum?) verwendet werden. In der Biodiversitätsinformatik erweist sich dabei die Ergänzung von POPM durch eine lokale Perspektive (Wo?) als sinnvoll. Für eine umfassende Beschreibung von POPM sei auf [114] verwiesen.

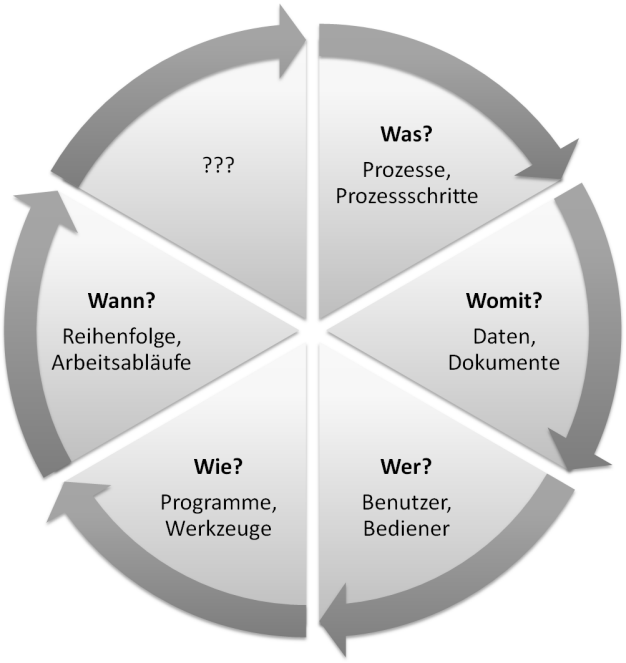

Abbildung 4.1: Darstellung der Perspektiven nach [250]

Die Anwendung von POPM kann sehr gute Ergebnisse in der praktischen Anwendung aufweisen. So wurde POPM häufig zur Modellierung von Prozessen in Krankenhäusern herangezogen und auch an die speziellen Anforderungen dieser Domäne angepasst [68]. Die praktischen Erfahrungen bei der Prozessmodellierung im Klinikum Fürth, Bayreuth und Erlangen haben gezeigt, dass entsprechende Prozessmodelle auch von Domänenexperten der Anwendungsgebiete gut verstanden werden, so dass die Verwendung dieser Art der Prozessmodellierung insbesondere auch im Rahmen der Biodiversitätsinformatik besonders gut geeignet ist.

Der folgende Anwendungsfall gibt eine typischen Prozess der Arbeit eines Kartierers wieder:

Prozess der Geländekartierung: Ein Kartierer ist auf der Suche nach biologischen Objekten z.B nach Pilzen in einem zuvor definierten Gebiet. Hat er einen Pilz gefunden, dokumentiert er die taxonomische Bezeichnung des Fundes sowie den Ort und Zeitpunkt der Kartierung. Der Prozess der Geländekartierung gestaltet sich dabei wie in Abbildung 4.2 beschrieben und kann durch Programme wie Diversity-Mobile unterstützt werden.

Dieser Prozess ist in der Biodiversitätsforschung sehr häufig anzutreffen. Der Prozesse selbst besteht aus mehreren Arbeitsschritten. Diese sind im unteren Bereich von Abbildung 4.2 separat aufgeführt. Diese Subprozesse können in einer flexiblen

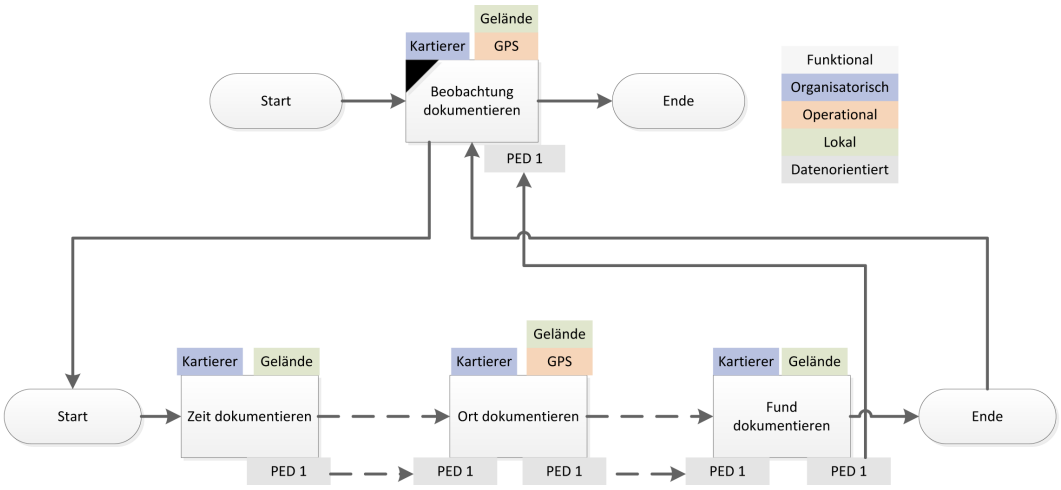

Abbildung 4.2: Prozess der Geländekartierung in der Biodiversitätsforschung

Reihenfolge ausgeführt werden. Die flexible Ausführung von Prozessen ist in Abbildung 4.2 in der Notation nach [108] mit unterbrochenen Pfeilen gekennzeichnet. In der datenorientierten Perspektive müssen die Ergebnisse der Prozessschritte 'Zeit dokumentieren', 'Ort dokumentieren' und 'Fund dokumentieren' erfasst werden. Dafür wird ein Erfassungsdokument angelegt und schrittweise um die Ergebnisse der Prozessausführung ergänzt. Erfassungsdokumente werden auch in anderen Prozessen zum Dokumentieren der Prozessergebnisse benötigt. Deshalb werden diese im Folgenden Allgemein als 'ProcessExecutionDocuments' (PED) bezeichnet. Die PED´s sind für eine bestimmte Art von Prozessen spezifisch. In Abbildung 4.2 wird das spezifische PED für den Prozess 'Beobachtung dokumentieren' mit 'PED 1' Bezeichnet. Das Ergebnis des Gesamtprozesses 'Beobachtung dokumentieren' ist damit ein 'PED 1', welches die erfassten Daten aller Subprozesse des Prozesses 'Beobachtung dokumentieren' enthält.

Aus dem Prozessmodell wird deutlich, dass neben der taxonomischen Bezeichnung noch weitere Daten erhoben werden müssen. Dies sind regelmäßig der Zeitpunkt und der Ort der Begehung, wobei die Messung des Ortes hier als GPS-Messung modelliert wurde. Es sind in der Praxis aber auch andere Methoden der Georeferenzierung wie die Positionsbestimmung über eine Landkarte oder die textuelle Beschreibung des Ortes üblich. In der datenorientierten Perspektive werden jeweils die Ergebnisse der einzelnen Subprozesse dokumentiert und zu einem PED des gesamten Prozesses zusammengefasst. Der Prozess 'Beobachtung dokumentieren' wird im Rahmen einer Begehung iteriert. Dadurch wird für die Dokumentation der Begehung eine Menge von PED´s erzeugt. Im Rahmen einer Begehung werden häufig Belege entnommen.

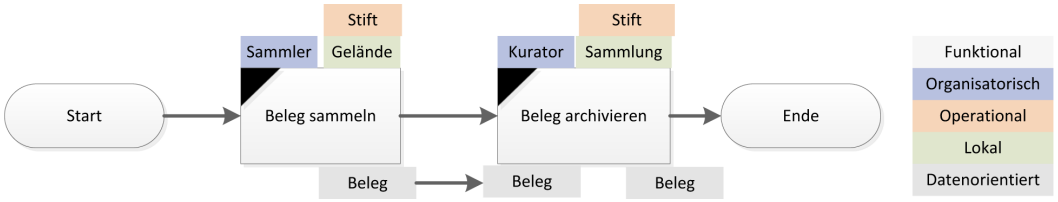

Abbildung 4.3: Prozess der Belegentnahme mit Inventarisierung

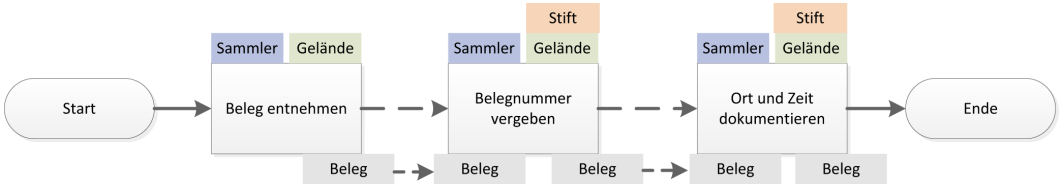

Abbildung 4.4: Subprozesse der Sammlung eines Belegs

Die Belegentnahme kann als Prozess modelliert werden.

Prozess der Belegentnahme: Der Unterschied des Prozesses der Belegentnahme zum reinen Kartierungsprozess liegt darin, dass zusätzlich ein physischer Teil des Fundobjekts entnommen und mit einem Begleitdokument versehen wird (siehe Abbildung 4.3). Dieses wird anschließend zur Archivierung eingereicht. Auch bei der Sammlung von Belegen können Subprozesse wie z.B. die Vergabe einer Belegnummer identifiziert werden (Abbildung 4.4).

Die Prozesse 'Beobachtung dokumentieren' und 'Beleg sammeln' werden im Rahmen einer Begehung häufig in Kombination am selben biologischen Objekt in beliebiger Reihenfolge ausgeführt (Abbildung 4.5). In der datenorientierten Perspektive entstehen dabei als Prozessergebnis unabhängig voneinander der Beleg und 'PED 1'. Da sich diese auf dasselbe biologische Objekt beziehen, muss dieser Zusammenhang ebenfalls dokumentiert werden. Dementsprechend ist es erforderlich, dass der Beleg 'PED 1' referenziert und umgekehrt. Folglich wird ein zusätzlicher Prozessschritt benötigt indem diese Referenz gesetzt wird wie in Abbildung 4.5 zu erkennen ist. Dies stellt zusätzliche Anforderungen an das Schema des PED des Prozesses 'Beobachtung dokumentieren', da dieses eine Referenz auf den Beleg erfassen muss. Diese zusätzliche Anforderung wird in Abbildung 4.5 dadurch visualisiert, dass das Prozessergebnis des Prozesses 'Beobachtung und Beleg referenzieren' als das ProcessExecutionDocument 'PED 2' dargestellt ist, welches das Schema von 'PED 1' um diese zusätzliche Anforderung erweitert.

An diesem einführenden Beispiel in die Prozessmodellierung in der Biodiversitätsforschung lässt sich erkennen, dass aus den Prozessmodellen, Anforderungen an

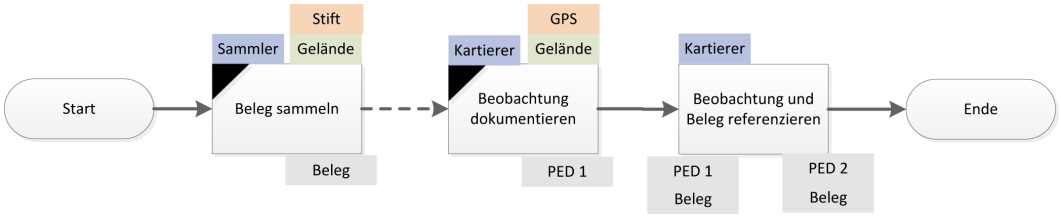

Abbildung 4.5: Kompositer Prozess aus Kartierung und Belegnahme

PED´s visualisiert und identifiziert werden können. Aus dem kombinierten Prozess der Begehung mit der Kartierung (Abbildung 4.5) wird deutlich, dass die generische Gestaltung von PED´s essentiell für die Erfassung der Prozesse in einer Domäne ist und dies durch die Prozessmodellierung in geeigneter Weise unterstützt wird. Damit ist die Anwendbarkeit von Prozessen zur Identifikation der Anforderungen an Schemata von PED´s gegeben.

4.2 Prozessorientierte Schemaevaluation (POSE)

In Kapitel 3 konnte gezeigt werden, dass das Kriterium der Vollständigkeit für die Qualität eines Datenstandards von entscheidender Bedeutung ist. Trotzdem konnte keines der vorgestellten Frameworks die Vollständigkeit eines Datenstandards zufriedenstellend messen. Daraus folgt die dringende Notwendigkeit der Entwicklung einer neuen Methode zur Messung der Vollständigkeit von Datenstandards. Mit POSE wird im folgenden Abschnitt eine Methode implementiert, die dies ermöglicht.

Im vorangegangenen Abschnitt konnte gezeigt werden, dass sich aus einem Prozessmodell Anforderungen an ein Datenschema ableiten lassen. Für die Modellierung von Datenstrukturen können aber auch andere Technologien wie z.B. die ER-Modellierung zum Einsatz kommen. Ein Problem bei der Evaluation von konzeptuellen Schemata ist aber, dass diese häufig auf Basis von subjektiven Meinungen, Erfahrung und gesundem Menschenverstand getroffen werden [170]. In der Praxis ist aber die mangelhafte Umsetzung von Anforderungen im Bezug auf das Datenschema der Grund für das Scheitern vieler Projekte [170]. Eine Evaluation auf Basis von subjektiven Kriterien ist nicht dazu geeignet, eine mangelhafte Umsetzung von Anforderungen zu erkennen. Dementsprechend fehlt es aktuell an einer strukturierten Vorgehensweise, um die Anforderungen an ein Datenschema klar zu formulieren und mit einem existierenden Datenschema zu abzugleichen.

Prozesse können aus folgenden Gründen eingesetzt werden, um dieses Problem zu lösen:

- Die Darstellung der Anforderungen eines Projekts in Form von Prozessen ermöglicht einen einfachen Zugang zu einem komplexen Problem. Somit können Anforderungen klar formuliert werden.
- Prozesse sind gut verständlich und als Technik weit verbreitet und akzeptiert.
- Mit Hilfe von Prozessen lassen sich alle relevanten Handlungen visualisieren, so dass die Verständlichkeit eines Datenschemas deutlich verbessert wird.
- Veränderte Anforderungen können durch die Evolution des korrespondierenden Prozessmodells dokumentiert werden.
- Die klare Strukturierung der Anforderungen ermöglicht die Generierung klar strukturierter Datenschemata mit Elimination von unkontrollierbaren Input (Spezialinteressen von Experten).
- In POPM ist durch die datenorientierte Prozessperspektive gut modellierbar, in welchem Prozessschritt Daten gespeichert werden müssen.

Diese Vorteile rechtfertigen die Anwendung von Prozessen zur Evaluation der Vollständigkeit von Datenschemata. Als Ausgangspunkt für die Evaluation von Datenschemata in der Biodiversitätsinformatik soll der Prozess der Geländekartierung dienen. Der Prozess der Geländekartierung stellt einen Prozess dar, der einzig zu dem Zweck der Datenaufnahme dient. In der datenorientierten Prozessperspektive wird bei Ausführung des Prozesses ein 'ProcessExecutionDocument (PED)'[1] angelegt, welches die Daten, die bei der Prozessausführung entstehen, zu diesem Zeitpunkt speichert. Folglich müssen im Schema eines PED Felder enthalten sein, in welchen die Daten der Prozessperspektiven zum Zeitpunkt der Prozessausführung enthaltenen gespeichert werden können. Jede verwendete Prozessperspektive muss dementsprechend in einem PED, in geeigneter Form repräsentiert sein.

Diesen Umstand macht sich die 'prozessorientierte Schemaevaluation' (POSE) zu Nutze. In dieser wird das Modell eines Prozesses dazu herangezogen, die Anforderungen an ein Datenschema zu spezifizieren wie in Abbildung 4.6 zu dargestellt wird. Ausgangspunkt dafür ist die Realität, welche durch ein Prozessmodell beschrieben

[1] In Folgendem wird in diesem Zusammenhang allgemein von einem 'ProcessExecutionDocument' oder PED gesprochen unabhängig davon ob es sich um eine schriftliches oder elektronisches Dokument handelt.

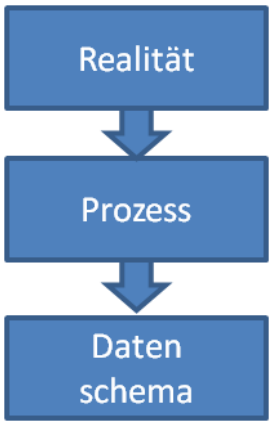

Abbildung 4.6: Moderation der Abbildung der Realität in ein Datenmodell durch Prozesse

werden kann. Dieses Prozessmodell spezifiziert in der datenorientierten Prozessperspektive PED´s, welche den Prozess beschreiben sollen. Die Anforderungen an das Schema des PED´s können dem Prozessmodell entnommen werden.

4.2.1 Aspekte in Schemata von PED´s

Damit rückt die datenorientierte Prozessperspektive in das Zentrum der Betrachtung. Wenn in einem Prozess Daten gespeichert werden sollen, muss in der datenorientierten Prozessperspektive diese Prozesses ein PED modelliert sein, welches den Prozess zum Ausführungszeitpunkt vollständig erfasst. Ein PED muss in seinem Schema Repräsentationen für folgende Merkmale eines Prozesses enthalten:

- alle Anforderungen der Prozessperspektiven des Prozessmodells
- Ausführungszeitpunkt
- Ausführungsort
- eventuell prozessunabhängige Anforderungen (z.B. zur Abbildung von Rechten wie 'Intelectual Property Rights' (IPR))

Für den Prozess der Geländekartierung können die notwendigen Aspekte des Schemas eines Dokumentes dem Prozessmodell aus Abbildung 4.2 entnommen werden. Dies ist z.B. die organisatorische Prozessperspektive, in welcher der Verantwortliche dokumentiert werden muss. Folglich muss ein Dokument für diesen Prozess über ein entsprechendes Feld verfügen. Darüber hinaus muss in der operationalen Prozessperspektive die Erhebung mittels GPS dokumentiert werden. Im funktionalen Aspekt

wird das Ergebnis des Zwecks der Prozessausführung gespeichert. Im Falle dieses Prozesses ist das die taxonomische Bestimmung. Als Merkmale der Prozessausführung werden Ausführungsort und Ausführungszeitpunkt erfasst.

Analog zu Perspektiven in Prozessmodellen können für Datenschemata verschiedene Bereiche definiert werden, welche orthogonal zueinander den Inhalt eines Datenschemas beschreiben. Diese Bereiche werden als 'Aspekte' bezeichnet.

Definition 4.1: Aspekt

Ein Aspekt eines PED ist ein Teilbereich einer Aussage zur eindeutigen Speicherung von Daten eines bestimmten Themenbereichs, der orthogonal zu allen anderen Aspekten eines PED´s steht.

Orthogonal bedeutet in diesem Zusammenhang, dass sich die Aspekte nicht überlappen, also eine thematisch disjunkte Gliederung eines Dokumentes ermöglichen. Die Auswahl der Aspekte ist dabei von der betrachteten Domäne und der Art der Prozesse dieser Domäne abhängig. Folgende Aspekte zeichnen sich dadurch aus, dass sie für die Domäne der Biodiversitätsinformatik relevant sind:

- Funktionaler Aspekt: Speicherung der Primärdaten. Das sind die Daten zu dessen Zweck der Prozess der Datenerhebung ausgeführt wurde (z.B. Daten von Messungen, taxonomische Bestimmungen).
- Organisatorischer Aspekt: Speicherung des Verantwortlichen der Prozessausführung
- Operationaler Aspekt: Speicherung der verwendeten Werkzeuge und Hilfsmittel
- Datenorientierter Aspekt: Speicherung von Referenzen auf andere Daten und Dokumente, die während eines Erhebungsprozesses erfasst wurden oder Speicherung von Referenzen auf physische oder virtuelle Objekte des Erhebungsprozesses (z.B. Mitnahme von Belegen, Multimediaobjekte, externe Daten)
- Temporaler Aspekt: Speicherung des Zeitpunkts der Prozessausführung
- Lokaler Aspekt: Speicherung des Ausführungsorts
- Verhaltensorientierter Aspekt: Im Verhaltensorientierten Aspekt wird die zeitliche Abfolge zwischen Prozessen erfasst. Somit werden in diesem Aspekt verschiedene Aussagen mit ihrer zeitlichen Reihenfolge verknüpft.

Kartierer:	Josef Simmel
Zeitpunkt:	14.07.2012
Sammelort(Klartext):	Großer Waldweg westlich Bruckhäusl (MTB 6939/2) und angrenzende Waldbereiche.
Latitude:	12.291050911
Longitude:	49.070976257
GPS-Chip:	SiRF III
Satteliten:	3
Fehlertoleranz	30 m
Taxonomische Bezeichnung:	Salix fragilis L.

Abbildung 4.7: Mangel an Genauigkeit im temporalen Dokumentenaspekt

Zur klaren sprachlichen Abgrenzung von den Perspektiven bei Prozessen werden im Folgenden die Begriffe Prozessperspektive und Dokumentenaspekt verwendet. Den Dokumentenaspekten sind die elementaren Fragen nach Was?, Wer?, Wie?, Wann? und Wo? zugeordnet. Die Auflistung der Dokumentenaspekte ist dabei analog zu den Prozessperspektiven nicht abschließend. Je nach Anwendungsfall kann es erforderlich sein, weitere Dokumentenaspekte aufzunehmen. Für die Erfüllung des Vollständigkeitskriteriums eines PED sind folgende Regeln einzuhalten:

1. Für jede Prozessperspektive muss ein korrespondierender Dokumentenaspekt existieren.

2. Der lokale und temporale Dokumentenaspekt muss existieren, um die Prozessausführung abzubilden.

Durch diese Regeln werden klare Anforderungen an das Schema eines PED formuliert. Sie dienen vor allem zur Identifikation von Anforderungen, die im Schema eines PEED nicht abgebildet sind (Fehler 1. Art nach Moody). Diese Anforderungen sind für die Anwendbarkeit eines PED´s zur Erfassung eines Prozesses besonders wichtig, da ein unvollständiges Schema zu Datenverlust führt und somit für die Dokumentation eines Prozesses ungeeignet ist.

4.2.2 Genauigkeit der Erfassung von Dokumentenaspekten

Es genügt nicht nur, dass alle Prozessperspektiven in einem Dokumentenaspekt repräsentiert sind. Diese Repräsentation muss auch mit einer bestimmten Genauigkeit erfolgen, damit das Schema des PED den Prozess ausreichend erfassen kann.

Ein PED zur Erfassung des Prozesses der Geländekartierung findet sich in Abbildung 4.7. Das Schema des PED enthält für alle Prozessperspektiven aus Abbildung 4.2 Dokumentenaspekte zur Aufnahme der Daten. Das Schema des PED´s kann

diese Dokumentenaspekte nur in einer bestimmten Genauigkeit erfassen. Aus dem Datensatz kann aber geschlossen werden, dass die Uhrzeit der Prozessausführung nicht erfasst wurde. Diese gehört aber regelmäßig zu den Anforderungen der Geländekartierung. Die Anforderungen des Prozesses an das Dokument zur Erfassung des Prozesses sind damit nicht vollständig erfüllt. Ein Mangel an Genauigkeit führt in der Evaluation der Vollständigkeit eines Dokuments zu einem Fehler 3. Art (vgl. Abschnitt 3.2.2 und folgender Abschnitt).

Die klare Untergliederung eines PED´s in Dokumentenaspekte ermöglicht es, diese Fehler im Design zu identifizieren. Die erforderliche Genauigkeit muss dabei durch Interaktion mit Domänenexperten ermittelt werden, wobei das Prozessmodell als Diskussionsgrundlage dient.

4.2.3 Evaluation von Schemata von Dokumenten zur Erfassung eines Prozesses

Die Analyse eines Datenschemas in Dokumentenaspekte hat die Aufgabe zu evaluieren, ob die Anforderungen eines Prozesses in einem Datenschema vollständig abgebildet wurden. Dazu werden die Anforderungen des Prozesses an die Dokumentenaspekte und Genauigkeit in einem **Anforderungsprofil** zusammengefasst. Analog dazu werden die Attribute eines PED´s nach Dokumentenaspekten zusammengefasst und die erfassbare Genauigkeit bestimmt. Dadurch entsteht eine **Schemaprofil** des Dokuments. Diese werden, wie in Abbildung 4.8 dargestellt ist, zur Evaluation miteinander abgeglichen. Grundlage für diese Vorgehensweise ist die Tatsache, dass alle Dokumentenaspekte eines PED aus einem Prozessmodell abgeleitet werden können. Nach POPM können aber auch alle Anforderungen eines Prozesses in den Prozessperspektiven formuliert werden und es kann somit ein Anforderungsprofil erstellt werden. Analog dazu können alle Felder des Schemas eines PED´s den Dokumentenaspekten zugeordnet werden. So kann auf dieser Basis das Schemaprofil erstellt werden. Damit ist der Abgleich zwischen den Anforderungen an ein Schema eines PED´s und dem tatsächlichen Schema eine PED´s möglich.

Das Anforderungsprofil liefert die Antwort auf die Frage 'Was muss dargestellt werden können?' und spezifiziert damit den Soll-Zustand eines Datenschemas zur Erfassung der Prozesse in einem bestimmten Prozessmodell. Das Schemaprofil beantwortet hingegen die Frage, was ein gegebenes Datenschema abbilden kann und stellt damit den Ist-Zustand dieses Datenschemas dar. Damit können Anforderungsprofil und Schemaprofil prinzipiell voneinander unabhängig und in beliebiger Reihenfolge erstellt werden. Da aber die genaue Kenntnis der Anforderungen einen besseren

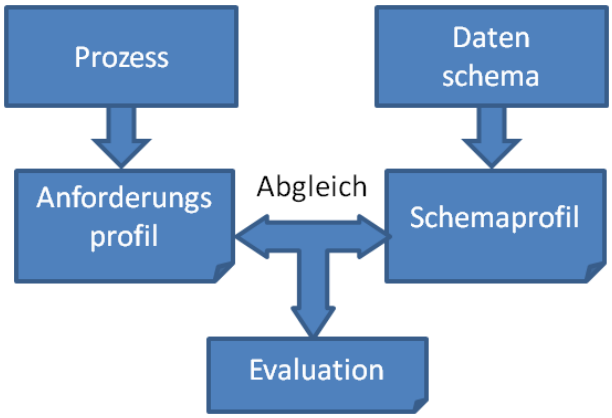

Abbildung 4.8: Struktur der Evaluation nach POSE

Zugang zur Anwendungsdomäne ermöglicht empfiehlt sich folgende Reihenfolge der Ausführung bei der Evaluation der Vollständigkeit eines Datenschemas mit POSE:

1. Prozessmodell erstellen
2. Anforderungsprofil erstellen
3. Schemaprofil erstellen
4. Soll-Ist-Abgleich

Die Fehler nach [168] (siehe Abschnitt 3.2.2 und Abbildung 3.6) werden in POSE in folgender Weise implementiert:

Fehler 1. Art Ein Element des Anforderungsprofils ist nicht im Schemaprofil vorhanden

Fehler 2. Art Ein Element des Schemaprofils ist nicht im Anforderungsprofil vorhanden

Fehler 3. Art Ein Element des Anforderungsprofils ist nicht mit der erforderlichen Genauigkeit im Schemaprofil abgebildet

Das Datenschema genügt genau dann den Anforderungen eines Prozesses, wenn alle Merkmale der Prozessausführung in ausreichender Genauigkeit repräsentiert sind. Dabei führen Fehler der 1. und 3. Art zu Datenverlusten, die später nicht mehr korrigiert werden können. Ein Fehler 1. Art führt im Allgemeinen dazu, dass das Datenschema zur Prozessbeschreibung ungeeignet ist. Fehler 2. Art hingegen können die Datenqualität negativ beeinflussen, indem Artefakte erzeugt werden und so

fehlerhafte Daten gespeichert werden. Dies ist deutlich schwieriger zu vermeiden – insbesondere wenn dasselbe Dokument zur Erfassung mehrere Prozesse verwendet wird. In diesem Fall muss das Dokument den Anforderungen aller Prozesse genügen, in denen es verwendet wird.

Die Evaluation eines Datenschemas kann selbst als Prozess verstanden werden. Ein Prozessmodell für die Vollständigkeitsevaluation nach POSE ist in Abbildung 4.9 dargestellt. POSE besteht dabei aus den kompositen Subprozessen 'P1: Prozesse der Anwendungsdomäne modellieren', 'P2: Anforderungen identifizieren', 'P3: Schema analysieren', und 'P4: Schemaprofil mit Anforderungsprofil abgleichen'. Dabei werden die Subprozesse von P2 und P3 in Abbildung 4.9 explizit dargestellt.

Ausgangspunkt für POSE ist das Prozessmodell eines bestimmten Anwendungsfalls, welches in P1 erzeugt wird. Aus den Prozessperspektiven dieses Prozessmodells wird in P2 das Anforderungsprofil erstellt. Dazu werden zum einen die benötigten Dokumentenaspekte aus dem Prozessmodell abgeleitet. Zum anderen müssen Ausführungsort und -zeitpunkt im Schema eines PED erfasst werden können. Zusätzlich müssen weitere prozessunabhängige Anforderungen wie die Erfassung von 'Intelectual Property Rights' (IPR) berücksichtigt werden können. Alle Anforderungen werden in dem Prozess 'Anforderungen nach Dokumentenaspekten ordnen' bestimmten Dokumentenaspekten zugewiesen. So wird in einer geordneten Liste von Anforderungen ermittelt, was ein Schema erfassen muss. Anschließend wird im Prozess 'Genauigkeit identifizieren' für jeden Eintrag der geordneten Anforderungsliste die erforderliche Genauigkeit bestimmt. Das Ergebnis ist das Anforderungsprofil, welches gleichzeitig das Ergebnis von P2 darstellt.

Analog zur Erstellung des Anforderungsprofils werden in P3 die Elemente des Schemas eines PED´s den verschiedenen Dokumentenaspekten zugewiesen und die erfasste Genauigkeit ermittelt. Dadurch entsteht das Schemaprofil des PED. In P4 werden die in P2 ermittelten Anforderungen mit dem nach Dokumentenaspekten strukturierten Dokument aus P3 verglichen. Durch diesen Vergleich werden als Ergebnis der Evaluation die Fehler der erster bis dritter Art identifiziert. Bei der Evaluation mit POSE ist zu beachten, dass in P2 spezifische Anforderungen des Prozesses formuliert werden. P2 ist von einem bestimmten Datenschema unabhängig und formuliert Anforderungen an alle potentiellen Schemata. P3 ist spezifisch für ein bestimmtes PED. Bei der Evaluation eines anderen PED muss P3 erneut ausgeführt werden. Eine neue Ausführung von P2 ist nicht erforderlich, da sich die Anforderungen der Prozesserfassung bei der Evaluation eines anderen Datenstandards nicht ändern. In P4 wird das Ergebnis der Evaluation für ein PED ermittelt.

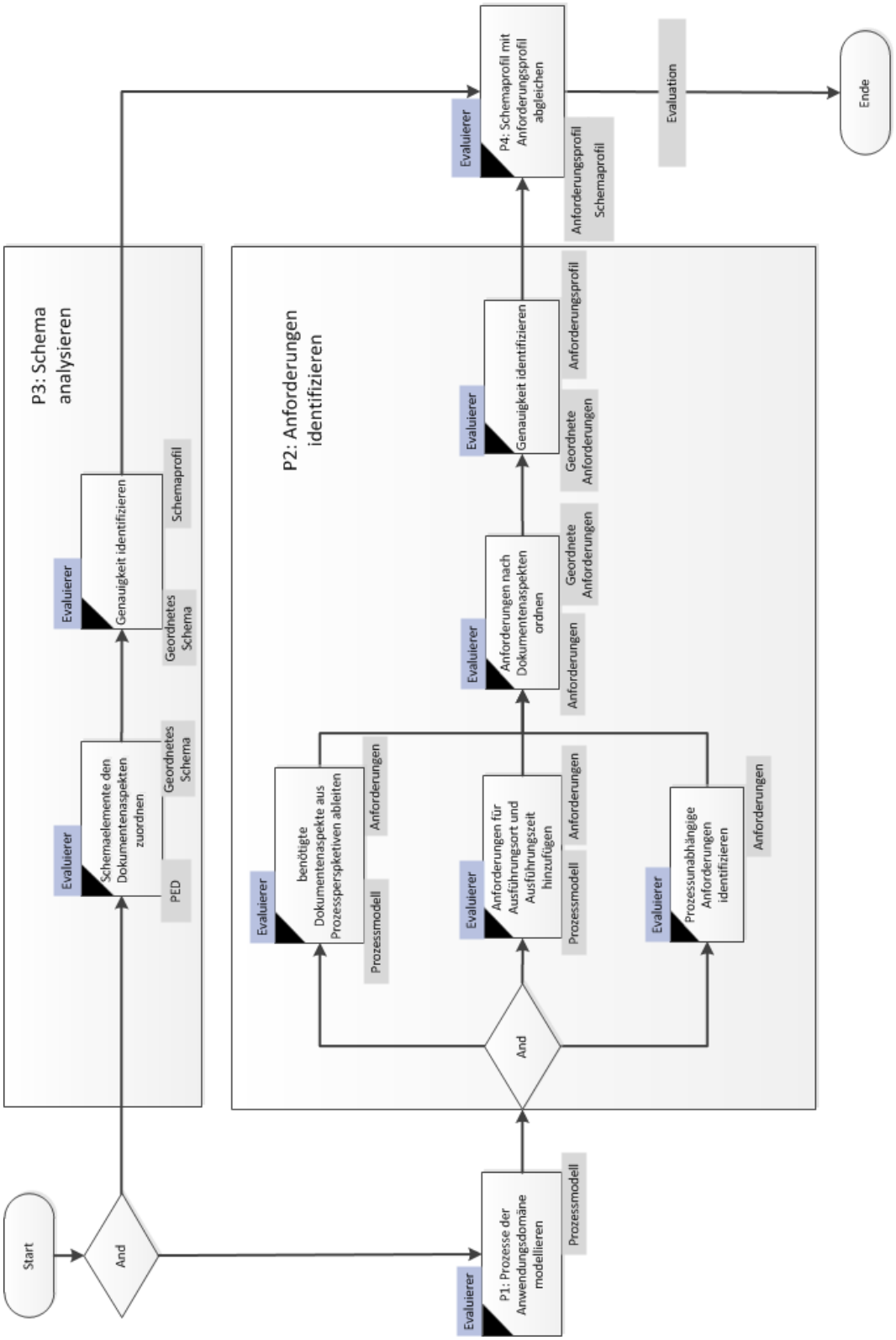

Abbildung 4.9: POSE als Prozess

Abbildung 4.10: IBF-Datensatz aus GBIF

4.2.4 Beispiel

Ausgangspunkt für dieses Beispiel ist eine Geländekartierung mit Belegnahme und Multimediaeinsatz, welche im IBF-Projekt ausgeführt wurde. Im konkreten Anwendungsfall wurde in einem bestimmtes Areal der Bestand von Pilzen kartiert und neben der Entnahme eines Belegs der Pilz zusätzlich fotografiert. Die Datenerhebung erfolgte zum einen über DiversityMobile und zum anderen über einen physischen Beleg. Anschließend wurden die Daten an die Server des SNSB-IT-Center übertragen und der Beleg am SNSB eingereicht. Von dort wurden die Daten über das GBIF-Portal publiziert. Das Ergebnis der Suche nach einem Datensatz aus dieser Kartierung ist in Abbildung 4.10 dargestellt. Dementsprechend soll in diesem Beispiel evaluiert werden, ob das Datenschema von GBIF die Anforderungen des Prozesses ausreichend erfüllt. Dazu wird in diesem Beispiel das Schema direkt aus dem publizierten Datensatz abgeleitet[2].

Es stellt sich nun die Frage, wie gut das vom GBIF-Portal verwendete Datenschema geeignet ist, um den zu Grunde liegenden Prozess zu erfassen. Für die Erstellung des Anforderungsprofil ist dazu zunächst ein Prozessmodell zu erstellen. Dieses ist in in Abbildung 4.11 dargestellt. Am Prozessmodell ist ersichtlich, dass für die Erfassung des gesamten Prozesses zwei PED´s und ein Beleg benötigt werden. Dabei ist der Beleg ein physisches Dokument wohingegen 'PED 1' und 'PED 2' elektronische Dokumente in DiversityMobile sind. Der in Abbildung 4.10 dargestellte Datensatz zeigt das Ergebnis des Prozesses 'Fund dokumentieren' und korrespondiert damit mit 'PED 1'.

[2]Tatsächlich verwendet GBIF intern ABCD und DwC. Die Schemata dieser Datenstandards sind veröffentlicht und werden in Abschnitt 4.4.1 und 4.4.2 evaluiert.

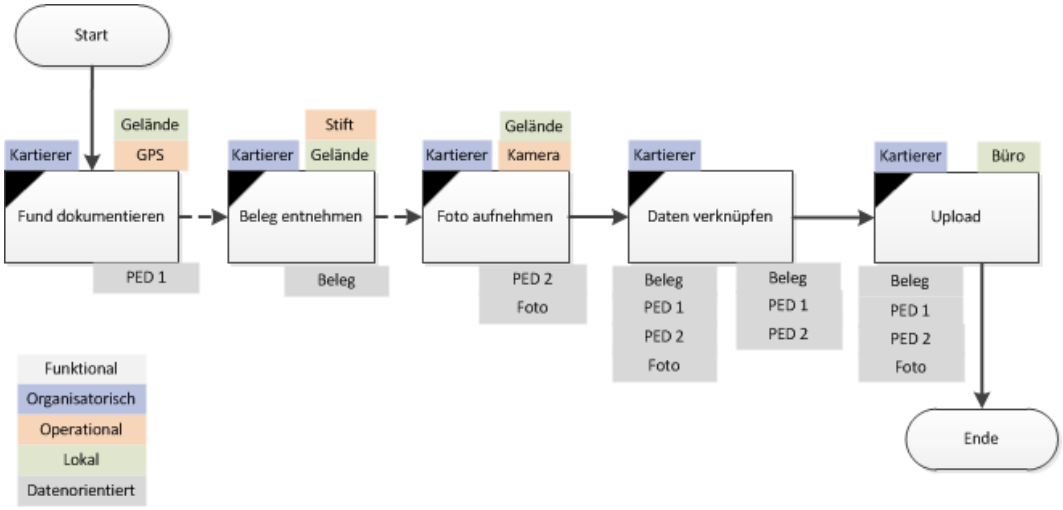

Abbildung 4.11: Prozessmodell für den Anwendungsfall des Beispiels

Folglich muss im nächsten Schritt ein Anforderungsprofil für 'PED 1' erstellt werden. Aus dem Prozessmodell kann für 'PED 1' in der funktionalen Prozessperspektive die Notwendigkeit eines Speicherplatzes für die taxonomische Identifikation und aus der organisatorischen Prozessperspektiven die Notwendigkeit eines Speicherplatzes für den Sammler ermitteln. Zusätzlich wird in der datenorientierten Prozessperspektive ein Speicherort für die Referenz auf den Beleg und das Foto gefordert. Darüber hinaus werden für die Dokumentation der Prozessausführung in der Biodiversitätsinformatik immer Ausführungsort und Ausführungszeitpunkt benötigt. Die Modellierung der erforderlichen Genauigkeit in Zusammenarbeit mit Domänenexperten ergibt damit das Anforderungsprofil aus Tabelle 4.1.

Aus dem publizierten Datensatz wird das Schemaprofil des Dokuments zu dem Prozess 'Beobachtung dokumentieren' erstellt. Dieses ist in Tabelle 4.2 dargestellt. Dazu wird aus dem in Abbildung 4.10 publizierten Datensatz ein Schema abgeleitet und dieses den Dokumentenaspekten zugeordnet. Felder, welche zur internen Verwaltung unter GBIF dienen, wurden dabei nicht berücksichtigt.

Nun kann das Anforderungsprofil für 'PED 1' mit dem Schemaprofil abgeglichen werden. Da sowohl im Anforderungsprofil als auch im Schemaprofil die Identifikation des Fundobjekts erfasst ist und sowohl die taxonomische Bezeichnung als auch eine URL gefordert wird und abgespeichert werden kann, sind die Anforderungen des funktionalen Dokumentenaspekts von 'PED 1' erfüllt. Dies ist im organisatorischen Dokumentenaspekt nicht der Fall, da das Anforderungsprofil für den Sammler die Genauigkeit der Identifikation des Sammler über eine URL erfordert. Dies kann aber das Schemaprofil nicht erfüllen, da im Schema die Möglichkeit der Speiche-

Aspekt	Attribut	Genauigkeit	Datentyp
Funktional	Identifikation	Taxonomische Bezeichnung URL	String String
Organisatorisch	Sammler	Name URL	String String
Operational	GPS	Abweichung des Ausführungsorts	Float
Temporal	Ausführungszeitpunkt	Jahr, Monat, Tag, Stunden, Minuten, Sekunden	DateTime
Lokal	Ausführungsort	WGS84-Latitude WGS84-Longitude	Float Float
Datenorientiert	Identifier	URL	String
	Referenz Beleg	Belegnummer	String
	Referenz Bild	URL	String

Tabelle 4.1: Anforderungsprofil von 'PED 1' zu dem Prozess 'Fund dokumentieren'

Aspekt	Attribut	Genauigkeit	Datentyp
Funktional	Identifikation	Taxonomische Bezeichnung URL	String String
Organisatorisch	Sammler	Name	String
	Institut	Code	String
Operational	---	---	---
Temporal	Ausführungszeitraum	Datum	DateTime
Lokal	Ausführungsort	WGS84-Latitude WGS84-Longitude	Float Float
	Ausführungsort	Textuell	String
Datenorientiert	Identifier	URL	String

Tabelle 4.2: Schemaprofil des Datensatzes aus Abbildung 4.10

Aspekt	Fehler	Ursache
Funktional	---	---
Organisatorisch	2. Art	Institut
	3. Art	URL Sammler
Operational	1. Art	GPS - Messgenauigkeit
Temporal	1. Art	Ausführungszeit
	2. Art	Begehungsanfang und -ende
Lokal	2. Art	Textuelle Beschreibung
Datenorientiert	1. Art	Referenz Beleg
	1. Art	Referenz Bild

Tabelle 4.3: Evaluation von 'PED 1' zu dem Prozess 'Fund dokumentieren' aus Abbildung 4.11

rung einer URL nicht vorgesehen ist. Der Sammler kann nur textuell erfasst werden. Dementsprechend liegt ein Mangel in der Genauigkeit der Speicherung des Sammlers im Schemaprofil vor. Dies ist ein Fehler 3. Art. Andererseits ist im Schemaprofil im organisatorischen Dokumentenaspekt die Speicherung des Instituts des Sammlers gefordert. Im Anforderungsprofil ist aber keine korrespondierende Anforderung zu finden. Damit liegt eine Übermodellierung des Schemas vor. Dies ist ein Fehler 2. Art und führt dazu, dass zusätzlich Daten erhoben und gespeichert werden müssen, die nicht in den Anforderungen eines Prozesses enthalten sind. Im datenorientierten Dokumentenaspekt wird im Anforderungsprofil eine Speichermöglichkeit für Referenzen auf den Beleg und das Bild gefordert. Diese sind im Schemaprofil überhaupt nicht vorhanden und diese Daten können um untersuchten Datenschema somit nicht gespeichert werden. Damit liegen zwei Fehler 1. Art vor, da beide Referenzen nicht gespeichert werden können. Durch systematischen Abgleich der Anforderungen mit dem Schemaprofil in allen Dokumentenaspekten entsteht die vollständige Evaluation für 'PED 1'. Diese ist in Tabelle 4.3 dargestellt.

Besonders kritisch sind bei der Evaluation sind Fehler 1. Art. Diese zeigen an, dass Anforderung überhaupt nicht erfüllt wurden. Dies ist in Tabelle 4.3 im daten-

orientierten Dokumentenaspekt die Referenzierung des Belegs und im operationalen Dokumentenaspekt die Erfassung der Genauigkeit der GPS-Bestimmung. Darüber hinaus existieren mehrere Fehler 3. Art. Der Prozess 'Fund dokumentieren' ist damit nicht ausreichend in 'PED 1' erfasst.

4.2.5 POSE in der Biodiversitätsforschung

In der Biodiversitätsforschung werden in einer Vielzahl von Prozessen Daten erfasst. So wird in diesem Rahmen nicht nur die Sammlung von Felddaten in Form von Belegen, sondern – wie in Kapitel 2.1 beschrieben – auch eine Vielzahl von ökologischen Zusammenhängen und taxonomischen Fragestellungen untersucht. Dementsprechend kann ein einziges Dokument nicht den Anforderungen aller Prozesse dieser Domäne genügen.

Es soll sich deshalb im Folgenden auf eine Auswahl von Prozessen beschränkt werden, welche im Rahmen der Projektarbeit von IBF als typisch identifiziert werden konnten. Diese werden als Grundlage für die Evaluation von Datenstandards mit POSE herangezogen. Selbstverständlich decken diese Fälle nicht alle tatsächlichen und denkbaren Fälle der Biodiversitätsinformatik ab. In ihnen werden aber praxisrelevante Anforderungen modelliert. Sind diese von einem Datenstandard nicht erfüllt, ist dieses zumindest fehlerhaft identifiziert. Für die Betrachtung anderer Anwendungsfälle kann aber das Kriterium der Flexibilität (siehe Abschnitt 4.4) herangezogen werden, welches die Prozesse zur Anpassung eines Datenstandards bewertet. Die Anwendungsfälle UC1-UC5 sind dabei in ihrer Beschreibung abstrakt formuliert. Zusätzlich gibt es zur Veranschaulichung in den Abbildungen 4.12 bis 4.16 Prozessmodelle von konkreten Beispielen. Die Anwendungsfälle UC1-UC5 stellen dabei typische Anwendungsfälle aus der Domäne der Biodiversitätswissenschaft in absteigender Bedeutung von UC1 nach UC5 dar. Dafür werden die Anforderungen von UC1 bis UC5 immer spezieller.

Use Case 1(UC1) Geländebegehung: Ein Wissenschaftler führt eine Geländekartierung aus. Im Rahmen dieser Begehung soll das Vorkommen von Arten untersucht werden. Das Forschungsinteresse besteht in einer Analyse der Veränderung von Artvorkommen im Laufe eines gewissen Zeitraums mit einer exakten Erfassung der Position über GPS (Abbildung 4.12). Die Daten der Prozessausführung werden in 'PED 1' gespeichert. Anschließend werden diese Daten an ein Repositorium übermittelt.

Use Case 2 (UC2) Geländebegehung mit Beleg: Analog zu UC1 - Doch nun wird zusätzlich ein Teil eines biologischen Objekts (z.B. ein Ast) als Beleg ein-

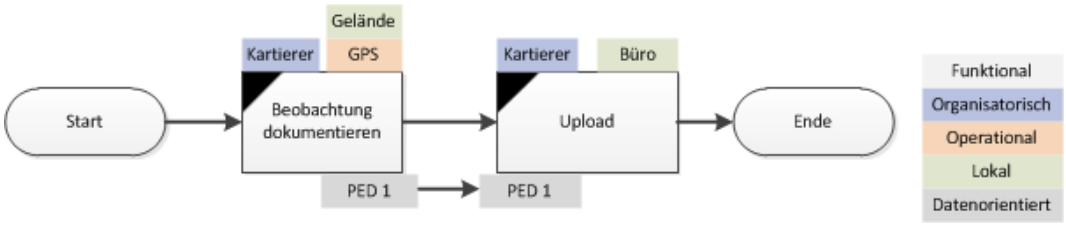

Abbildung 4.12: Geländekartierung ohne Belegentnahme

gesammelt und zur Archivierung bei einer biologische Sammlung eingereicht. Damit der Beleg der Beobachtung zugeordnet werden kann, wird dies in einem Begleitdokument in schriftlicher Form festgehalten (Abbildung 4.13). UC2 kombiniert UC1 mit der Entnahme eines physischen Belegs. Dadurch entstehen im Gelände zwei Dokumente ('PED 2' und der Beleg), die miteinander verknüpft werden müssen. In der anschließenden elektronischen Erfassung in der Sammlung wird 'PED 3' erzeugt, welches die Sammlungsnummer als einen spezifischen Identifier des Belegs in der Sammlung und Informationen über die Art und den Standort der Archivierung erfasst. Durch die paarweise Verknüpfung von 'PED 2' mit dem Beleg und mit 'PED 3' sind diese mittelbar miteinander verknüpft. Nach dem Upload in das Repository ist es die Aufgabe des IT-Systems, diesen Zusammenhang in 'PED 2' und 'PED 3' explizit abzuspeichern.

Use Case 3 (UC3) Geländebegehung mit Multimediadokumentation: Ausgangspunkt ist die Geländebegehung. Dabei werden Objekte kartiert, die in Verbindung mit anderen Organismen leben (z.B. Pilze und Pflanzen). Zur Dokumentation wird eine Multimediaaufnahme (z.B. ein Foto) mit den beobachteten Objekten erstellt und in das Repositorium einer Sammlung hochgeladen. Die Beziehung zwischen dem Multimediadokument und der Dokumentation des Fundes muss dabei erhalten bleiben. Forschungsinteresse ist die Art und Weise wie die verschiedenen Organismen zusammenleben (Abbildung 4.14). UC3 unterscheidet sich von UC2 dadurch, dass anstelle eines physischen Belegs ein Multimediadokument als Nachweis der Beobachtung aufgenommen wird und zusätzlich zwei Beobachtungen miteinander verknüpft werden müssen. Die Daten bezüglich der Multimediaaufnahme müssen in 'PED 6' in einem eigenen PED gespeichert werden und können so in 'PED 4' und 'PED 5' referenziert werden. Dabei ist insbesondere der Speicherort des Multimediadokuments von Interesse, welcher im Uploadprozess in 'PED 6' aktualisiert werden muss.

Use Case 4 (UC4) Beleg mit DNA Analyse: Einem in einer Sammlung archivierten Beleg wird zur DNA-Analyse bei einem Forschungslabor eingesendet,

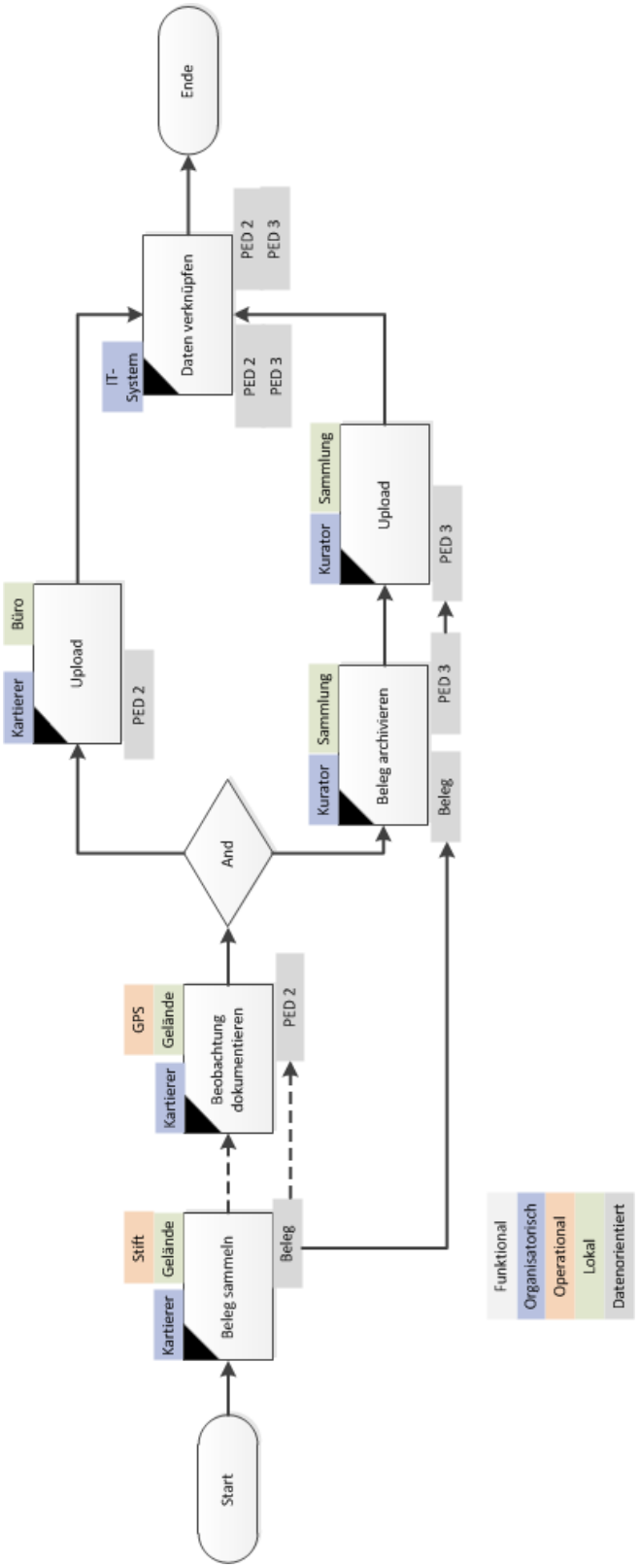

Abbildung 4.13: Geländekartierung mit Belegentnahme

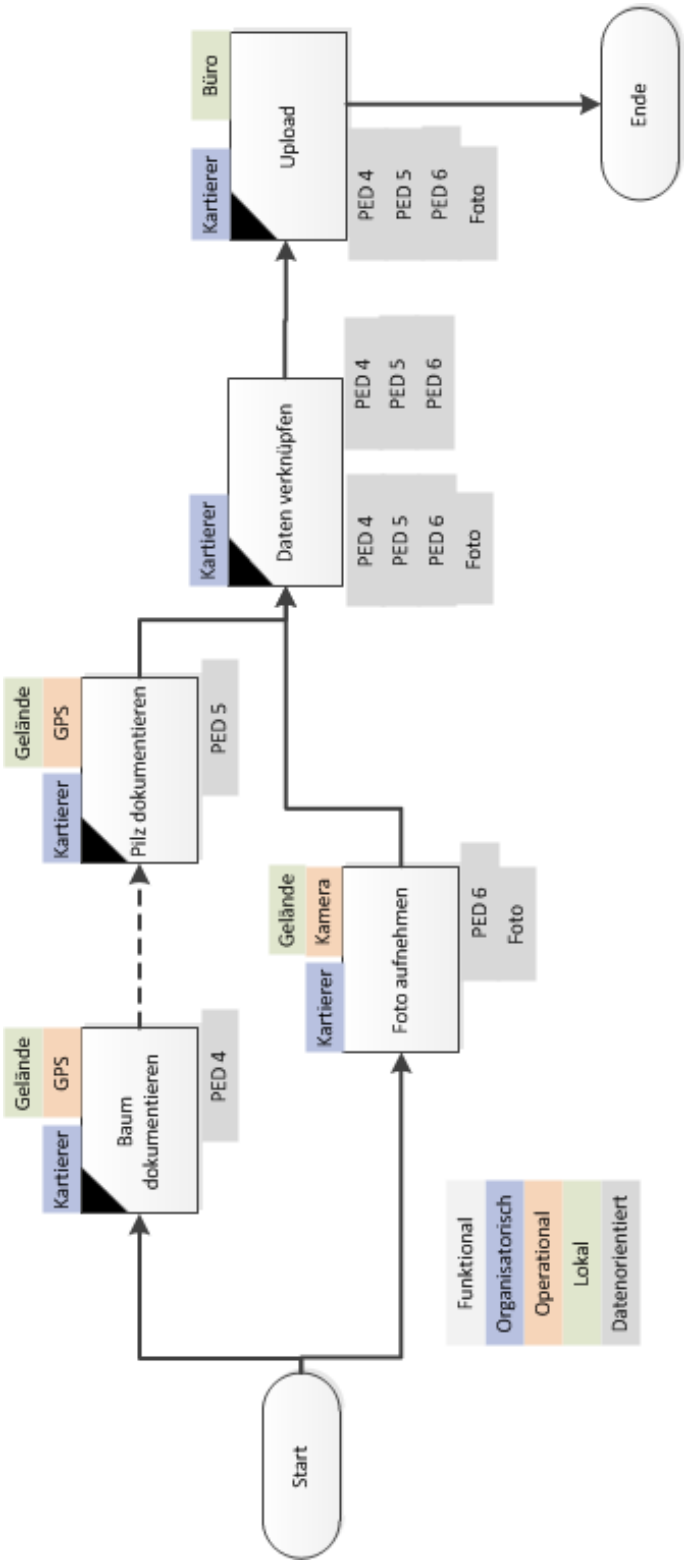

Abbildung 4.14: Geländekartierung mit Verknüpfung von Organismen und Multimediaaufnahme

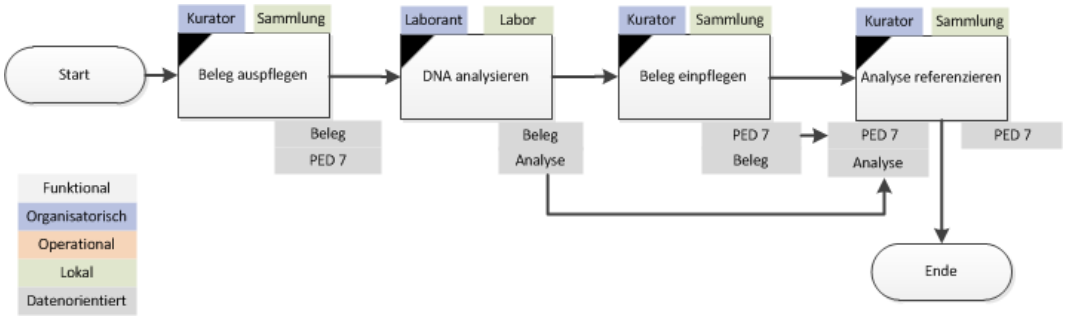

Abbildung 4.15: DNA Analyse eines archivierten Belegs

welches die DNA der Probe untersucht und die DNA-Analyse an die biologische Sammlung zurückschickt (Abbildung 4.15). In UC4 wird ein in einer Sammlung eingepflegter Beleg an einen Dienstleister zur DNA-Analyse übertragen. Dazu wird der Beleg ausgepflegt und an das Institut übersendet. Für die Verwaltung eines Belegs in einer Sammlung ist 'PED 7' verantwortlich, in welchem alle Schritte der Belegverwaltung hinterlegt werden. Nach erfolgter Analyse muss der Beleg wieder eingepflegt werden, so dass diesem eine Historie zugeordnet werden muss. Zusätzlich entsteht als Dokument die DNA-Analyse. An dieser Stelle können keine Anforderungen zur allgemeinen Strukturierung von DNA-Analysen gegeben werden, da hierbei die Grenzen der Domäne der Biodiversitätsinformatik weit überschritten werden würden. Es ist im organisatorischen Dokumentenaspekt von Interesse, wer für die Analyse verantwortlich war und im temporalen Dokumentenaspekt wann diese durchgeführt wurde. In der Analyse werden diese Daten erfasst und es wird in 'PED 7' eine Referenz auf die eigentlichen DNA-Analyse gesetzt.

Use Case 5 (UC5) Ökologie in Kultivierung mit Messungen: Auf einer Versuchsfläche werden Organismen (z.B. Bäume der Art 'Salix fragilis' in einem botanischen Garten) angepflanzt. Es soll das Wachstum der Organismen, sowie von Teilen der Organismen detailgetreu erfasst werden (z.B. Äste und Blätter der Bäume). Die Teile eines Organismus können dabei wiederum mit anderen Organismen in Beziehung stehen (z.B. Gallenbefall von Blättern in Abbildung 4.16).

UC5 stellt eine Prozess dar, wie er in ökologischen Projekten gefunden werden kann. Hierbei ist weniger die Standortinformation von Interesse, sondern die Entwicklung einer kultivierten Pflanze im Rahmen einer Zeitreihenanalyse. Dadurch werden fundamental andere Anforderungen an die Dokumente zur Archivierung gestellt. In den einzelnen Schritten müssen die Dokumente mit den Dokumenten der jeweils nächsthöheren Ebene verknüpft werden.

Die Anwendungsfälle UC1-UC5 decken das typische Arbeitsfeld von Wissenschaft-

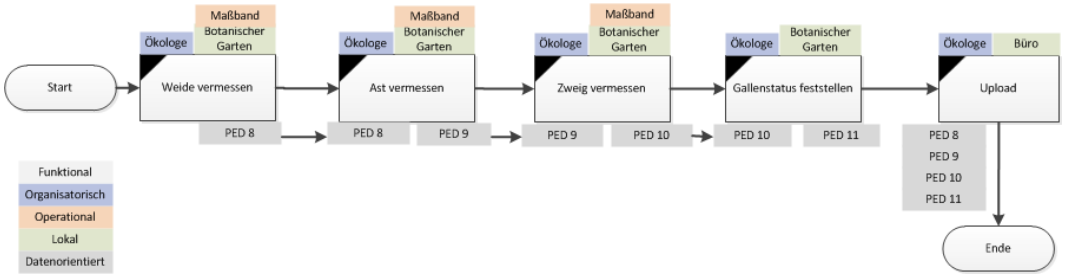

Abbildung 4.16: Ökologische Messungen

lern in der Biodiversitätsforschung ab. Alle Anwendungsfälle sind im Umfeld des IBF-Projekts aufgetreten und werden auch in anderen Bereichen der Biodiversitätsforschung verwendet. Der allgemeine Charakter der Anwendungsfälle lässt sich wie folgt begründen:

- Biodiversitätsinformatik deckt ein weites Spektrum an Projekten ab. Dieses beinhaltet z.B. ökologische, botanische und auch fungologische Projekte. In allen diesen Projekten ist es aber eine zentrale Aufgabe, das Vorkommen einer Art zu einem bestimmten Zeitpunkt an einem bestimmten Ort zu dokumentieren (UC1).

- Die Verifikation von Funden mit Belegen ist eine zentrale Anforderung aus der Sammlungsverwaltung (siehe Kapitel 2). So werden am SNSB insgesamt etwa 30 Millionen Einzelobjekte archiviert. Diese Belege wurden über den Zeitraum von mehr als 100 Jahren gesammelt, aber noch heute zählt die Arbeit mit Belegen zu Standardmethoden der Geländekartierung. Da aber zunehmend Beobachtungsdaten direkt im Feld erhoben werden, muss der kombinierte Prozess unterstützt werden (UC2).

- Durch die Multimediaaufnahme von Bild-, Video und Tondokumenten kann in den Fällen, in der eine sicher Artbestimmung im Feld erfolgen kann, auf die klassische Belegnahme verzichtet werden. Damit wird diese durch die Multimediadokumentation ersetzt und ist eine zentrale Anforderung für Projekte in der Biodiversitätswissenschaft (UC3).

- In ökologischen Projekten und in Projekten zu biologischen Organismen, welche mit anderen Organismen zusammenleben, müssen diese Beziehungen und Modalitäten des Zusammenlebens dokumentierbar sein. Dies ist insbesondere für ökologische Projekte eine zentrale Anforderung (UC3).

- In den letzten Jahren hat die genetische Analyse von Organismen in der Praxis zunehmend an Bedeutung gewonnen. Dementsprechend werden genetische Analysen zunehmend auch in Biodiversitätsprojekten eingesetzt. Die genetische Sequenzierung eines biologischen Objektes ist allerdings mit einem großen Umfang an Daten verbunden, für die von der Bioinformatik eigene Datenstandards entwickelt wurden. Da dies eine benachbarte Wissenschaft ist, empfiehlt es sich für die Biodiversitätsinformatik auch auf diese Datenstandards zurückzugreifen und keinen eigenen Standards für Genetik zu entwickeln. Dementsprechend müssen Ergebnisse der Gensequenzenzierung als externe Dokumente berücksichtigt werden. (UC4)

- Die Verwaltung von Belegen gehört zum klassischen Aufgabengebiet von naturwissenschaftlichen Sammlungen. Dabei wird die Gensequenzierung zur genauen Artbestimmung im Allgemeinen von den Sammlungen angeordnet. Da dazu ein Beleg oder der Teil eines Belegs einer Sammlung (wie auch bei dem Verleih eines Belegs) zeitweise entnommen wird, muss das Ein- und Auspflegen eines Belegs berücksichtigt werden (UC4).

- Messungen beschreiben den Zustand eines biologischen Objektes zu einem Zeitpunkt und sind die Grundlage für die Zeitreihenanalyse in Biodiversitätsprojekten. Diese sind ein zentrales Thema der Biodiversitätswissenschaft (UC5).

- Die detailgetreue Erfassung von biologischen Objekten mit der Erfassung von Teilen eines Objektes ist insbesondere in Projekten von Bedeutung, in denen kultivierte biologische Objekte und das Zusammenleben dieser mit anderen Organismen untersucht werden. Sie kann aber auch bei einer Kartierung von Bedeutung sein, wenn z.B. dokumentiert werden soll, dass ein spezieller Ast von einem Pilz befallen ist. Diese Anforderung trat aber im IBF-Projekt vor allem bei der Untersuchung des Gallenbefalls von Pflanzen auf. Allerdings ist zu vermuten, dass auch in anderen ökologischen Projekten eine ähnlich detaillierte Erfassung von biologischen Objekten erforderlich ist (UC5).

Selbstverständlich treten nicht alle Anwendungsfälle in jedem biologisch Projekt auf und es können theoretisch noch weitere Fälle formuliert werden, wobei die Anwendungsfälle UC1-UC5 nach Auskunft der Domänenexperten im IBF-Projekt typische Fälle abdecken. Im IBF-Umfeld wurden mit [218, 121] Arbeiten publiziert, in denen diese Anwendungsfälle relevant sind. Auch außerhalb des Kontextes des IBF Projekts wurden Arbeiten veröffentlicht, deren Anforderungen durch die Prozesse UC1-UC5

beschrieben werden können [222, 36]. Dabei ist auch das 'Global Biodiversity Assessment' (GBA) aus dem Jahr 1995 zu nennen [103]. In diesem war die Messung von GPS-Daten und die genetische Sequenzierung noch nicht üblich, allerdings können sind die Grundlagen der Kartierung des GBA[3] und die Anforderung der Erfassung von Gensequenzen[4] aus der GBA abgeleitet werden. Damit geben die Anwendungsfälle UC1-UC5 die typische Arbeitsweise in der Biodiversitätswissenschaft wider, die bei Bedarf um weitere Anforderungen ergänzt werden können.

Für die Anwendungsfälle UC1-UC5 wurden Anforderungsprofile formuliert, welche als Grundlage der Evaluation für die Datenstandards in der Biodiversitätsinformatik dienen. Diese Anforderungsprofile sind in Anhang A vollständig aufgelistet und werden zur Prüfung der Vollständigkeit von Datenstandards in Abschnitt 4.4 verwendet.

4.3 Evaluation von Datenstandards

Im folgenden Abschnitt werden die Kriterien für die Evaluation der Datenstandards (siehe Abschnitt 4.4) formuliert und beschrieben, wie diese zur Messung in der Biodiversitätsinformatik implementiert werden.

Für die Evaluation von Datenstandards ist das Kriterium der **Vollständigkeit** von entscheidender Bedeutung, da dieses letztlich darüber entscheidet, ob ein Datenmodell überhaupt nutzbar ist. Dieses wird über die Evaluation eines Datenstandards mit POSE an den Anforderungen der Prozesse der Biodiversitätsinformatik gemessen. Grundlage hierfür sind die Anforderungsprofile der Anwendungsfälle UC1-UC5 aus Abschnitt 4.2.5, welch in Anhang A aufgelistet sind.

Da Domänen wie die Biodiversitätsforschung einem stetigen Wandel unterworfen sind, ist auch die Anpassungsfähigkeit an zukünftige Entwicklungen in einem hohem Maße relevant. Dies wird mit dem Kriterium der **Flexibilität** gemessen und trägt dem Evolutionsgedanken eines Datenstandards Rechnung, wie er in [142, 181] beschrieben ist. Das Kriterium der **Verständlichkeit** durch alle Anwender im Sinne von [168] kann nicht in das Framework integriert werden, da das Verständnis der Anwender in einem so komplexen System wie einem Datenstandard für eine gesamte Domäne nur mit aufwändigen Untersuchungen gemessen werden kann. Allerdings erleichtert die Visualisierung der Anforderungen in Prozessen die Verständlichkeit eines Datenstandards, so dass durch die Evaluation eine Datenstandards mit POSE die Verständlichkeit diese Datenstandards verbessert wird.

[3] Kapitel 7
[4] Kapitel 8

Ein wichtiges Anwendungsfeld für Datenstandards ist die Verwendung bei der Datenübertragung in Infrastrukturen (siehe Kapitel 7.1). Aus diesem Anwendungshintergrund können weitere Kriterien abgeleitet werden. Dabei kommt dem Kriterium der **Redundanzfreiheit** eine besondere Bedeutung zu. Dieses hat das Ziel, das Datenaufkommen zu reduzieren und misst, ob gleichartige Datensätze ohne neuen Informationsgehalt mehrfach übertragen werden müssen. Darüber hinaus wird mit dem Kriterium der **Referenzierbarkeit** gemessen, ob wichtige Elemente des Datenstandards eindeutig referenziert werden können. Dies ist notwendig für die Aktualisierbarkeit der Daten und die Unterstützung von **Data Provenance**. Die Unterstützung von Data Provenance wird dabei selbst als Kriterium aufgenommen. Dieses misst, inwieweit die Herkunft der Daten dokumentiert ist und Veränderungen an Datensätzen erfasst werden können.

Daraus ergeben sich folgende, hierarchische Kriterien für die Evaluation von Datenstandards in der Biodiversitätsinformatik, für deren Bewertungen eine Skala von '++ (sehr gut erfüllt)' bis '−− (überhaupt nicht erfüllt)' verwendet wird:

1. Vollständigkeit
2. Flexibilität
3. Data Provenance
4. Referenzierbarkeit
5. Redundanzfreiheit

Für die Evaluation der **Vollständigkeit** wird mit POSE überprüft, ob ein Datenstandard den Anforderungslisten der Anwendungsfälle UC1 bis UC5 aus Abschnitt 4.2.5 genügt. Dabei kann ein Standard aus einem einzigen Schema bestehen, welches ein Dokument definiert oder aus einer Menge von Schemata für die Erstellung von Dokumenten. Teilweise sind diese Schemata zu umfassend, als dass ein vollständiges Schemaprofil erstellt werden könnte (ABCD hat über 1300 Konzepte). Außerdem wird ein Datenstandard in einer Vielzahl von Prozessen verwendet, so dass Fehler 2. Art unvermeidbar sind. Dementsprechend wird ein Datenstandard danach bewertet, ob die Anforderungsprofile der Prozess UC1 bis UC5 in dem Standard abgebildet werden können und somit keine Fehler 1. Art und 3. Art auftreten. Die Anwendungsfälle sind von absteigender Wichtigkeit, so dass UC1 den wichtigsten Anwendungsfall darstellt. Treten in einem Datenstandard Fehler 1. oder 3. Art im Anforderungsprofil

von UC1 auf[5] wird dieser als ungeeignet bewertet (−−). Ist nur UC1 vollständig erfüllt, führt dies zu der Bewertung (−). Ein Standard gilt als eingeschränkt geeignet, wenn die Anforderungen der Profile aus UC1 und UC2 vollständig vollständig erfüllt sind. Ein Datenstandard ist vollständig (++), wenn er alle Kriterien der Profile aus UC1-UC5 erfüllt. Wird nur ein Anwendungsfall aus UC3 bis UC5 erfüllt, wird der Datenstandard mit '+' bewertet.

In der **Flexibilität** wird überprüft, ob und wie für einen Datenstandard ein Prozess zur Anpassung implementiert ist, wenn sich die Anforderungen an diesen verändern. Damit wird beim Kriterium der Flexibilität keine Eigenschaft eines Datenschemas an sich, sondern seiner Einbettung in organisatorische Strukturen erfasst. Dementsprechend werden bei der Analyse der Flexibilität die organisatorischen Prozesse untersucht, die erforderlich sind, um das Datenmodell an neue Anforderungen anzupassen. Dabei wird das Kriterium als nicht erfüllt betrachtet (−−), wenn kein Prozess definiert ist, der es erlaubt, neue Anforderungen in das Datenmodell aufzunehmen. Das Kriterium wird als teilweise erfüllt betrachtet $(-, 0, +)$, wenn Ausführung des Anpassungsprozesses so langwierig ist, dass dieser nicht in einer annehmbaren Zeit zu einem Ergebnis führt (z.B. umfangreiche Ratifizierung). Die konkrete Bewertung hängt vom Aufwand der Anpassung ab. Das Kriterium gilt als erfüllt, wenn ein transparenter Prozess die Einführung neuer Anforderungen zeitnah gestattet (++).

Für die Unterstützung von **Data Provenance** wird ein Datenstandard danach untersucht, ob in diesem folgende Daten dokumentiert werden können:

DP1 Ursprung von Datensätzen (Bearbeiter)

DP2 Datum der Speicherung des Datensatzes

DP3 Versionen von Datensätzen

DP4 Veränderung an Datensätzen (z.B. Maßeinheit bei Messungen)

Sind alle diese Kriterien erfüllt, wird die Unterstützung von Data Provenance mit (++) bewertet. Das Fehlen einer dieser Möglichkeiten, wird mit einer Abwertung um eine Stufe bewertet (bis maximal '−−'). Das DP2 dokumentiert, wann ein Datensatz in einem Datenspeicher angelegt wurde, nicht wann der Prozess der Datenerhebung ausgeführt wurde. Genauso muss für die Erfüllung von DP1 gespeichert werden, wer den Datensatz gespeichert hat, nicht wer die Daten erhoben hat.

[5]Mit Ausnahme der Genauigkeit der GPS-Messung im operationalen Dokumentenaspekt und der Erfassung von Referenzen über eine URL, welche separat bewertet werden

Das Kriterium der **Referenzierbarkeit** misst, ob im Datenstandard ein Mechanismus zur Sicherung der Identität von Datensätzen und einzelnen Attributen innerhalb einer Infrastruktur implementiert ist. Dabei muss ein Datensatz nicht nur in einem Datenspeicher sondern für die gesamte Infrastruktur (z.B. über eine URN) eindeutig sein. Die Referenzierbarkeit von folgenden primären Konzepten muss von einem Datenstandard nicht nur ermöglicht, sondern auch durchgesetzt werden:

- Beobachtungen und Belege
- taxonomische Identifikation
- Multimediaobjekte
- Personen

Können für alle primären Konzepte Referenzen gesetzt werden und werden diese auch durchgesetzt, wird ein Datenstandard mit (+) bewertet. Ist die Angabe von Referenzen nur fakultativ, führt dies zur Abwertung um eine Stufe. Auch das Fehlen eines primären Konzepts führt jeweils zur Abwertung um eine Stufe. Erfüllt ein Datenstandard alle Kriterien für eine gute Bewertung und werden zusätzlich sekundäre Kriterien, wie die Referenzierbarkeit von veränderlichen persönlichen Daten wie der Telefonnummer oder aber Orten unterstützt, wird ein Standard mit (++) bewertet.

Das Kriterium der **Redundanzfreiheit** gilt als erfüllt, wenn jedes Konzept genau einmal auftritt und dieses aus anderen Konzepten referenziert wird (++). Für jedes redundante Auftreten eines der primären Konzeptes in einem Datenstandard folgt eine Abwertung um eine Stufe. Damit ist dieses Kriterium ein Maß für den Grad der Normalisierung eines Datenstandards.

4.4 Vorstellung und Evaluation der Datenstandards in der Biodiversitätsinformatik

Im folgenden Abschnitt werden Standards zur Datenspeicherung und -übertragung aus dem Bereich der Biodiversitätsinformatik evaluiert. Die Auswahl orientiert sich an der Relevanz des Standards für die Biodiversitätsinformatik. Außerdem werden interessante neuartiges Ansätze evaluiert, sofern sich ihr Ansatz fundamental von etablierten Standards unterscheidet. Die Standards werden kurz vorgestellt und anschließend evaluiert. Dabei orientiert sich die Reihenfolge der Evaluation grob an der Bedeutung der Standards. Eine Fazit der Evaluation findet sich in Abschnitt 4.4.9.

4.4.1 ABCD

'Access to Biological Collections Data' (ABCD) [229] ist ein Standard, der primär zur Übertragung von Sammlungsdaten als XML-Schema geschaffen wurde. ABCD wurde 2005 in der Version 2.06 von der TDWG [232] ratifiziert und beinhaltet (siehe [237]) Spezifikationen zur Sammlung von lebenden und konservierten Belegen, sowie Beobachtungen im Feld. Primärziel ist die Unterstützung des Austauschs und der Integration von detailliert beschriebenen Sammlung- und Beobachtungsdaten. Ziel des Designs ist es dabei möglichst allgemein und verständlich zu sein [237]. Dabei sind sich die Autoren von ABCD bewusst, dass nur ein Bruchteil der über 1300 spezifizierten Felder in einem konkreten Anwendungsfall benötigt wird. Taxonomische Spezialfälle wie z.B. die Synonymie werden von dem Standard nicht abgedeckt.

ABCD gehört zu den wichtigsten Datenaustauschformaten in der Biodiversitätsinformatik. Insbesondere der Anschluss von GBIF-Knoten an das GBIF-Portal erfolgt über Wrapper (z.B. PyWrapper, BioCase), die mit dem ABCD-Schema arbeiten (siehe [18]). Ein wichtiger Unterstützer des ABCD-Schemas ist das BGBM (siehe Abschnitt 2.3.1), das an der Entwicklung maßgeblich beteiligt ist. Damit hat ABCD seine Ursprünge in der Sammlungsverwaltung.

Vollständigkeit: ABCD organisiert Daten in Form des Konzepts des 'DataSets' als Wurzelelement. Dieses enthält eine Liste von 'DataUnits', welche die eigentlichen Sammlungs- und Beobachtungsdaten beinhalten. ABCD kann UC1 ausreichend erfassen. Für UC2 ist die Berücksichtigung von Belegen in Konzept 400 explizit vorgesehen. Diese wird sonst im wesentlichen analog zu einer Beobachtung behandelt. ABCD kann den Standort eines Belegs im Archiv nicht abbilden, sondern nur die verwaltende Organisation (Fehler 3. Art). Für UC3 kann die Verknüpfung zu Multimedia-Objekten erfasst werden, allerdings nicht Ausführungsort und Zeitpunkt der Aufnahme. Die Referenz auf andere biologische Einheiten ist nur möglich, wenn diese innerhalb desselben 'Datasets' übertragen werden. Die Dokumentation der Arbeit einer Sammlung wie in UC4 ist mit ABCD nicht möglich, da Vorgänge wie das Auspflegen eines Belegs aus einer Sammlung und die Verknüpfung eines externen Datensatzes nicht erfolgen kann. UC5 kann abgebildet werden, sofern alle Daten innerhalb desselben 'Datasets' übertragen werden. Damit ist das Kriterium der Vollständigkeit wohl für UC1 erfüllt, ABCD weist aber in den anderen Anwendungsfällen erhebliche Mängel auf.

Flexibilität: ABCD wird von der TDWG als Standard ratifiziert. Eine Aktualisierung des Standards kann nur auf den jährlichen Treffen der TDWG beschlossen werden. In der aktuellen Form von ABCD in der Version 2.06 wird keine Möglich-

keit angeboten, neue Konzepte in den Standard aufzunehmen, ohne dass dieser über einen entsprechenden Beschluss verändert wird. Die Notwendigkeit der Anpassung des Schemas an künftige Anforderungen war dabei den Entwicklern von ABCD bewusst, da die Möglichkeit Erweiterungen von ABCD zu generieren in [234] explizit vorgegeben ist. Dies stellt aber keine Anpassung des Standards an aktuelle Entwicklungen dar, da Software, die den ABCD-Standard verwendet, so definierte Erweiterungen nicht unterstützen kann. Folglich kann ein Prozess zur Aktualisierung des Standards nur über eine Änderung auf einem TDWG-Meeting definiert werden. Anpassungen an Standards benötigen dabei einen Zeitraum von mehreren Jahren.

Data Provenance: ABCD speichert den Ursprung eines Datensatzes über 'LastEditor' für Units. Der Zeitpunkt der Speicherung des Datensatzes wird über das Konzept 'LastEdited' für Units gespeichert. Der Zeitpunkt der Veränderungen von anderen Konzepten wie z.B. ABCD-Metadaten kann nicht dokumentiert werden. Dies ist aber für den Anwendungszweck von ABD nur eine geringe Einschränkung, so dass DP1 und DP2 als erfüllt betrachtet werden. Eine Möglichkeit der Speicherung einer Versionsnummer von Datensätzen ist gegeben. Allerdings können keine Referenzen auf andere Versionen eines Datensatzes gesetzt werden. Datensätze werden bei der Datenübertragung nicht verfolgt und Veränderungen an diesen können nicht dokumentiert werden. DP3 und DP4 sind deshalb nicht erfüllt.

Referenzierbarkeit: Die Referenzierbarkeit der primären Konzepte wird durch ABCD ermöglicht, aber nicht durchgesetzt. Die Referenzierbarkeit der sekundären Konzepte ist nicht erfüllt. Somit ist das Kriterium der Referenzierbarkeit nur teilweise erfüllt.

Redundanzfreiheit: ABCD hat sich zur Aufgabe gemacht, seine Anwendungsdomäne möglichst vollständig und detailliert zu erfassen. Dazu wird eine Vielzahl von Konzepten von Verantwortlichen wie Name, Telefonnummer oder Emailadresse tief in andere Konzepte eingebettet und wiederholt verwendet. Dies gilt besonders für Konzepte des Themenbereichs 'Intelectual Property Rights' (IPR). Das Kriterium der Redundanzfreiheit ist damit nicht erfüllt.

Bei der Erfüllung des Vollständigkeitskriteriums zeigt ABCD gute Ansätze, die für die Darstellung der Domäne allerdings nicht ausreichend sind. Dies ist nachteilhaft, da keine Strukturen zu einer flexiblen Anpassung des Modells existieren. Da ABCD vor dem Hintergrund der Datenübertragung geschaffen wurde, fällt die redundante Datenspeicherung und die unzureichende Aktualisierbarkeit besonders stark ins Gewicht. Damit sind die Anforderungen an den Standard zu weiten Teilen nicht erfüllt.

4.4.2 DwC

DarwinCore ist ein von der TDWG ratifizierter Standard [232], welcher auf dem Standard des DublinCore aus dem Bibliothekswesen beruht und auch auf Konzepte aus diesem zurückgreift. Er bietet eine Nomenklatur von Begriffen mit dem Ziel, Informationen in der Biodiversitätsdomäne auszutauschen [231]. Damit ist DwC primär ein Übertragungsstandard. Inhaltlich zielt DwC auf taxonomische Konzepte und ihrem Vorkommen in der Natur ab [231]. Die Spezifikation erfolgt als XSD. Damit sind die wesentlichen Objekte, die mit DwC dokumentiert werden sollen, Beobachtungen, Belege und Proben [231]. Dabei muss zwischen dem allgemeinen DwC-Standard und dem sogenannten Simple-DarwinCore unterschieden werden. Letzterer enthält nur eine Teilmenge der Konzepte des allgemeinen DwC. Die Verwendung dieser nach gewissen Regeln einschränkt. Der Simple DarwinCore ist einfacher zu handhaben und deshalb in der Praxis weiter verbreitet – allerdings weniger ausdrucksmächtig. Dementsprechend müssen die beiden Standards bei der Analyse der Vollständigkeit, Verständlichkeit und Redundanzfreiheit getrennt behandelt.

Vollständigkeit: DwC verwendet 'Occurence', 'Event'. 'Location', 'Geological Context', 'Identification', 'Taxon', 'ResourceRelationship' und 'MeaseurementOrFact' als sogenannte Basisklassen, um Daten zu erfassen. Die Verwendung von DwC auf Anwendungsfälle der Biodiversitätsforschung, in denen diese Basisklassen nicht ausreichen, ist damit nicht vorgesehen. Die Klassen 'ResourceRelationship' und 'MeasurementOrFact' stellen Hilfsklassen dar, welche es erlauben, die bilaterale Beziehung zwischen zwei Instanzen der Basisklassen auszudrücken bzw. eine Messung anzugeben. Eine Messung muss anschließend zusätzlich über eine 'ResourceRelationship' einer Instanz einer Basisklasse zugeordnet werden.

DwC: Der DwC hat als Wurzelelement das 'DarwinCoreRecordSet', in welches Instanzen aller Basisklassen eingebettet werden können aber nicht müssen (Minimale Kardinalität 0 für alle Basisklassen). Die Basisklassen desselben 'DarwinCoreRecordSets' können dabei über die 'ResourceRelationShip' zueinander in Beziehung gesetzt werden. Die Art der Beziehung wird dabei textuell in nicht verschlagworteter Form beschrieben. UC1 wird im funktionalen Dokumentenaspekt vollständig unterstützt. Im organisatorischen Dokumentenaspekt fehlt die Möglichkeit, den Kartierer über eine URN zu erfassen, so dass hier ein Fehler 3. Art vorliegt. Die Erfassung des operationalen Aspekts erfolgt über die Möglichkeit, die Messgenauigkeit zu beschreiben. UC1 ist damit in ausreichender Weise erfasst. Die Erfassung des datenorientierten Aspekts für UC2 und die Referenzierung eines Beleges kann mit DwC erfasst werden. Allerdings kann für einen Beleg keine Standort in einer Sammlung erfasst werden, so

dass UC4 nicht erfasst werden kann. Die Referenzierung eines Multimediaobjektes in UC3 ist möglich. In UC5 können zwar Messungen und Messmethode erfasst werden. Die Erfassung der Messmethode ist aber nur in textueller Form vorgesehen. UC2 bis UC5 sind damit nicht ausreichend erfasst.

Simple-Dwc: Der Simple-Dwc hat als Wurzelelement das 'SimpleDarwinCoreRecordSet', welcher eine Menge von 'SimpleDarwinCoreRecords' (SDwCR) beinhaltet. Ein 'SDwCR' beinhaltet Bereiche für alle Basisklassen, in denen beliebige Elemente aller Basisklassen angegeben werden können. Basisklassen selbst dürfen nicht als Felder verwendet werden. Damit kann demselben 'SimpleDarwinCoreRecord' sowohl eine 'OccurenceID' wie auch eine 'EventID' zugeordnet werden, was aus semantischer Sicht nicht sinnvoll ist. Die Unterteilung in verschiedene Basisklassen wird aufgehoben. Damit sich mit Simple-Dwc überhaupt sinnvoll Dokumente strukturieren lassen, muss der Anwender bei der Verwendung von Simple-Dwc gewisse Regeln beachten [264]. Dies führt zu Einschränkungen, so dass es nicht möglich ist, mehr als eine Beziehung oder Messung pro 'SimpleDarwinCoreRecord' anzugeben. Damit ist die Erfassung von UC2 bis UC5 nicht möglich. Für UC1 gelten die gleichen Einschränkungen wie im allgemeinen DwC.

Flexibilität: Als ein von der TDWG zertifizierter Standard kann an einen Aktualisierung des Standards nur auf den jährlichen Treffen der TDWG beschlossen werden. Das Kriterium der Flexibilität ist damit aus denselben Gründen wie bei ABCD nur ansatzweise erfüllt.

Unterstützung von Data Provenance: DwC speichert den Ursprung eines Datensatzes nur über das Institut, welchem ein Kartierer zugeordnet ist. Dabei ist in DwC die Speicherung des Instituts nur über ein Kürzel oder einen Code vorgesehen. Es geht aus der Dokumentation zu DwC nicht hervor, ob eine Mitarbeiter des Instituts auch den Datensatz angelegt hat. Damit ist DP1 nicht erfüllt. DwC speichert in den 'RecordLevelTerms' über das Feld 'modified', wann ein Datensatz zuletzt verändert wurde. DP2 ist damit erfüllt. Eine Möglichkeit der Speicherung von Versionen von Datensätzen ist nicht explizit gegeben. Datensätze werden bei der Datenübertragung nicht verfolgt und Veränderungen an diesen können nicht dokumentiert werden. Das DP3 und DP4 sind deshalb nicht erfüllt.

Referenzierbarkeit: DwC speichert in beiden Varianten Daten des organisatorischen Aspekts in nicht referenzierter Form ab. Die Referenzierung für Taxa wird im funktionalen Dokumentenaspekt ermöglicht, aber nicht durchgesetzt. Die Referenzierbarkeit der sekundären Konzepte ist nicht erfüllt. Somit ist das Kriterium der Referenzierbarkeit nur teilweise erfüllt.

Redundanzfreiheit: Durch die Verwendung globaler Definitionen ist ein in DwC formatiertes Dokument nicht so stark verschachtelt wie ein Dokument in ABCD, da hier Konzepte referenziert werden können. So kann redundante Datenspeicherung vollständig vermieden werden. Dies ist bei Simple-DwC jedoch nicht möglich, so dass bei der Abbildung von Beziehungen Daten in redundanter Form gespeichert werden müssen.

Allgemeiner DwC: Konzepte, die mit den Basisklassen erfasst werden können, können über Referenzierung redundanzfrei gespeichert werden. Allerdings können nicht alle benötigten Konzepte wie z.B. der Verantwortliche im organisatorischen Dokumentenaspekt in den Basisklassen erfasst werden. Entsprechende Konzepte müssen somit redundant gespeichert werden. Das Kriterium ist damit teilweise erfüllt.

Simple-Dwc: Die Struktur des Simple-Dwc zwingt den Anwender bereits in einfachen Anwendungsfällen dazu, für jede benötigte Kombination der Basisklassen ein eigenes 'SimpleDarwinCoreRecord' anzulegen. Es treten zwar innerhalb eines 'SimpleDarwinCoreRecords' keine Felder mit ähnlicher Bedeutung mehrfach auf, es muss aber innerhalb einer Menge von entsprechenden 'Records' immer wieder redundant Information gespeichert werden, da Beziehungen zwischen 'Records' nur eingeschränkt verwendet werden können. Dies führt im Beispiel einer Artenliste an einem Ort dazu, dass alle Einträge, die nicht die Identifikation betreffen, wie z.B. die Informationen der organisatorischen und lokalen Aspekte, für jede identifizierte Art redundant gespeichert werden müssen.

Sowohl der Allgemeine DwC als auch Simple-Dwc implementieren die Anforderungen nicht ausreichend. Insbesondere von der Verwendung des Simple-Dwc ist abzuraten, da die Spezifikation in [264] das Vollständigkeitskriterium nicht erfüllt. Außerdem ist der Simple-DwC schlecht strukturiert und zwingt den Anwender zur redundanten Datenhaltung.

4.4.3 SDD

Structured Descriptive Data (SDD) wurde 2005 von der TDWG als Standard ratifiziert [232] und soll die DELTA (DEscription Language for TAxonomy) als Standard der TDWG für die Erfassung, Austausch und Speicherung für beschreibende Daten ersetzen [236]. Der theoretische Hintergrund zu diesem Standard findet sich in [90]. Beschreibende Daten in diesem Sinne sind Daten über intrinsische Eigenschaften von Organismen, an Hand welcher Individuen, Populationen oder Taxa kontextunabhängig identifiziert werden können [90]. SDD ist in der Version 1.0 von der TDWG ratifiziert [91]. In der aktuellen Version 1.1 [92] ist SDD noch nicht von der TDWG

ratifiziert. Trotzdem soll die Version als aktuelle Variante der Ausgangspunkt der folgenden Evaluation sein. SDD ist über eine XSD spezifiziert und verwendet neben den eigenen Begriffen auch Begriffe aus dem Unified Biosciences Information Framework (UBIF) [92], welcher von der TDWG entwickelt wird.

Vollständigkeit: SDD arbeitet mit dem Konzept 'DataSets' als Wurzelelement, unter welchem Attribute des organisatorischen, lokalen und temporalen Aspekts direkt angegeben werden können. Unter dieser Ebene steht eine Menge von Containern zur Speicherung der eigentlichen Datensätze ('DataSet') zur Verfügung. Innerhalb dieser können Daten zu Taxonomien, Belegen und Beschreibungen gespeichert werden. Für die Aufnahme der beschreibenden Daten sind dabei sogenannte 'Characters' vorgesehen, die auch die Speicherung von Messungen ermöglichen und zueinander in Beziehung gesetzt werden können. Ein 'DataSet' kann dabei beliebig viele dieser Elemente aufnehmen und diese zueinander in Beziehung setzen. Da SDD keine Geoinformationen speichert, kann SDD die Anforderungen des lokalen Aspekts aus UC1 bis UC3 nicht erfassen. Die Erlassung von UC4 ist nicht möglich, da SDD die Sammlungsverwaltung von Belegen nicht unterstützt. Allerdings kann SDD UC5 bis auf den lokalen Dokumentenaspekt gut erfassen. UC5 liegt auch im primären Anwendungsgebiet von SDD.

Flexibilität: Als ein von der TDWG zertifizierter Standard kann eine Aktualisierung des Standards nur auf den jährlichen Treffen der TDWG beschlossen werden. Der Standard hat aber über seine Spezifikation die Möglichkeit über 'Charakters', neue Arten von beschreibenden Daten in das Modell aufzunehmen. Damit kann praktisch jede Art von beschreibenden Daten in das Modell mit aufgenommen werden und SDD ist für seinen primären Anwendungszweck tatsächlich flexibel. In allen anderen Bereichen können Änderungen aber nur durch eine Änderung der Spezifikation von SDD über die TDWG erfolgen.

Data Provenance: SDD speichert die Herkunft von Datensätze über die Konzepte 'Technical Metadata' und 'DataSet Metadata' [236]. In den 'Technical Metadata' wird dabei der Entstehungszeitpunkt eines Dokuments über einen Zeitstempel gespeichert. DP2 ist damit erfüllt. In den 'DataSet Metadata' ist es möglich, die Urheber eines Datensatzes zu dokumentieren. Darüber hinaus wird an dieser Stelle die Eigentümerschaft eines Datensatzes gespeichert. DP1 ist deshalb erfüllt. Eine Möglichkeit der Speicherung von Versionen von Datensätzen ist nicht gegeben. Datensätze werden bei der Datenübertragung nicht verfolgt und Veränderungen an diesen können nicht dokumentiert werden. Das DP3 und DP4 sind deshalb nicht erfüllt.

Referenzierbarkeit: In SDD können auf Beobachtungen und Belege keine Referenzen gesetzt werden. Die Angabe von Personen erfolgt in den 'Metadata' in Freitextform. Das Setzen von Referenzen wird dadurch ermöglicht aber nicht durchgesetzt. Für die Identifikation von Taxa wird ein Identifier benötigt, der allerdings auch frei eingegeben werden kann. Auf die selbe Weise wird mit Multimediadaten umgegangen. Damit kann zwar durchgängig eine Referenz angegeben werden, allerdings ist die Eindeutigkeit in einem Netzwerk nicht garantiert und das Kriterium der Referenzierbarkeit damit nur in Ansätzen erfüllt (-).

Redundanzfreiheit: SDD unterstützt die interne Referenzierung für die Elemente eines 'Datasets' und im organisatorischer Dokumentenaspekt die externe Referenzierung. Allerdings werden neben diesen Referenzen auch weitere Elemente mitgeführt. Das Datenschema ist damit teilweise redundant und das Kriterium teilweise erfüllt.

SDD zeigt gute Ansätze in Bezug auf Referenzierbarkeit und Flexibilität und ist insbesondere zu seinem Primärzweck der Speicherung von beschreibenden Daten gut geeignet. Allerdings ist die Vollständigkeit insbesondere im Bezug auf die Anwendungsfälle nicht gegeben, da SDD Georeferenzierung nicht unterstützt. Somit kann SDD als allgemeines Schema für die Biodiversitätsinformatik nicht empfohlen werden.

4.4.4 CDM

Das Common Data Model (CDM) geht auf das EDIT (European Distributed Institute for Cypertaxonomy)-Projekt zurück [59] und dient als Grundlage für die im EDIT-Projekt entwickelte Software. Der Anwendungsbereich ist damit die Verwendung in Softwareprojekten. Eine direkte Anwendung von CDM zur Datenübertragung ist nicht möglich. CDM ist aktuell in der Version 3.1 [61] als Klassenbibliothek für Java verfügbar. EDIT wurde maßgeblich vom BGBM entwickelt, welches ein Teilnehmer des EDIT-Projektes ist [59]. Das CDM ist über UML spezifiziert. Die CDM ist OpenSource und steht über [60] zum Download zur Verfügung. Inhaltlich fußt CDM nach [60] auf den Konzepten der TDWG Ontology [230].

Vollständigkeit: Die Klassen des CDM [61] gestatten bis auf die Ungenauigkeit der GPS-Messung im operationalen Dokumentenaspekt die Erfassung des Anforderungsprofils von UC1. In UC2 ist die Referenzierung eines Belegs möglich. Die Verlinkung von Multimediaobjekten in UC3 kann vollständig erfasst werden. Für UC4 kann die Referenz auf ein externes DNA-Dokument gesetzt werden. Im lokalen Dokumentenaspekt kann nur das Institut angegeben werden und nicht der genaue

Ort in der Sammlung. Die Erfassung des Auspflegens aus dem Katalog der Sammlung ist nicht möglich. UC4 ist damit nicht erfasst. Für UC5 können die Messungen vollständig erfasst werden. Die Möglichkeit zum Setzen von Referenzen auf andere Objekte wie in UC3 und UC5 gefordert, konnte im Rahmen der Evaluation nicht identifiziert werden.

Flexibilität: CDM ist ein proprietärer de facto Standard, der als Klassenbibliothek den Wissenschaftlern in der Biodiversitätsforschung zur Verfügung gestellt wird. CDM ist OpenSource, so dass theoretisch auf individueller Basis Erweiterungen geschrieben werden können. Allerdings wird kein Prozess zur flexiblen Erweiterung des Modells für die Community spezifiziert. Damit ist das Kriterium nicht erfüllt.

Data Provenance: CDM ermöglicht die Speicherung des Urhebers eines Datensatzes über die Klasse 'Person'. In dieser kann eine Person auch einem Institut zugeordnet werden. CDM ermöglicht die Speicherung des Zeitpunkts der Datenerhebung. Allerdings wurde keine Möglichkeit gefunden in CDM den Zeitpunkt der Speicherung des Dokumentes oder des Urhebers von Modifikationen an einem Dokument zu speichern. Es konnte auch keine Möglichkeit identifiziert werden, verschiedene Versionen eines Datensatzes zu speichern oder Veränderungen an Datensätzen zu dokumentieren. Die Kriterien der Unterstützung von Data Provenance sind deshalb nicht erfüllt.

Redundanzfreiheit: Die Klassen sind thematisch in disjunkte Pakete sinnvoll aufgeteilt. Innerhalb der Pakete werden die Themenbereiche durch die verwendeten Klassen sauber gegliedert. Das Kriterium der Redundanzfreiheit ist damit erfüllt.

Referenzierbarkeit: CDM unterstützt keine Referenzierung von Objekten durch eine URL oder URN. Das Kriterium ist damit nicht erfüllt.

CDM besitzt als Klassenbibliothek eine gute Abdeckung der Domäne der Biodiversitätsforschung, wenngleich auch die Wartbarkeit über Schnittstellen mit anderen Softwareprodukten nur schwer möglich ist. Eine besondere Schwierigkeit bei der Verwendung von CDM für die Datenübertragung liegt darin, dass CDM nur als Java-Klassenbibliothek verfügbar ist. Damit ist die Verwendung als Standard zur Datenübertragung an eine bestimmte Technologie gebunden.

4.4.5 DiversityCollection (DC)

Das Datenschema von 'DiversityCollection' (DC) [261] ist Grundlage der 'Diversity Workbench', welche an den 'Staatlichen naturwissenschaftlichen Sammlungen Bayerns' (SNSB) in München entwickelt wird [50]. Ziel von 'DiversityCollection' ist die Sammlungsverwaltung und Speicherung von Beobachtungsdaten aus der Biodiversi-

tätsforschung. Der Fokus liegt damit auf der Speicherung der Daten und der Verwendung des Datenmodells in der 'Diversity Workbench'. Da die Anwendungsfälle UC1-UC5 innerhalb des IBF-Projekt aufgetreten sind und dieses in Zusammenarbeit mit dem SNSB und der 'Diversity Workbench' erfolgt, wurde das Datenmodell an diese Anwendungsfälle angepasst.

Vollständigkeit: DiversityCollection kann UC1 bis UC3 vollständig erfassen. Die Erfassung von UC4 ist nicht möglich, da im datenorientierten Dokumentenaspekt keine Referenzen auf externe Dokumente zur Speicherung von Genanalysen gesetzt werden können. UC5 kann vollständig erfasst werden. Allerdings muss ein Anwender von DiversityCollection eine Vielzahl von Attributen pflegen, die nicht in den Anforderungsprofilen enthalten sind (Fehler 2. Art). Diese wirken sich allerdings nicht auf die Bewertung aus. Damit kann ist das Ergebnis der Vollständigkeitsbewertung von DC '+'.

Flexibilität: DC ist ein proprietärer de facto Standard, der vom SNSB entwickelt wird. Die Software der 'Diversity Workbench' ist OpenSource, wird aber nur auf Anfrage zur Verfügung gestellt. Es wird kein Prozess zur flexiblen Erweiterung des Modells für die Community spezifiziert und das Datenmodell auch nur ausschließlich intern weiterentwickelt. Damit ist das Kriterium nicht erfüllt.

Data Provenance: DC ermöglicht die Speicherung des Urhebers eines Datensatzes und den Änderungszeitpunkt eines Datensatzes durchgängig im Datenmodell. Das Institut eines Kartierers kann nicht gespeichert werden. Dies ist aber nicht notwendig, da DC speziell für das SNSB konzipiert wurde und somit auch alle Kartierer diesem Institut zugeordnet sind. Das DP1 und DP2 sind damit erfüllt. Eine Möglichkeit der Speicherung von Versionen von Datensätzen ist nicht explizit gegeben. Datensätze werden bei der Datenübertragung nicht verfolgt und Veränderungen an diesen können im Allgemeinen nicht dokumentiert werden. Eine Ausnahme ist die taxonomische Bestimmung, da für diese eine Historie angelegt werden kann. Das DP3 und DP4 sind deshalb nicht erfüllt.

Redundanzfreiheit: Die Datenstruktur in DiversityCollection ist über Tabellen ausgehend von der Tabelle 'Specimen' organisiert. Ausgehend von dieser Tabelle wiederholen sich Felder zur Speicherung derselben Information im organisatorischen und lokalen Dokumentenaspekt. Innerhalb der einzelnen Tabellen werden Daten im temporalen Dokumentenaspekt häufig redundant gespeichert. Das Kriterium der Redundanzfreiheit ist damit nicht erfüllt.

Referenzierbarkeit: DiversityCollection unterstützt die Referenzierung der taxonomischen Identifikation, Personen und Multimediaobjekten aber nicht für Be-

obachtungen und Belege. Die Referenzierung wird dabei teilweise durchgesetzt und es wird auch die Referenzierung von sekundären Konzepten wie Geoinformationen ermöglicht. Dies wirkt sich allerdings nicht auf die Bewertung aus, da nicht alle primären Konzepte referenzierbar sind. Das Ergebnis in diesem Kriterium für DC ist damit '0'.

Das Datenmodell von DiversityCollection zeichnet sich durch eine gute Abdeckung der Vollständigkeit in den Anwendungsfällen aus. Dies ist aber insbesondere durch die Zusammenarbeit mit dem SNSB in dem IBF-Projekt bedingt. Die Anpassung an neue Anforderungen kann nur durch direkten Kontakt mit den Entwicklern erfolgen. Die Hauptmängel des Schemas liegen in der redundanten Speicherung von Informationen sowie dem inkonsistenten Aufbau.

4.4.6 NBN Exchange Format

Das National Biodiversity Network Exchange Format (NBNEF) ist das von der NBN verwendete Datenaustauschformat und arbeitet textbasiert [176] mit tab-separierten Dateien. Die Daten werden in einem Dokument nach dem in NBNEF spezifizierten Schema in Spalten organisiert, von denen einige als 'notwendig' gekennzeichnet sind. Intern müssen in NBNEF Primärschlüssel der Datenbankverwaltung für Taxa und Orte des NBN referenziert werden, welche das NBN nur auf Anfrage zur Verfügung stellt. Es besteht optional die Möglichkeit, in einen XML-Header die übertragenen Datensätze genauer zu beschreiben. Inhaltlich hat das NBNEF die Aufgabe, Beobachtungen von Arten zu dokumentieren [176].

Das NBNEF ist ein nationaler, proprietärer Standard, der vom NBN insbesondere zur Speicherung von Daten aus Großbritannien verwendet wird. Daten werden dabei über das NBN-Portal der Öffentlichkeit zur Verfügung gestellt. Die NBN Community arbeitet unabhängig von der TDWG, so dass Exporte der Daten nach DwC oder ABCD nicht vorgesehen sind. Der Export in NBNEF wird aber von Software zur Datenaufnahme wie 'Recorder' oder 'MapMate' unterstützt [176], welche in der Biodiversitätsforschung in der Praxis verwendet werden.

Vollständigkeit: NBNEF kann UC1 ausreichend erfassen. UC2 und UC4 können nicht erfasst werden, da NBNEF als reiner Standard für Beobachtungen die Verwaltung von Sammlungen nicht unterstützt. UC3 kann nicht erfasst werden, da die Referenzierung anderer Datensätze nicht möglich ist und die Referenzierung von Multimediaobjekten nicht möglich ist. UC5 kann nicht erfasst werden, da die Beziehung zwischen biologischen Objekten nicht erfasst werden können.

Flexibilität: Der NBNEF ist ein proprietäres Datenschema, für welches kein

Prozess zur Anpassung publiziert wurde. Das Schema ist über eine oder mehrere Hilfsspalten, welche als 'Attributes' bezeichnet werden, erweiterbar. Diese werden als Erweiterung des speziellen Datensatz betrachtet, so dass keine Vergleichbarkeit zu anderen Datensätzen mit denselben Attributen besteht. Damit ist die Semantik der Hilfsspalten und ihre Verwendung in Softwareprodukten unklar und das Kriterium ist nicht erfüllt.

Data Provenance: NBNEF kann über die Metadaten nur einen Namen und eine ID für einen Datensatz angeben. Der Zeitpunkt der Speicherung und der Verantwortliche können nicht gespeichert werden. Auch weitere Konzepte zur Unterstützung von Data Provenance fehlen. Damit kann kein Kriterium der Unterstützung von Data Provenance erfüllt werden.

Redundanzfreiheit: NBNEF ist thematisch nach den Aspekten von POSE unterteilt. Konzepte werden dabei nicht wiederholt verwendet. Das Kriterium der Redundanzfreiheit ist damit erfüllt.

Referenzierbarkeit: NBNEF unterstützt Referenzierung im funktionalen und lokalen Dokumentenaspekt mit proprietären Schlüsseln. Damit können keine Referenzen in einer Form gesetzt werden, dass diese für die Datenübertragung in einer Infrastruktur verwendet werden kann.

Da das NBNEF das Kriterium der Vollständigkeit nicht erfüllt und auch keine Möglichkeit zur Erweiterbarkeit bietet, ist es als allgemeines Schema für die Biodiversitätsinformatik ungeeignet. Nichtsdestotrotz zeichnet sich das NBNEF durch eine einfache aber gut gewählte Struktur aus. Nachteilig ist aber die Referenzierung über interne Schlüssel, welche Nutzern nur auf Anfrage zur Verfügung gestellt werden, da hierdurch die Daten speziell auf das NBNEF zugeschnitten werden.

4.4.7 EML

Die 'Ecological Metadata Language' (EML) ist ein vom 'Knowledge Network for Biocomplexity' (KNB) publizierter proprietärer Standard, der primär im Bereich der Ökologie verwendet wird [128]. Der Standard an sich ist dabei aber ausschließlich für die Aufnahme von Daten geeignet, welche im Bereich der Biodiversitätsinformatik als Metadaten bezeichnet werden. Unter diesem Begriff werden Daten des organisatorischen, operationalen und lokalen Aspekts verstanden, welche als Kopfdaten einer Sammlung von Datensätzen fungieren. Die Aufnahme von Primärdaten an sich ist nicht vorgesehen. Diese werden über eine externe Datei (Excel, csv, oder ein freies Format) den Metadaten zugefügt. Dementsprechend enthält EML für Primärdaten kein Datenschema, das evaluiert werden könnte. Aufgrund dieser Einschränkung wird

auf eine Analyse von EML verzichtet, da die Anforderungen an die Vollständigkeit und Referenzierbarkeit der Anwendungsfälle von vornherein nicht erfüllt werden können. Als proprietärer Standard erfüllt EML auch die Anforderungen in Bezug auf die Flexibilität nicht und unterstützt Data Provenance nur unzureichend, da nur das erste und zweite Kriterium der Data Provenance-Unterstützung erfüllt werden. Die Metadaten werden nicht redundanzfrei erfasst. EML ist damit nicht als Datenstandard für die Biodiversitätsinformatik geeignet.

4.4.8 OBOE mit EML

In der Ökologie hat in letzter Zeit die 'Extensible Observation Ontology' (OBOE) nach [149] zur Dokumentation von Beobachtungen eine gewisse Bedeutung erlangt. Der Fokus von OBOE ist die Dokumentation von Beobachtungen. Technisch basiert OBOE auf einer Ontologie. OBOE wird dabei z.B. in Projekten wie dem 'Semantic Tools Project' des 'National Center for Ecological Analysis and Synthesis (NCEAS)' der Universität Santa Barbara verwendet [178]. Durch die Organisation der Daten in einer Ontologie ist die Erfassung von beschreibenden Daten in Form von Messungen sehr gut möglich. Ein weiterer Vorteil der Organisation als Ontologie besteht in der Möglichkeit, Konzepte zu verknüpfen und somit die semantische Relation von Konzepten darzustellen (z.B. Vergleichbarkeit von Messungen in Metern und Zentimetern). OBOE ist in der Grundstruktur abstrakter als die bisher betrachteten Standards, da in OBOE keine konkreten Elemente zur Beschreibung der Domäne entwickelt wurde, sondern eine Metastruktur, welche die Einbettung der Domäne in diese ermöglicht. Dieser Prozess wird in [149] beschrieben.

Da OBOE andere Dokumentenaspekte wie den organisatorischen Dokumentenaspekt nicht unterstützt, wird in [149] die kombinierte Anwendung von OBOE mit EML empfohlen. Da sich OBOE und EML gut ergänzen, soll in diesem Abschnitt der kombinierte Einsatz evaluiert werden.

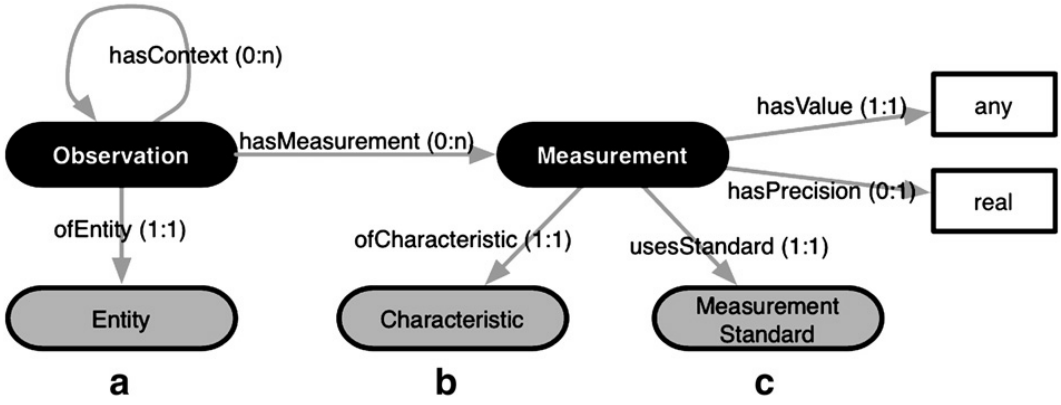

Abbildung 4.17: Grundstruktur von OBOE nach [149]

Vollständigkeit: OBOE hat seinen Anwendungsschwerpunkt in der Dokumentation von Beobachtungen. Dazu schafft OBOE keine feste Struktur zur Organisation der Daten, sondern bettet diese in eine Metastruktur ein, in welcher Daten im funktionalen Dokumentenaspekt einem Datensatz über eine Messung zugefügt werden. So wird die taxonomische Bestimmung als eine Messung des biologischen Objekts betrachtet, welches als Wert die taxonomische Bezeichnung trägt. Prinzipiell kann über die Anbindung als Messung jedes beliebige Konzept in das Schema integriert werden. Dies soll aber nur soweit erfolgen, wie dies durch die Bedeutung des Begriffs 'Messung' gerechtfertigt erscheint. Eine Verknüpfung mit einem externen Dokument wäre dafür ein Gegenbeispiel. Da OBOE den organisatorischen Dokumentenaspekt nicht unterstützt, wird zur Evaluation in diesem Bereich EML evaluiert.

Durch die Modellierung der 'Attributes' des funktionalen und lokalen Aspekts als Messung kann OBOE mit EML das Anforderungsprofil von UC1 vollständig erfassen. Da eine Referenz im datenorientierten Dokumentenaspekt nicht als Messung interpretiert werden kann, ist die Erfassung von UC2 und UC3 im Bezug auf die Referenzierung des Belegs bzw. des Multimediaobjektes nicht möglich. Allerdings kann in UC3 die Referenzierung zwischen biologischen Objekten erfasst werden. Da keine Sammlungsverwaltung unterstützt wird, kann das Anforderungsprofil von UC4 nicht erfasst werden. UC5 bietet sehr gute Anwendungsmöglichkeiten für die in OBOE verwendete Modellierung im funktionalen Dokumentenaspekt als Messung. UC5 kann vollständig erfasst werden. Damit werden insgesamt zentrale Anforderungen nicht erfüllt und OBOE mit EML kann die Anwendungsfälle nicht ausreichend erfassen.

Flexibilität: OBOE ist in der Grundstruktur abstrakter als die bisher betrachteten Standards. Da für OBOE keine konkreten Elemente zur Beschreibung der Domäne entwickelt wurden, sondern eine Metastruktur (siehe Abschnitt 5.2), erlaubt

OBOE mit der Definition neuer Messungen und Entitäten eine stetige Erweiterung des Datenmodells. Diese ist aber durch die Metastruktur von OBOE auf diese beiden Konzepte beschränkt. OBOE ist damit sehr flexibel und es können mit gutem Abstraktionsvermögen viele Sachverhalte in der Struktur von OBOE modelliert werden. Für EML hingegen ist kein Prozess zur flexiblen Anpassung definiert, so dass das Kriterium als teilweise erfüllt gewertet wird.

Data Provenance: Die Unterstützung von Data Provenance erfolgt über die Erfassung von Metadaten in EML. EML gestattet es, über das Modul 'eml-dataset' den Autor eines Datensatzes, den Zeitpunkt der Änderung und eine Historie der Änderungen eines Datensatzes zu dokumentieren. Das DP1 und DP2 sind damit erfüllt. Es ist allerdings nicht möglich mit EML Versionen von Datensätzen zu verwalten und Veränderungen zu dokumentieren, so dass DP3 und DP4 nicht erfüllt sind.

Referenzierbarkeit: Durch Modellierung als Ontologie ist die Aktualität der in OBOE beschriebenen Konzepte stets gewährleistet, da Konzepte in Ontologien stets über eine URL identifiziert werden. Dies ist für die Daten z.B. des organisatorischen Aspektes in EML allerdings nicht der Fall. Das Kriterium der Referenzierbarkeit ist damit teilweise erfüllt.

Redundanzfreiheit: Da mit OBOE die eigentliche Modellierung der Domäne dem Anwender überlassen wird, liegt es in der Verantwortung des Anwenders seinen Anwendungsfall redundanzfrei zu modellieren. Das Kriterium ist auf OBOE nicht anwendbar.

OBOE liefert einen interessanten Ansatz zur Speicherung und Übertragung von Daten der Biodiversitätsforschung. Insbesondere ist die Erstellung von flexiblen Strukturen aufgrund des abstrakten Ansatzes möglich und man kann OBOE als ein Metamodell für die Biodiversitätsinformatik betrachten. Darin liegt aber auch zugleich die Herausforderung von OBOE, da dem Anwender in der Biodiversitätsforschung mit der Erstellung eines Schemas aus einem abstrakteren Metaschema eine schwierige Aufgabe gestellt wird. Insbesondere stellt OBOE keine vordefinierten Konzepte zur Verfügung, so dass sich der Nutzer zwangsläufig mit der Modellierung von Konzepten aus der Metastruktur auseinandersetzen muss.

4.4.9 Ergebnis

In vorangegangen Abschnitt wurden die wichtigsten Standards zur Datenspeicherung und -übertragung auf ihre Eignung als Standard für die Gesamtdomäne untersucht. Es konnte gezeigt werden, dass die in Abschnitt 4.3 gestellten Kriterien von keinem Standard zufriedenstellend erfüllt werden konnten. Dies ist insbesondere beim

	Vollständigkeit	Flexibilität	Data Provenance	Referenzierbarkeit	Redundanzfreiheit
ABCD	-	-	0	0	- -
DwC	-	-	0	0	+ +
Simple DwC	-	-	0	0	- -
SDD	- -	0	0	-	0
CDM	+	- -	- -	- -	+ +
DC	+	- -	0	0	-
NBNEF	-	- -	- -	- -	+ +
EML + OBOE	-	0	0	0	N/A

Tabelle 4.4: Ergebnis der Evaluation

Vollständigkeitskriterium kritisch, da durch die Erfüllung diese Kriteriums die Anwendbarkeit eines Standards erst ermöglicht wird. Das Ergebnis der Evaluation ist in der Tabelle 4.4 dargestellt. Dabei ist dieses Ergebnis eine umfassende Evaluation von Datenstandards in der Biodiversitätsinformatik, die bisher noch nicht in dieser Ausführlichkeit ausgeführt wurde. Damit liefert diese Arbeit einen umfassenden Überblick über die Ist-Situation der Datenstandards in der Biodiversitätsinformatik.

Die Evaluation zeigt, dass keiner der untersuchten Datenstandards die Anforderungen der Biodiversitätsinformatik im Bezug auf die Datenspeicherung unterstützt. Insbesondere zur Unterstützung der Flexibilität sind die aktuellen Datenstandards ungeeignet und auch im Bezug auf die Vollständigkeit erfüllen die existierenden ´Datenstandards nicht die Anforderungen. Dementsprechend muss ein neuer Datenstandard entwickelt werden, der die Anforderungen der Biodiversitätsinformatik erfüllt. Dieser wird in Kapitel 5 mit 'PODSL-Biodiv' eingeführt.

Kapitel 5

PODSL-Biodiv: Ein erweiterbarer Datenstandard für die Biodiversitätsinformatik

Im folgenden Kapitel wird mit der 'Process Oriented Data Schema Language' (PODSL) ein Metamodell zur Formulierung eines Datenstandards in der Biodiversitätsinformatik eingeführt. Auf Basis dieses Metamodells wird mit PODSL-Biodiv ein flexibler Datenstandard für die Anwendung in der Biodiversitätsinformatik geschaffen.

Dazu wird in Abschnitt 5.1 zunächst der Kontext von PODSL dargestellt. Durch diese werden Anforderungen an PODSL formuliert, die sich durch Metamodellierung lösen lassen. In Abschnitt 5.2 werden die Grundlagen der Metamodellierung und das 'Open MetaModeling Environment' (OMME) zur Erstellung von Metamodellen vorgestellt. Anschließend werden die zentralen Anforderungen an PODSL und Konzepte zu deren Umsetzung in Kapitel 5.3 formuliert. Diese dienen als Grundlage für die Entwicklung von PODSL und PODSL-Biodiv, die in Abschnitt 5.4 vorgestellt werden.

In Abschnitt 5.4.3 wird mit PODSL-Biodiv ein Datenstandard für die Biodiversitätsinformatik in PODSL formuliert. PODSL-Biodiv stellt einen neuen Ansatz für einen Datenstandard in der Biodiversitätsinformatik dar. Die Eigenschaften von PODSL-Biodiv werden anschließend in Abschnitt 5.5 vorgestellt. Der Vorteil gegenüber existierenden Ansätzen liegt in der Flexibilität des Datenstandards und der Fokussierung auf Prozesse zur Abbildung der Anwendungsdomäne. Anschließend wird die Übertragung in andere Datenstandards in Abschnitt 5.6 besprochen. Dabei wird insbesondere auf das Mapping zwischen PODSL-Biodiv und DwC eingegangen, welches in Anhang B aufgelistet ist. PODSL-Biodiv wird anschließend in Abschnitt 5.7

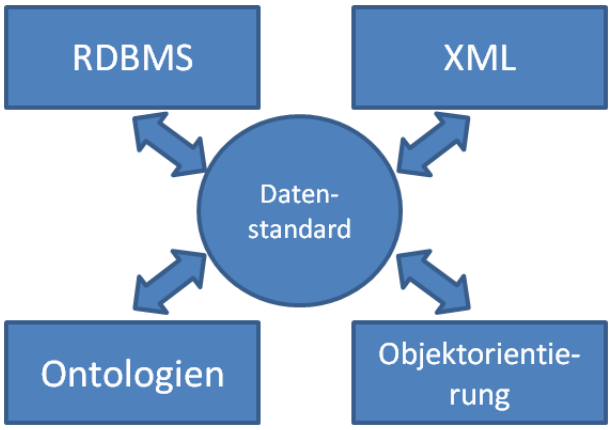

Abbildung 5.1: Spannungsfeld eines Datenstandards in der Biodiversitätsinformatik

mit dem Evaluationsframework aus Kapitel 4 bewertet. PODSL-Biodiv erfüllt die Kriterien des Evaluationsframeworks gut. Abschließend wird ein Überblick über die Ergebnisse diese Kapitels in Abschnitt 5.8 gegeben.

5.1 Kontext von PODSL

In der Biodiversitätsinformatik werden verschiedene Technologien zur Datenspeicherung eingesetzt. Ein Datenstandard für diese Domäne befindet sich dementsprechend in einer Konfliktsituation, wie sie in Abbildung 5.1 dargestellt ist. Der Datenstandard muss mit Datenspeichern und Programmen auf Basis von verschiedenen Technologien kommunizieren. Diese sind aber nur bedingt zueinander kompatibel oder plattformspezifisch. Der Datenstandard muss den Austausch von Daten über diese Technologiegrenzen hinweg ermöglichen.

Einen Ansatz zum technologieübergreifenden Datenaustausch bieten 'Objekt Relationale Mapper' (ORM) wie Hibernate [124, 12]. Hibernate ermöglicht den Übergang zwischen relationalen Datenbanken und objektorientierter Programmierung oder XML und objektorientierter Programmierung [124]. ORM´s arbeiten aber im Allgemeinen plattformspezifisch. So arbeitet Hibernate z.B. auf der Basis von Java [124] [1].

Es stellt sich die Frage, wie der Datenaustausch über Technologiegrenzen erfolgen kann. Eine Lösung hierfür bietet die Metamodellierung, wie in Abbildung 5.2 dargestellt ist. Die Datenmodelle aus den verschiedenen Technologiebereichen werden durch ein moderierendes Metamodell ineinander abgebildet. Dies ist wichtig, wenn Daten aus einer Datenbank in einem Computerprogramm verwendet werden sollen.

[1] Es wird mit NHibernate auch eine .NET Variante angeboten.

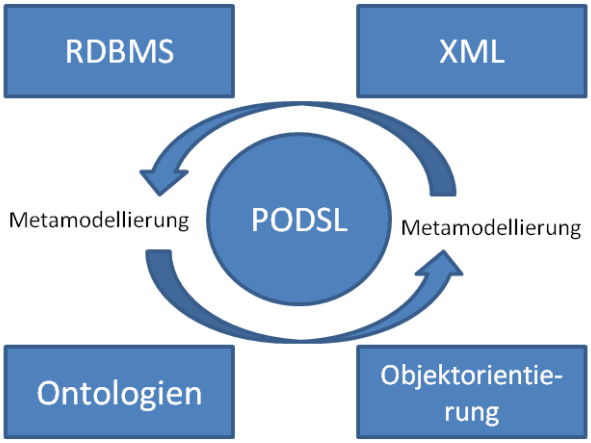

Abbildung 5.2: Metamodellierung als Lösung des Konflikts

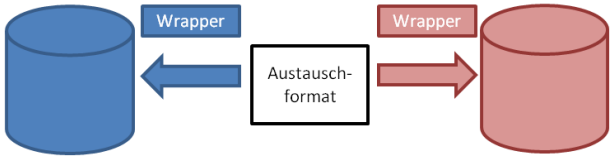

Abbildung 5.3: Datenaustausch zwischen heterogenen Modellen

Damit muss PODSL mit den spezifischen Anforderungen der Objektorientierung, XML, relationalen Datenbanken und Ontologien umgehen können.

Darüber hinaus ist die Datenübertragung ein wichtiges Anwendungsgebiet von PODSL. PODSL muss zwischen verschiedenartigen Schemata moderieren können, wie in Abbildung 5.3 dargestellt ist. Ein Austauschformat in PODSL muss dazu die Entitäten beider Datenspeicher erfassen können. Die Anpassung an die lokalen Datenmodelle wird von Wrappern übernommen. Dabei ist unerheblich welche Technologien die Datenspeicher verwenden. Diese Situation ist insbesondere in der Biodiversitätsinformatik anzutreffen, da in dieser Domäne eine Vielzahl von heterogenen Datenspeichern miteinander kommunizieren müssen. Dementsprechend wurde sich bei PODSL darauf konzentriert, den Datenaustausch zwischen Datenspeichern mit heterogenen Schemata zu ermöglichen.

5.2 Metamodellierung

Im folgenden Abschnitt wird als Exkurs eine kurze Einführung in die Metamodellierung gegeben, soweit dies für das weitere Verständnis dieser Arbeit erforderlich ist. Für einen tieferen Einblick in die Metamodellierung wird der interessierte Leser auf [250] verwiesen. Grundlage für die Metamodellierung ist der Begriff des Modells,

welcher nach [225] die Abbildung eines Originals in Form einer künstlichen Entität ist, welche nur die Merkmale des Originals enthält, die für einen spezifischen Anwendungszweck notwendig sind. Da ein Modell auch künstliche Entitäten abbilden kann, ist es möglich, dass ein Modell wiederum Modelle beschreibt [250]. Ein Modell das ein anderes Modell beschreibt, wird als Metamodell bezeichnet [86, 214].

Grundlagen für die Metamodellierung sind in der objektorientierten Programmierung (siehe Abschnitt 3.1.2) zu finden. In dieser existieren mit der 'Klasse' und dem 'Objekt' bzw. 'Instanz' zwei elementare Arten von Entitäten [250]. Klassen beschreiben den Typ eines Objektes und legen die Grundstruktur von Objekten fest. Objekte werden durch Instantiierung aus Klassen erzeugt und belegen die in der Klasse spezifizierten Eigenschaften mit konkreten Werten. Im Kontext der Metamodellierung geht damit ein konkretes Modell durch Instantiierung aus einem Metamodell hervor.

Ein Standard für die Metamodellierung wird mit der 'Meta Object Facility (MOF)' [188] durch die 'Object Management Group' (OMG) spezifiziert. Dieser kann als der wichtigste Metamodellierungsstandard betrachtet werden [250]. So ist MOF die Grundlage der ebenfalls von der OMG spezifizierten UML [189]. Mit MOF wird eine Modellhierarchie – wie in Abbildung 5.4 dargestellt – spezifiziert. Die Instanzebene, welche die Objekte zur Abbildung der Realität enthält, wird als M_0-Ebene bezeichnet. Die Elemente der Instanzebene sind Instanzen des Modells, welches auf der M_1-Ebene angesiedelt ist. Dem Metamodell wird die Ebene M_2 und dem Meta-Metamodell die Ebene M_3 zugeordnet. Diese Hierarchiebildung könnte theoretisch immer weiter fortgesetzt werden. Nach der Spezifikation von MOF [188] muss eine Mindestanzahl von zwei Ebenen definiert sein, wobei theoretisch beliebig viele Ebenen unterstützt werden können. Der übliche Fall ist eine vierschichtige Architektur mit den Ebenen $M_0 - M_3$ (vgl. Abbildung 5.4). Auch UML liegt ein vierschichtiges Metamodell zugrunde [189]]. Ist dies der Fall muss die M_3-Ebene als eine selbstbeschreibende Ebene spezifiziert werden.

Das Konzept der Metamodellierung ist weit verbreitet und kann auch zur Modellierung von RDBMS angewendet werden. So lässt sich für ein RDBMS mit den Ebenen 'Systemtabelle', 'Tabelle' und 'Zeile' eine dreischichtige Metaebenenhierarchie definieren [188]. Genauso kann in der Biodiversitätsinformatik bei der Verwendung von ABCD, welcher als Standard in Form von XSD spezifiziert ist, das Tripel 'XSD', 'ABCD' und 'Datensatz' als eine Metaebenenhierarchie betrachtet werden.

In der strikten Metamodellierung nach [7] dürfen zwischen den einzelnen Ebenen einer Hierarchie nur Instanziierungsbeziehungen existieren. Für dieses strikte Paradigma konnten allerdings in [250] Nachteile identifiziert werden. Mit dem 'Open

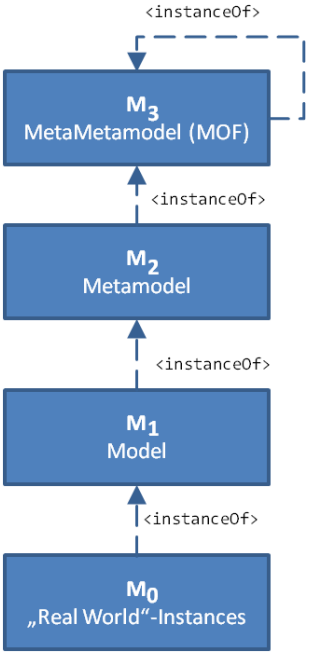

Abbildung 5.4: vierschichtige Meta-Ebenen-Hierarchie in der Meta Object Facility (MOF)

MetaModeling Environment' (OMME) wurde eine Metamodellierungsumgebung geschaffen, in welcher Metamodelle, die dem Paradigma der strikten Metamodellierung widersprechen, implementiert werden können. OMME unterstützt dabei auch Elemente der Metamodellierung wie 'Clabjects', 'Deep Instantantiation' und 'Powertypes' und bietet somit umfassende Möglichkeiten zur Erstellung von Metamodellen an [250].

Durch Metamodellierung in OMME werden Modellierungssprachen spezifiziert. Modellierungssprachen verfügen nach [32] über eine abstrakte Syntax, eine konkrete Syntax, eine Semantik, Abbildungen auf andere Sprachen und die Möglichkeit zur Erweiterbarkeit. Die Definition von abstrakter und konkreter Syntax, sowie der Semantik können in OMME spezifiziert werden [250]. Die Abbildung auf andere Sprachen wird in Abschnitt 5.6 mit dem Mapping von PODSL-Biodiv auf DwC vorgenommen. Die Erweiterbarkeit wird in PODSL in Abschnitt 5.3.6 vorgestellt.

5.3 Prinzipien von PODSL und PODSL-Biodiv

In Abschnitt 4.3 wurde eine Reihe von Kriterien formuliert, die ein Datenstandard für die Biodiversitätsinformatik erfüllen muss. In diesem Abschnitt werden Konzepte vorgestellt, welche die Erfüllung dieser Kriterien für PODSL und PODSL-Biodiv er-

möglichen. Die Metastruktur von PODSL wird in Abschnitt 5.4.1 ausführlich besprochen. Für den folgenden Abschnitt ist es aber relevant zu wissen, dass PODSL in eine dreischichtige Metahierarchie eingebettet ist und somit aus einem Metamodell, einem Modell und der Instanzebene besteht. Da in OMME allgemein für Modellelemente der Begriff 'Konzept' verwendet wird, wird dieser im folgenden für Modellelemente in PODSL verwendet.

5.3.1 Entitäten und Beziehungen

Basis für die Speicherung von Daten in PODSL ist die Strukturierung der Daten in Entitäten und Beziehungen. Dazu werden diese Konzepte auf der M_2-Ebene von PODSL eingeführt. Eine Entität enthält analog zur ER-Modellierung eine Menge von Attributen, welche den Datentyp spezifizieren und eine Entität genauer beschreiben können. Durch Beziehungen wird in PODSL spezifiziert, dass zwischen zwei Entitäten eines Beziehung besteht. Dazu können analog zur ER-Modellierung Kardinalitäten angegeben werden. Zusätzlich muss aber auch die Art der Beziehung klassifiziert werden können. Die Art einer Beziehung beschreibt die Qualität der Beziehung z.B. ob die Beziehung zwischen zwei biologischen Objekten die Art des Zusammenlebens beschreibt oder aber ob zwischen zwei biologischen Objekten eine Teil-Ganzes Beziehung besteht und wird in PODSL über Modalitäten realisiert, welche den Beziehungen hinzugefügt werden können.

Ausgangspunkt für die Modellierung ist das ER-Metamodell in OMME aus [250]. Dieses wird in folgender Weise angepasst[2]:

- Zuordnung der Beziehungen zu Dokumentenaspekten
- Berücksichtigung von Modalitäten in Beziehungen

Diese werden insbesondere zur Modellierung von Dokumenten zur Speicherung der Ausführung von Prozessen benötigt. Die Zuordnung einer Beziehung zu einem Dokumentenaspekt bezieht sich dabei auf die in Abschnitt 4.2.1 vorgestellten Dokumentenaspekte. Diese Zuordnung gibt an, welcher Dokumentenaspekt durch die Beziehung beschrieben wird. So sind z.B. Zeitangaben dem temporalen Dokumentenaspekt zugeordnet.

Die Angabe einer Modalität beschreibt die Art und Weise der Beziehung zwischen den Entitäten. In der klassischen ER-Modellierung wird über die Benennung der Beziehung die Funktion der Beziehung modelliert. Zusätzlich ist es für die Modellierung

[2]Durch dieses Anpassung wird das Konzept der Relationen der ER-Modellierung signifikant verändert. Deshalb wird in diesem Kontext in PODSL von Beziehungen gesprochen.

mit PODSL allerdings erforderlich, die Beziehung durch ein Attribut zu beschreiben. Diese Funktion erfüllen Modalitäten in PODSL. Eine Modalität entspricht damit einem speziellen Beziehungsattribut in der ER-Modellierung. Zusätzlich werden die aus ER-Modellierung bekannten Kardinalitäten in PODSL berücksichtigt.

Als Beispiel kann die Speicherung der Koexistenz von zwei biologischen Objekten dienen, wie z.B. die Koexistenz eines Pilzes mit einem Baum (siehe Abbildung 5.5)[3]. Sowohl der Pilz als auch der Baum werden in PODSL als Instanzen einer Entität modelliert. Zwischen diesen biologischen Objekten besteht eine Beziehung. Diese kann 'parasitischer' aber auch 'symbiotischer' Natur sein. In der Biodiversitätsforschung ist es wichtig, die Art dieser Beziehung zu erfassen. In PODSL wird die Beziehung zwischen dem Pilz und der Pflanze durch eine oder mehrere Modalitäten erfasst. Im Beispiel ist dies die Modalität 'FormOfCoexistenceModality'. In dieser wird es ermöglicht ein Attribut anzugeben, welche beschreibt, ob eine parasitische oder symbiotische Beziehung vorliegt. Dazu wird auf ein kontrolliertes Vokabular (siehe Abschnitt 5.3.8) zurückgegriffen.

5.3.2 Speicherung der Prozessausführung

Wie in Abschnitt 4.1 gezeigt wurde, ist die Strukturierung von Tätigkeiten in Form von Prozessen in vielen Anwendungsgebieten von entscheidender Bedeutung. Auch in der Biodiversitätsinformatik lassen sich Arbeitsabläufe mit Prozessen beschreiben. Damit aus der Ausführung dieser Prozesse Wissen entstehen kann, ist es notwendig, die Ergebnisse dieser Prozesse zu dokumentieren und mit anderen Wissenschaftlern zu teilen. Die Ausführung eines Prozesses ist stets mit einer Aussage verknüpft (4.2.1). Dabei wird für alle Prozessperspektiven, die im Prozessmodell abstrakt modelliert sind, mit konkreten Werten belegt, in dieser Aussage der konkrete Wert angegeben, welcher bei der tatsächlichen Ausführung des Prozesses verwendet wird. So wird in dem Prozess der Identifikation eines biologischen Objektes bei der Geländekartierung abstrakt vorgeschrieben, dass ein Sammler ein Fundobjekt an einem bestimmten Ort zu einem bestimmen Zeitpunkt identifiziert.

[3]Das Foto der Birke steht unter Creative Commons Attribution 3.0 Unported-Lizens und wurde von Harald Bischoff aufgenommen. Die Weitergabe und Vervielfältigung des Bildes ist nur durch Nennung seines Namens und unter dieser Lizenz möglich. Das Bild es Birkensporlings ist linzenzfrei.

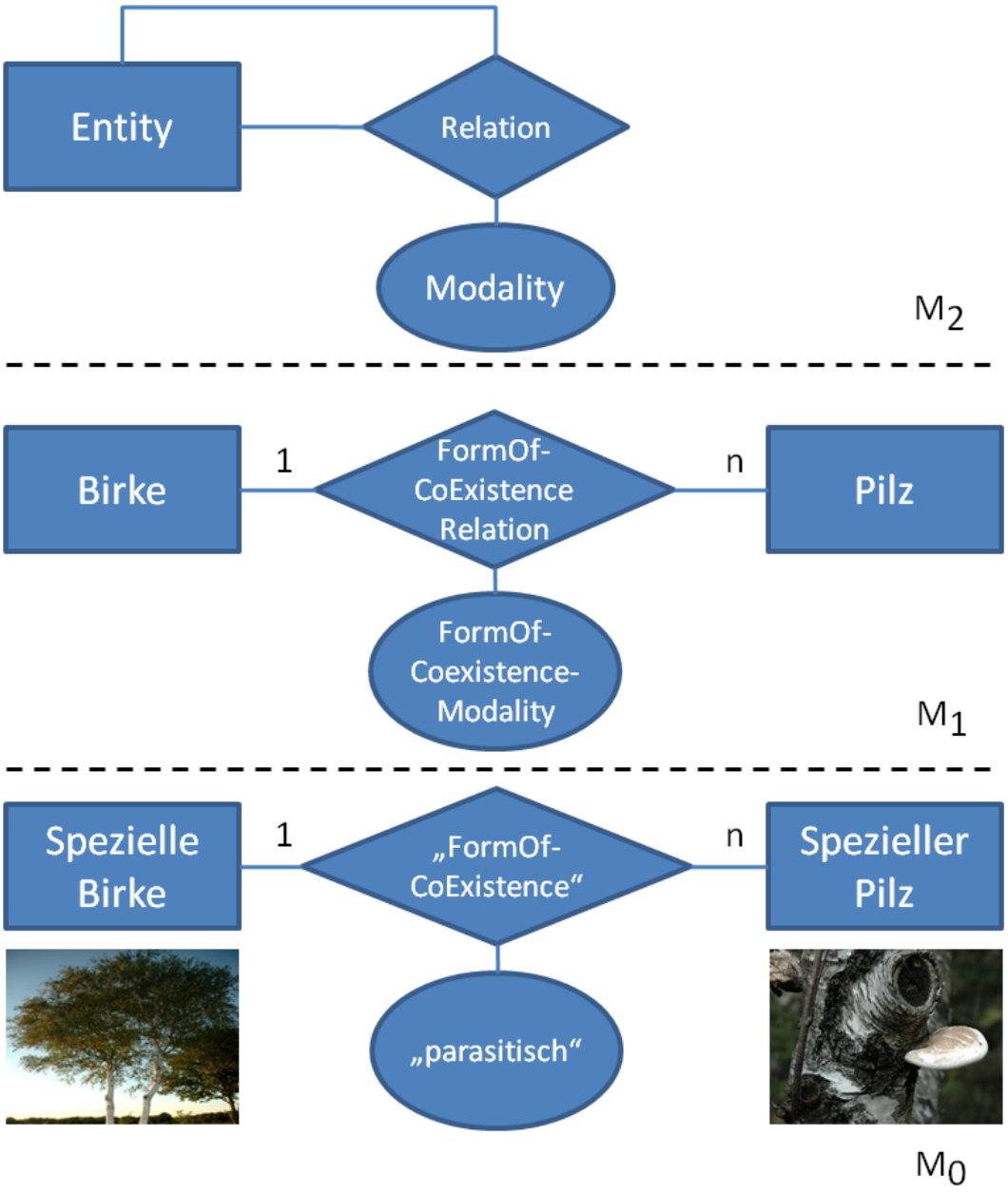

Abbildung 5.5: Modellierung einer Beziehung auf verschiedenen Metaebenen

Die konkrete Ausführung des Prozesses führt zu einer Aussage wie:

'Josef Simmel hat am 27.3.2012 um 14.12 Uhr ein bestimmtes biologisches Objekt als Quercur robur (Eiche) identifiziert, welches sich an den GPS Koordinaten 49.4628332, 11.3526638 befindet'.

Aussagen dieser Art werden durch die Zuordnung ihrer Elemente in Dokumentenaspekte in logische, thematisch unabhängige Bereiche untergliedert. Die Eigenschaften eines Prozesses werden in POPM über Perspektiven modelliert. Wenn die Ausführung eines in POPM modellierten Prozesses dokumentiert wird, muss ein entsprechendes Erfassungsdokument über Elemente verfügen, die es gestatten, den Zustand der Prozessperspektiven zum Ausführungszeitpunkt zu speichern. Die Perspektiven eines Prozesses werden damit auf bestimmte Dokumentenaspekte abgebildet.

Um eine Prozessausführung zu dokumentieren, wird in PODSL das 'ProcessExecutionDocument' (PED) als spezielle Entität auf der M_2-Ebene eingeführt (siehe Abbildung 5.6). Das PED wurde bereits in Kapitel 4 zur Erfassung der Ausführung von Prozessen eingeführt. Somit werden den Perspektiven eines POPM-Prozesses die Dokumentenaspekte in dem PED gegenübergestellt. In den Dokumentenaspekten wird die Prozessausführung über Beziehungen, die dem jeweiligen Dokumentenaspekt zugeordnet sind, mit Entitäten verknüpft. Damit ist das 'ProcessExecutionDocument' das zentrale Dokument zur Speicherung der Ausführung eines Prozesses und Grundlage für die Speicherung der Prozessausführung in der Metastruktur von PODSL.

Die Strukturierung nach Dokumentenaspekten erlaubt die systematische Gliederung eines Datenstandards. Die Verwendung von Dokumentenaspekten ermöglicht die Erfassung von Prozessen in PED´s (siehe Abschnitt 4.2). In den Dokumentenaspekten werden dem 'ProcessExecutionDocument' Entitäten zugeordnet, wie in Abbildung 5.6 demonstriert ist. Jedem PED kann ein bestimmtes Objekt zugewiesen werden, welches den Gegenstand der Prozesshandlung darstellt. Dieses ist der primäre Gegenstand der Prozesshandlung. In dem obigen Beispiel einer Aussage ist dies das biologische Objekt, das kartiert wird. Es ist zu beachten, dass der Prozess der Geländekartierung keine Aussage der Form

'Josef Simmel hat am 27.3.2012 um 14.12 Uhr eine Quercur robur (Eiche) an den GPS Koordinaten 49.4628332, 11.3526638 kartiert'.

treffen darf, da 'Quercus robur' eine taxonomische Bezeichnung ist und damit das Ergebnis der Handlung 'Objekt identifizieren' darstellt. Ein anderer Sammler könnte bei der Identifikation im Bezug auf dasselbe biologische Objekt durchaus zu einem

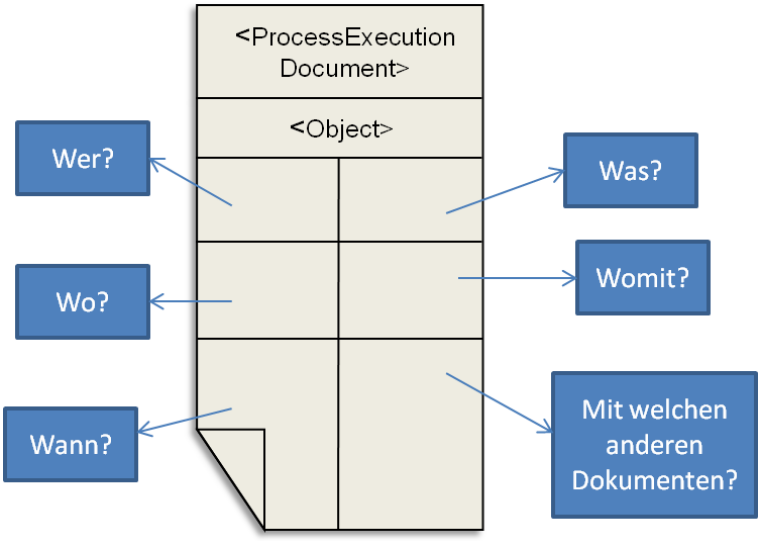

Abbildung 5.6: Unterteilung eines 'ProcessExecutionDocuments' nach Dokumentenaspekten

anderen Ergebnis kommen, da eine Identifikation fehlerbehaftet sein kann. Im Metamodell von PODSL wird dieser Umstand dahingehend berücksichtigt, dass in jedes 'ProcessExecutionDocument' über eine bestimmte Beziehung verfügen muss, welche das 'ProcessExecutionDocument' mit dem Objekt des Prozesses verknüpft. In Abbildung 5.6 ist dies dadurch symbolisiert, dass für die Speicherung der Objektrelation mit <Object> ein separater Bereich im 'ProcessExecutionDocument' existiert.

Für die Dokumentenaspekte wurden in Abbildung 5.6 separate Bereiche gezeichnet, welche Speicherplatz für die Antworten auf die Fragen 'Wer?', 'Was?', 'Wo?', 'Womit?', 'Wann?' und 'Mit welchen anderen Dokumenten?' symbolisieren. Im funktionalen Dokumentenaspekt (Was?) wird gespeichert, welche Prozesshandlung ausgeführt wird und das Ergebnis dieser Handlung. So wird im funktionalen Dokumentenaspekt das Objekt der Prozessausführung mit dem Prozessergebnis verknüpft. Der funktionale Dokumentenaspekt bildet damit den Kern des 'ProcessExecutionDocuments'. In der obigen Beispielaussage ist dies das Identifizieren des biologischen Objektes als 'Quercus robur'. Das 'Identifizieren' stellt die Primärhandlung des Prozesses dar und ist in einem korrelierendem Prozessmodell Teil der funktionalen Prozessperspektive. Im funktionalen Dokumentenaspekt muss dementsprechend in Speicherplatz für diese Daten zur Verfügung stehen.

Die anderen Dokumentenaspekte dienen dazu die Prozessausführung näher zu beschreiben. Im organisatorischen Dokumentenaspekt werden z.B. eine Beziehung zu dem Verantwortlichen (Wer?) der Prozessausführung gespeichert, wohingegen im lo-

kalen und temporalen Dokumentenaspekt der Ausführungsort und die Ausführungszeit des Prozesses gespeichert werden. Die Dokumentenaspekte korrelieren dabei zu den Prozessperspektiven aus POPM. Dabei kann analog zur Prozessmodellierung ein Dokumentenaspekt mehrfach verwendet werden. Dies ist erforderlich, wenn in einem korrelierendem Prozessmodell eine Prozessperspektive mehrfach genutzt wird. Dies tritt beispielsweise dann auf, wenn in einem Prozess die operationale Prozessperspektive bei einer Messung mehrere Messinstrumente benötigt werden (Messung der Position mit Karte und Kompass).

Zur Modellierung der Dokumentenaspekte werden in PODSL Beziehungen verwendet, die den Dokumentenaspekten zugeordnet sind. So wird das 'ProcessExecutionDocument' mit den Entitäten verknüpft, die bei der Prozessausführung den jeweiligen Dokumentenaspekt beschreiben. So wird für die Speicherung der Beispielaussage im organisatorischen Dokumentenaspekt eine Beziehung zu dem Sammler 'Josef Simmel' gespeichert und nicht der Wert 'Josef Simmel' an sich. Die Verwendung von Beziehungen erlaubt deshalb das referenzieren des Sammlers. Dies hat Vorteile gegenüber der direkten Speicherung des Wertes des Sammlers als Text, da über die Referenz auf eine Entität Sammler der vollständige Datensatz des Sammlers zugänglich gemacht wird. Darüber hinaus ist über die Referenz der Sammler eindeutig identifiziert und es entstehen keine fehlerhaften Daten bei Namensgleichheit und durch Veränderungen im Datensatz des Sammlers.

Auf der M_1-Ebene von PODSL werden Instanzen des 'ProcessExecutionDocument' für alle relevanten Prozesse der Anwendungsdomäne spezifiziert. Jedem Prozess einer Anwendungsdomäne ist damit ein spezielle 'ProcessExecutionDocument' zugeordnet. Die 'ProcessExecutionDocuments' bieten damit die Möglichkeit, die Daten, die bei der Prozessausführung entstehen, dauerhaft zu speichern. Dazu werden für alle Dokumentenaspekten Beziehungen zu Entitäten hergestellt, welche diesen Dokumentenaspekt beschreiben. Die Beziehung in PODSL erfüllt dabei die Funktion einer Referenz auf eine bestimmte Entität.

Dies soll am Beispiel des Prozesses der Identifikation eines biologischen Objektes demonstriert werden, wie in Abbildung 5.7 dargestellt ist. In diesem Prozess wird einem biologischen Objekt eine taxonomische Identifikation zugeordnet. Dafür ist insbesondere wichtig, wer diese Identifikation durchgeführt hat. Dies ist im Prozessmodell in der organisatorischen Perspektive modelliert. In der operationalen Perspektive sind dem Prozess Hilfsmittel der Identifikation zugeordnet. Dies können z.B. Bestimmungshandbücher oder wissenschaftliche Veröffentlichungen sein.

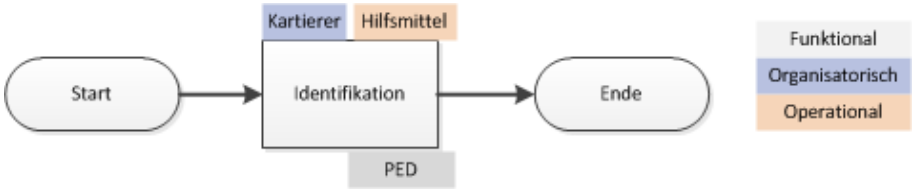

Abbildung 5.7: Prozessmodell zur Identifikation eines biologischen Objektes

Wird dieser Prozess ausgeführt, müssen die Daten die bei der Prozessausführung entstehen, in einem Dokument gespeichert werden. Dazu wird auf der M_1-Ebene von PODSL-Biodiv mit der 'Identification' ein spezielles 'ProcessExecutionDocument' spezifiziert, um genau diese Prozessausführung zu erfassen. Im Zentrum der Prozessausführung steht dabei ein biologisches Objekt, das identifiziert wird. Dies ist das Objekt des 'ProcessExecutionDocuments'. In der funktionalen Perspektive des Prozesses ist die Identifikation dieses Objektes modelliert. Im Dokument wird dies durch eine Beziehung zwischen dem 'ProcessExecutionDocument' und der Entität 'Taxon' im funktionalen Dokumentenaspekt modelliert. Die Entität Taxon ist dabei ein Schema mit Attributen, welche die Bestimmung von biologischen Objekten ermöglichen. Da im Prozessmodell in der organisatorischen Perspektive als Verantwortlicher der Kartierer und in der operationalen Perspektive Hilfsmittel zur Bestimmung modelliert sind, muss das 'ProcessExecutionDocument' über Dokumentenaspekte verfügen, welche diese Daten bei der Prozessausführung erfassen können. Dazu enthält das 'ProcessExecutionDocument' eine Beziehung auf die Entität 'Person', welche dem organisatorischen Dokumentenaspekt zugeordnet ist und analog im organisatorischen Dokumentenaspekt eine Beziehung auf das Hilfsmittel der Bestimmung. Zusätzlich werden als Merkmale der Prozessausführung der Zeitpunkt der Prozessausführung und der Ort gespeichert. Die vorläufige Modellierung des 'ProcessExecutionDocument' für die Identifikation ist in Abbildung 5.8 dargestellt.

Ein wesentliches Element bei der Identifikation von biologischen Objekten ist die Sicherheit der Bestimmung. Diese ist eine Eigenschaft der Identifikationshandlung und somit dem funktionalen Dokumentenaspekt des 'ProcessExecutionDocuments' zugeordnet. Um diesen zusätzlichen Faktor zu berücksichtigten, wird dem funktionalen Dokumentenaspekt 'Identifikation' eine Modalität hinzugefügt, welche für die Speicherung der Sicherheit der Bestimmung verantwortlich ist[4]. Dies ist eine Eigenschaft der Beziehung im funktionalen Dokumentenaspekt und wird in PODSL der Modellierung dieses Aspekts zugeordnet, wie in Abbildung 5.9 dargestellt ist.

[4]Dazu wird in PODSL-Biodiv ein kontrolliertes Vokabular verwendet. Siehe Abschnitt 5.3.8

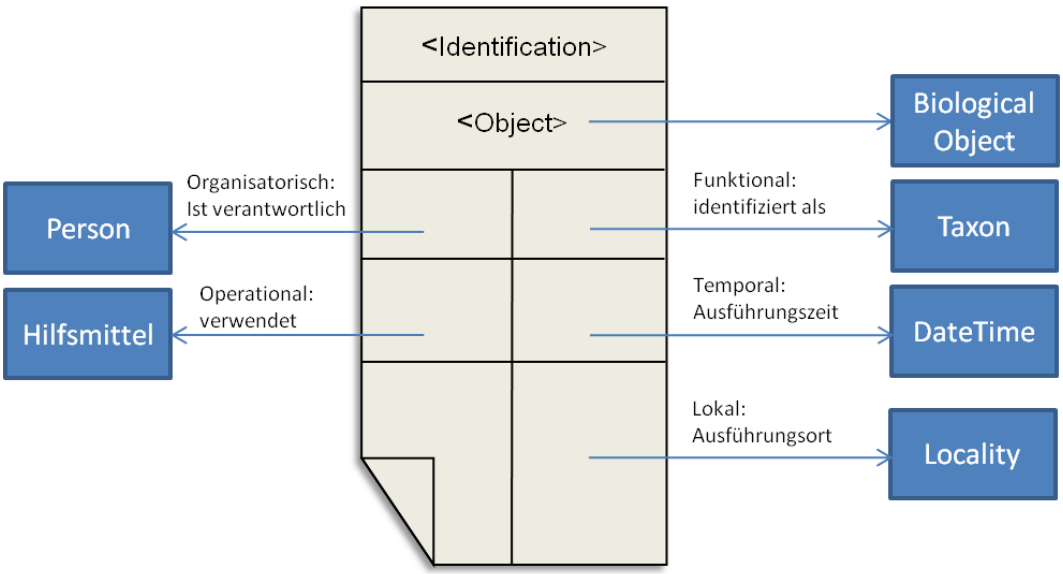

Abbildung 5.8: Dokumentenaspekte im 'ProcessExecutionDocument' für die Identifikation eines biologischen Objektes

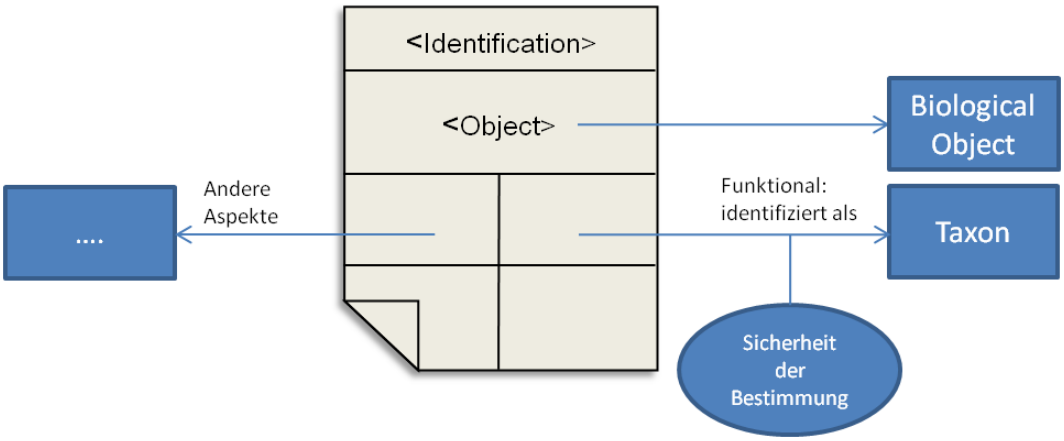

Abbildung 5.9: Modalität im funktionalen Dokumentenaspekt der Identifikation

Durch das Konzept der Modalitäten lässt sich damit die Art und Weise der Prozessausführung genau erfassen. Die Gliederung eines Datenstandards in Dokumentenaspekte fokussiert Anforderungen in Bezug auf die Kriterien der Vollständigkeit und der Redundanzfreiheit. Durch die strukturierte Gliederung nach Aspekten ist gut zu erkennen, ob eine Anforderung nicht, nicht ausreichend oder mehrfach in einem Schema repräsentiert ist.

5.3.3 Unterstützung von kompositen Prozessen

In der Prozessmodellierung ist es eine gängige Methode, komplexe Prozesse in mehrere Subprozesse zu unterteilen. Hierzu werden in POPM komposite Prozesse verwendet. Dies ist auch bei der Modellierung eines 'ProcessExecutionDocuments' relevant, da in PODSL auch komposite Prozesse erfasst werden müssen. Dabei spielen Sequenzen von Prozessen eine Rolle, bei denen die Reihenfolge der Prozessausführung relevant ist. In der Prozessmodellierung wird dies über die verhaltensorientierte Perspektive modelliert. Dementsprechend muss die Metastruktur eines 'ProcessExecutionDocuments' die Erfassung von Subprozessen und die Reihenfolge der Prozessausführung erfassen können. Dies wird im verhaltensorientierten Dokumentenaspekt eines 'ProcessExecutionDocuments' abgebildet.

Die Subprozesse eines Prozesses werden im Datenmodell über separate 'ProcessExecutionDocuments' erfasst. Diese werden im verhaltensorientierten Dokumentenaspekt mit dem 'ProcessExecutionDocument' des übergeordneten Prozesses verknüpft. Eine Besonderheit des verhaltensorientierten Dokumentenaspektes ist, dass Sequenzen nicht direkt über eine 1:n Beziehung modelliert werden können, da über diese die Reihenfolge der Prozessausführung nicht dargestellt werden kann. Im Metamodell von PODSL wird dieser Umstand über eine spezielle Struktur, der 'ProcessExecutionSequence' berücksichtigt. In dieser werden Dokumente zur Prozessausführung in der Reihenfolge ihrer Ausführung gespeichert.

Das Prozessmodell eines beliebigen, abstrakten Prozesses ist in Abbildung 5.10 dargestellt. In dieser Abbildung wird deutlich, dass die Reihenfolge der Prozessausführung von Bedeutung ist und diese auch in einem 'ProcessExecutionDocument' zur Modellierung des verhaltensorientierten Dokumentenaspektes benötigt wird.

Eine entsprechende grafische Darstellung der Modellierung des verhaltensorientierten Dokumentenaspektes ist in Abbildung 5.11 dargestellt. Die 'ProcessExecutionSequence' enthält die Dokumente aller Subprozesse in geordneter Reihenfolge. Da diese wiederum selbst 'ProcessExecutionDocuments' sind, lassen sich Prozesshierarchien beliebiger Tiefe in PODSL darstellen. Wie bei der Prozessmodellierung ist es

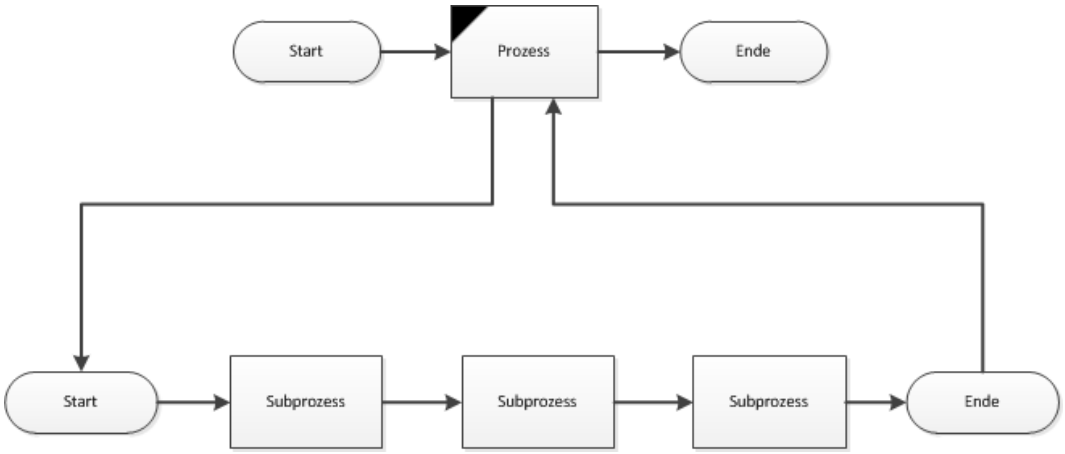

Abbildung 5.10: Kompositer Prozess mit Subprozessen

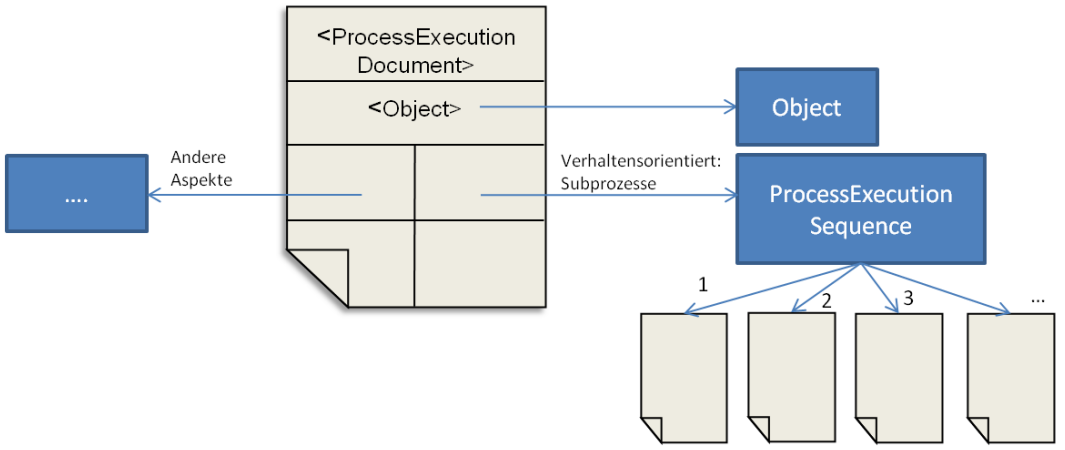

Abbildung 5.11: 'ProcessExecutionSequence' zur Erfassung des verhaltensorientierten Dokumentenaspektes

dabei eine Designentscheidung, ob diese Möglichkeit vollständig ausgenutzt werden muss. Sind in den Subprozessen alle Werte bis auf die Werte des funktionalen Dokumentenaspekt gleich, bietet es sich an, dies über ein 'ProcessExecutionDocument' zu modellieren, in welchem der funktionale Dokumentenaspekt mehrfach verwendet wird. In der Prozessmodellierung mit POPM werden zur Vereinfachung ähnliche Modellierungselemente wie z.B. der evidenzbasierte Entscheider verwendet [68].

5.3.4 Referenzen

In einem Datenspeicher muss jedes zu speichernde Element eindeutig gekennzeichnet sein. Im Rahmen der RDBMS geschieht dies über die Vergabe von Primärschlüsseln. Im Bereich der Ontologien werden Entitäten über URN eindeutig gekennzeichnet.

Insbesondere bei der Übertragung von Daten in Netzwerken ist es von Vorteil, eine Entität über eine URN zu identifizieren, da diese in allen Teilen des Netzwerks eindeutig ist. Bei der Kennzeichnung über Primärschlüssel in einem RDBMS hingegen ist die Identifikation im Allgemeinen nur lokal für diese spezielle Datenbank garantiert.

Das Konzept der Referenzierung über URN´s hat in der Biodiversitätsinformatik über die sogenannten 'Life Science Identifier (LSID)' Einzug erhalten. LSIDs wurden von der OMG als Standard spezifiziert [184]. Die öffentliche Version findet sich unter [185]. Die Struktur einer LSID ist nach [186] durch die Form

'URN:LSID:Authority:Namespace:Object:[Revision-ID]'

spezifiziert. Dabei sind die Präfixe 'URN' und 'LSID' obligatorisch. Unter 'Authority' ist die Webadresse der einspeisenden Organisation zu verstehen. Diese muss über 'Namespace' und 'Object' die Eindeutigkeit der Referenz garantieren und kann über 'Revision-ID' optional eine Versionsnummer verwalten.

Über die Verwendung von LSID in der Biodiversitätsinformatik herrscht Uneinigkeit [233]. Für die Verwendung von LSID ist von der TDWG mit [233] eine Guideline als Standard publiziert worden. In [233] wurde an LSID bemängelt, dass eine LSID nicht direkt über einen Browser aufgelöst werden können, da das Format der LSID nicht dem HTTP-Protokoll entspricht. So kann beispielsweise eine LSID wie

'urn:lsid:zoobank.org:act:8BDC0735-FEA4-4298-83FA-D04F67C3FBEC'

nicht direkt aufgelöst werden, sondern muss zunächst in

'http://zoobank.org/urn:lsid:zoobank.org:act:8BDC0735-FEA4-4298-83FA-D04F67C3FBEC'

umgeschrieben werden. Dementsprechend wird in [233] auch eine Guideline für die Verwendung von 'Global Unique Identifiern' (GUID) publiziert. Eine Präferenz für die Verwendung von LSIDs oder GUIDs wird in [233] nicht publiziert.

Nach der LSID Spezifikation [185] ist die Auflösung einer Referenz über Webservices (sie Kapitel 6) vorgesehen. Ein alternativer Ansatz zur eindeutigen Referenzierung wird mit EZID in [163] beschrieben.

Für die Verwendung in einem Datenstandard in einer Infrastruktur ist der Name 'Life Science Identifier' irreführend, da auch Entitäten wie Orte, Institute, Multimediaobjekt o.ä. referenziert werden müssen, die nicht direkt ihren Ursprung in den Lebenswissenschaften haben. Die Unterteilung eines Identifiers in eine Kennung für das Repositorium und eine Kennung für ein konkretes Objekt eines Repositorium ist

aber bei der Verwendung in Netzwerken wie BDEI (siehe Kapitel 8) vorteilhaft, da über den Identifier auf das Repositorium geschlossen werden kann.

Beim Setzen von Referenzen muss zusätzlich beachtet werden, was genau referenziert werden soll. Dazu muss jedes Element des Schemas von PODSL und PODSL-Biodiv auf der M_2 und M_1-Ebene eindeutig referenziert sein. Diese Referenz dient dazu, ein Element einer Modellierungssprache eindeutig zu kennzeichnen. Dazu werden Konzepte in OMME über Identifier der Form

'model:/repository/Modell/Konzeptname'

referenziert. So wird z.B. durch

'model:/www.ai4.uni-bayreuth.de/PodslM1Core/Person'

das Konzept Person eindeutig referenziert. Dem Identifier ist zu entnehmen, dass das Konzept Person auf der M_1-Ebene[5] in PODSL spezifiziert ist. Die Verwendung dieser Referenz für ein Konzept 'Person' zeigt an, dass unter 'Person' genau das an der Referenz spezifizierte Konzept zu verstehen ist. Da die Attribute von Person ebenfalls Modellelemente von PODSL in OMME sind, sind diese über eine eigene Referenz auflösbar. Auf dieses Weise ist über OMME jedes Modellelement eindeutig referenziert.

Andererseits müssen auch Datensätze auf M_0-Ebene eindeutig referenziert werden. Für diese Datensätze bietet sich eine Referenz in Form einer LSID an, da für die Auflösung der Referenz das Repository und eine Identifier im Repository benötigt werden. Da die Verwendung von LSID´s allerdings umstritten ist, soll für PODSL auf einen Identifier der Form

'repository/Object'

zurückgegriffen werden. Dieser wird im Folgenden als PODSL-Identifier bezeichnet. Dabei wird ein Objekt in einem Repositorium über eine GUID eindeutig gekennzeichnet. Der Unterschied zwischen den Referenzen wird in Abbildung 5.12 veranschaulicht. Im oberen Teil ist eine Referenz auf das Konzept 'Person' gesetzt. Dadurch wird in PODSL das Schema der 'Person' eindeutig referenziert. Wird der Identifier aufgelöst, wird das Schema des Konzepts 'Person', welches über die Attribute 'firstName', 'lastName' und 'dateOfBirth' verfügt, zurückgegeben. Diese Attribute werden in PODSL wiederum mit einem Identifier versehen, so dass diese eindeutig referenziert sind. Die Attribute selbst sind Elemente von OMME und können über den Identifier in OMME aufgelöst werden.

[5]Genau genommen wird angezeigt, dass Person im Kernmodul von M_1 liegt. Siehe dazu Abschnitt 5.4.3.

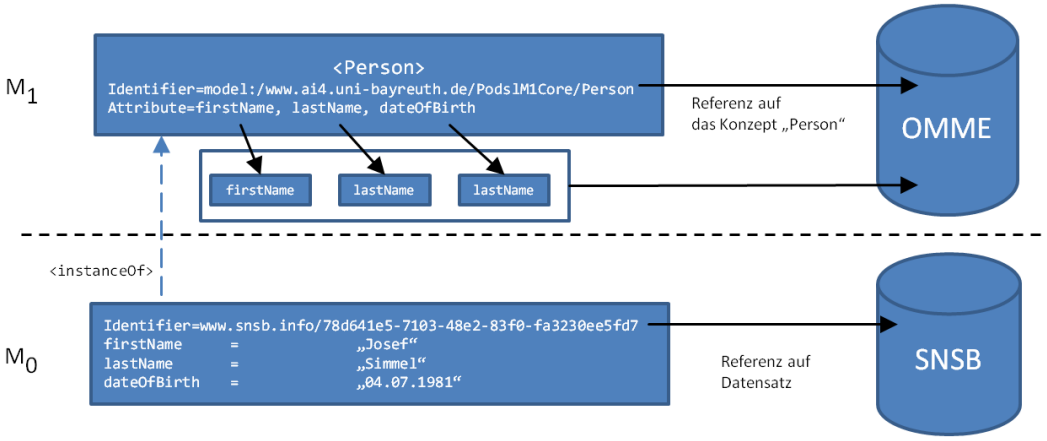

Abbildung 5.12: Referenzen auf verschiedenen Ebenen

Im unteren Abschnitt ist die Referenz einer Instanz einer Person in einem Repository abgebildet. Eine Auflösung der Referenz gibt eine konkrete Instanz einer Person zurück.

5.3.5 Referenzen auf externe Dokumente

Ein Datenstandard kann eine so komplexe Domäne wie die der Biodiversitätsinformatik nicht vollständig erfassen. So kann es erforderlich sein, dass innerhalb einer Domäne auch Dokumente aus benachbarten Domänen gespeichert werden müssen. Dies ist in der Biodiversitätsforschung bei der Erstellung von genetischen Analysen der Fall. Die Genetik ist eine separate Domäne, die über eigene Standards verfügt und auch nicht in die Domäne der Biodiversitätsforschung integriert werden kann. Dementsprechend muss es PODSL-Biodiv ermöglichen, Referenzen auf externe Dokumente zu setzen, welche nicht in PODSL modelliert sind.

Dokumente, welche nicht das Metaschema von PODSL implementieren, werden als externe Dokumente bezeichnet. Diese werden innerhalb eine Infrastruktur wie BDEI (siehe Kapitel 8) in Form von Dateien gespeichert. Mit der Referenz auf ein externes Dokument wird damit eine Datei referenziert, für welche der Speicherort in der Infrastruktur aufgelöst werden kann. Dies ist insbesondere relevant bei der Speicherung von Multimediaobjekten wie Bilder, Videos und Tonaufnahmen. Diese liegen regelmäßig innerhalb einer Infrastruktur in Form einer Datei vor[6]. Um ein Multimediaobjekt über eine Infrastruktur zugänglich zu machen, muss dieses in PODSL-Biodiv referenziert werden und für die Teilnehmer der Infrastruktur aufge-

[6]RDBMS ermöglichen die Integration von Multimediadaten in Form von 'Binary Large Objects' (BLOB) in Tabellen. Die Verwendung dieser ist aber ungenerisch und in Infrastrukturen ungünstig

löst werden. Dementsprechend werden für externe Dokumente in PODSL Konzepte aufgenommen, welche als Begleitdokumente bezeichnet werden. Diese beschreiben die externen Dokumente über Attribute wie Name, Autor oder Institut und geben zusätzlich den Speicherort an.

5.3.6 Vererbung

Ein zentrales Konzept zur vollständigen Erfassung einer Domäne und zur Vermeidung von Redundanzen in PODSL ist das Konzept der Vererbung (siehe Abschnitt 3.1.2). In diesem Konzept wird aus einer Klasse eine weitere Klasse abgeleitet, welche über alle Eigenschaften der Basisklasse verfügt und zusätzlich weitere Eigenschaften aufweisen kann. Bei der Erstellung der Schemata von Dokumenten und Entitäten in PODSL kann dies ausgenutzt werden, um unterschiedliche Anforderungen an die Genauigkeit (siehe Abschnitt 4.2.2) abzubilden und Redundanzen zu vermeiden.

Ausgangspunkt für die Vererbung in PODSL sind Basiskonzepte auf der M_1-Ebene für die auf der M_2-Ebene spezifizierten Modellelemente[7]. So ist auf der M_2-Ebene von PODSL das Konzept 'EntityType' spezifiziert. Auf der M_1-Ebene wird in PODSL 'BaseEntity' als eine spezielle Entität instanziiert. In 'BaseEntity' werde grundlegende Eigenschaften festgelegt, welche alle Entitäten in PODSL aufweisen müssen. Von 'BaseEntity' werden alle anderen Entitäten der M_1-Ebene abgeleitet, wie in Abbildung 5.13 dargestellt ist.

Auf diese Weise lässt sich eine Hierarchie von Entitäten ableiten, die immer spezielleren Anforderungen genügen. Analog zu den Entitäten werden für Beziehungen und die Prozessausführungen mit 'BaseRelation' und 'BaseProcessExecutionDocument' Basiskonzepte gebildet, von welchen für genauere Anforderungen speziellere Konzepte abgeleitet werden. Durch die Ableitung neuer Konzepte aus diesen Basiskonzepten können Sachverhalte immer exakter dargestellt werden. So werden in Abbildung 5.13 aus der 'organizational BaseEntity' die Klassen 'Institut' und 'Person' abgeleitet.

Auf diese Weise können zur Modellierung von Genauigkeitsanforderungen Ableitungshierarchien gebildet werden. Da Attribute in OMME als Modellelemente eindeutig referenziert werden, kann in OMME Mehrfachvererbung unterstützt werden. Dadurch erbt ein Konzept alle Attribute und Beziehungen seiner Oberkonzepte.

Abbildung 5.14 zeigt diese Modellierung ausgehend von der 'Local BaseEntity'. Dabei werden die Konzepte 'Locality Description' und 'Coordinates' von der 'Local-

[7] Analog zur Verwendung der Klasse 'Thing' in OWL bei Ontologien, von der auch alle anderen Klassen ableiten [255].

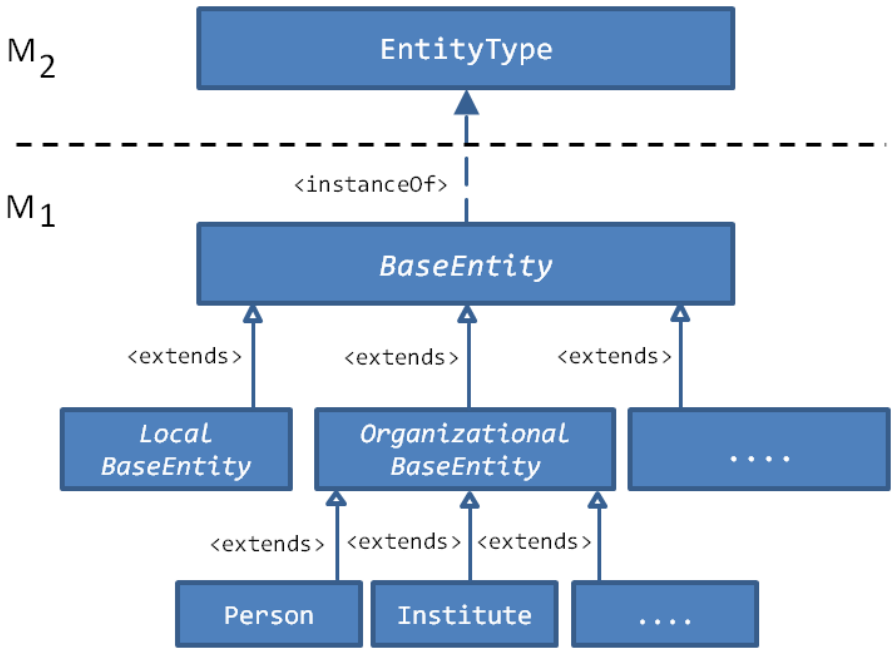

Abbildung 5.13: Vererbung ausgehend von der Basisklasse 'BaseEntity'

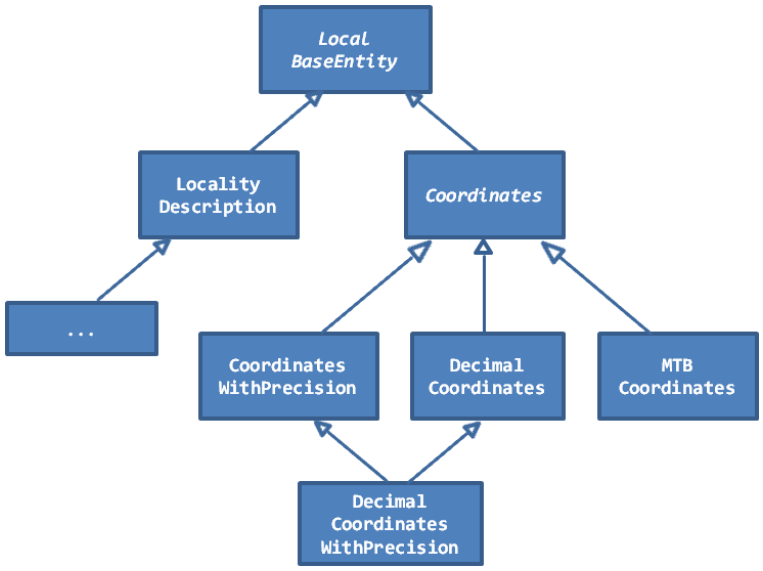

Abbildung 5.14: Ableitung für Ortsbeschreibungen

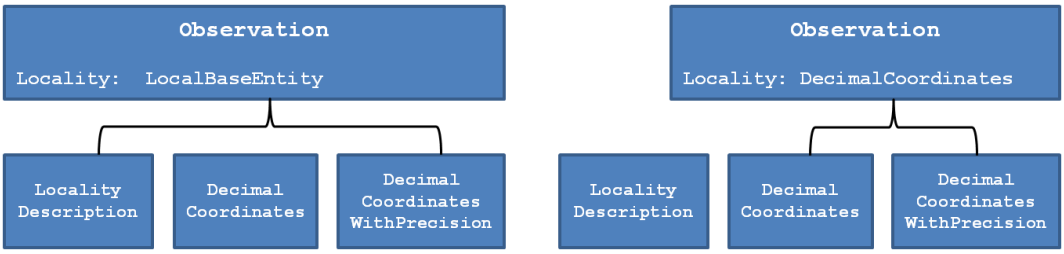

Abbildung 5.15: Prinzip der Ersetzbarkeit für die Dokumentation einer 'Observation' im lokalen Dokumentenaspekt

BaseEntity' abgeleitet. Dies zeigt an, dass diese Konzepte zur Beschreibung von Orten geeignet sind und geben für die Angabe des Ortes jeweils ein bestimmtes Format an. Dabei kann über die 'Locality Description' ein Ort über einen Text beschrieben werden, wohingehend 'Coordinates' die Angabe von Koordinaten in dezimale Form ermöglicht. Da Koordinaten in verschiedener Form angegeben werden können, ist eine weitere Spezialisierung erforderlich. Mit 'MTB'[8] und 'DecimalCoordinates'[9] werden zwei verschiedene Formate für die Angabe von Koordinaten spezifiziert. Zusätzlich wird mit 'CoordinatesWithPrecision' ein Konzept abgeleitet, welches es ermöglicht, die Qualität der Koordinatenangabe aufzunehmen.

Das Konzept 'DecimalCoordinatesWithPrecision' erbt von 'DecimalCoordinates' und 'CoordinatesWithPrecision', so dass dieses Konzept Attribute zur dezimalen Koordinatenangabe z.B. in WGS84 und zur Angabe der Präzision der Erfassung der Koordinaten enthält. Die Ableitung von den Basiskonzepten erfüllt dabei die Funktion eines 'Interfaces' aus der objektorientierten Programmierung.

Neue Konzepte können in PODSL ausschließlich durch Ableitung von einem bereits bekannten Konzept gebildet werden. Dies hat den Vorteil, dass das Prinzip der Ersetzbarkeit (siehe Abschnitt 3.1.2) aus der objektorientierten Programmierung eingesetzt werden kann, welches besagt, dass ein Objekt einer abgeleiteten Klasse immer ein Objekt der Basisklasse ersetzen kann [134].

Dies kann bei der Modellierung von Dokumenten zur Erfassung von Prozessen, wie in Abbildung 5.15 demonstriert wird, ausgenutzt werden. Die Abbildung zeigt die Modellierung des lokalen Aspekts in einem Prozessdokument zur Erfassung des Prozesses 'Observation' in zwei Varianten. In der ersten Variante (links) wird für das Attribut 'Locality' eine beliebiges Instanz eines Konzeptes gefordert, welches von der 'Local BaseEntity' abgeleitet ist. Dementsprechend sind Objekte der Konzepte

[8] Messtischblätter in Form von topographischen Karten und Bestimmung der Position durch den Rechts-Hoch-Wert.

[9] Über dieses Format werden GPS-Daten im WGS84-Format in PODSL erfasst.

'Locality Description', 'DecimalCoordinates' und 'DecimalCoordinatesWithPrecision' aufgrund des Prinzips der Ersetzbarkeit dazu geeignet.

In der zweiten Variante (rechts) wird hingegen für das Attribut 'Locality' eine höhere Genauigkeit gefordert. Es sind ausschließlich Instanzen von Konzepten, die von 'DecimalCoordinates' abgeleitet werden, dazu in der Lage, die Genauigkeitsanforderung zu erfüllen. Instanzen von 'Locality Description' erfüllen dies nicht (siehe Abbildung 5.14). Dies ist hingegen für Instanzen von der Klasse 'DecimalCoordinates' und 'DecimalCoordinatesWithPrecision' der Fall.

Somit können bei der Modellierung von Dokumenten zur Speicherung der Prozessausführung verschiedenen Genauigkeitsstufen angegeben werden. Falls für die Erfassung eines Dokumentenaspektes kein genaues Format spezifiziert werden soll, kann dieses durch die Verwendung eines Basiskonzepts generisch modelliert werden. Dies trägt wesentlich zur Flexibilität und Erweiterbarkeit von PODSL bei. Darüber hinaus ist das Konzept der Vererbung beim Einsatz von PODSL in Infrastrukturen von Vorteil. Da die Referenzierbarkeit von Daten auf M_0-Ebene gewährleistet ist und alle Entitäten von der 'BaseEntity' abgeleitet werden, muss ein Datenspeicher nicht alle Attribute einer Entität bei der Datenübertragung in einer Infrastruktur erfassen können. Da die Referenz auf das Original über den Identifier stets aufgelöst werden kann, können die Daten von Attributen, die nicht erfasst wurden, stets über die Infrastruktur beschafft werden (siehe Kapitel 8.4.5).

5.3.7 Unterstützung von Data Provenance

Die Unterstützung von Data Provenance hat das Ziel, Strukturen zur Verwaltung von Herkunft, Versionen und Veränderungen an Datensätzen zu erfassen (siehe Abschnitt 3.1.5). Die Herkunft eines Datensatzes wird in PODSL-Biodiv bereits durch die Verwendung von Identifiern ermöglicht, welche Auskunft über das Repository eines Datensatzes geben. Allerdings ist es erforderlich zu dokumentieren, welche Datensätze in andere Datenspeicher übertragen wurden und ob diese zur Anpassung an das lokale Datenmodell modifiziert wurden. Dies spielt insbesondere bei der materialisierten Datenintegration (siehe Abschnitt 6.1) eine Rolle.

Um dies zu realisieren, wird in PODSL mit der 'ProvenanceTable' eine Konzept zur Speicherung von Provenance-Informationen in Form einer speziellen Entität geschaffen, welche auf M_1-Ebene des Metamodells angesiedelt ist. Dieses enthält Referenzen auf alle Versionen des Datensatzes in einem Netzwerk. Die 'ProvenanceTable' ist ein reines Verwaltungsdokument und enthält außer für die Dokumentation der Data Provenance keine Datenstrukturen. Bei der Anwendung in Infrastrukturen werden

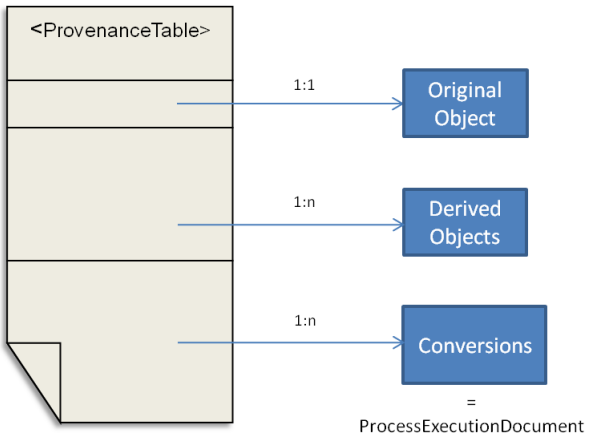

Abbildung 5.16: Schema der 'ProvenanceTable'

die Data-Provenance Dokumente kontinuierlich abgeglichen (siehe Kapitel 8).

Ein wesentliches Element von Data Provenance ist die Erfassung von Konvertierungen von Datensätzen. Dabei ist zu beachten, dass die Konvertierung eines Datensatzes selbst ein Prozess ist, welcher aus mehreren Teilprozessen bestehen kann. Somit kann die Konvertierung eines Datensatzes über spezielle 'ProcessExecutionDocuments' in PODSL erfasst werden. Diese enthalten als Objekt den Originaldatensatz. Im funktionalen Dokumentenaspekt wird dabei eine Beziehung auf den veränderten Datensatz als Output der Konvertierung angelegt. Verantwortliche und Art und Weise der Konvertierung können dabei über den organisatorischen Dokumentenaspekt und die anderen Dokumentenaspekte erfasst werden. Der Output der Konvertierung muss zusätzlich in der 'ProvenanceTable' als neue Version hinterlegt werden.

Die Struktur der 'ProvenanceTable' lässt sich damit in drei Teile untergliedern. Der erste Teil besteht in einer 1:1 Beziehung auf den Originaldatensatz. Der zweite Teil enthält eine 1:n Beziehung auf alle abgeleiteten Versionen des Originaldatensatzes. Im dritten Teil werden die Prozesse der Konvertierungen erfasst. Diese wird über eine 1:n Beziehung auf die 'ProcessExecutiondocuments' der Prozesse der Datenkonvertierung modelliert. Das Schema der 'ProvenanceTable' ist in Abbildung 5.16 dargestellt. Der Originaldatensatz kann dabei ein beliebiger Datensatz in PODSL sein, also eine beliebige Entität bzw. ein beliebiges 'ProcessExecutionDocument'. Dies gilt auch für abgeleitete Datensätze. Da Konvertierungen Prozesse sind, werden diese durch 'ProcessExecutionDocuments' erfasst. Damit besteht der Bereich 'Conversions' in Abbildung 5.16 aus Referenzen auf 'ProcessExecutionDocuments'.

Die Herkunft eines Datensatzes ist dabei in PODSL über den Verweis auf den

Originaldatensatz und den Identifier gespeichert, da dieser alle Informationen zum ursprünglichen Repositorium enthält. Wird ein Datensatz verändert, wird der Originaldatensatz nicht gelöscht, sondern eine neue Version des Datensatzes angelegt. Dabei kann die Modifikation und der Zeitpunkt oder aber der Verantwortliche der letzten Änderung über den Verweis auf den Prozess der Konvertierung ermittelt werden. Somit kann über die 'ProvenaceTable' die Herkunft eines Datensatzes und alle Veränderungen lückenlos dokumentiert werden.

5.3.8 Verwendung von kontrolliertem Vokabular

In der Biodiversitätsinformatik werden die Werte vieler Attribute über ein kontrolliertes Vokabular angegeben. Die Begriffe in einem kontrollierten Vokabular werden in dieser Arbeit als Schlagworte bezeichnet. Dabei wird für ein Attribut eine Liste an erlaubten Werten spezifiziert, der z.B. für die Klassifikation eines Belegs benötigt wird. Ein gutes Beispiel hierfür ist die Verwendung von taxonomischen Gruppen in DiversityCollection. Diese sind z.B. 'Pflanze', 'Pilz' oder 'Säugetier'. Die taxonomischen Gruppen enthalten Information in Textform.

Allerdings ist die Verwendung des Datentyps 'String' zur Beschreibung von taxonomischen Gruppen nicht geeignet, da in diesem Fall die freie Texteingabe möglich wäre. Damit können ungeeignete Werte für die taxonomische Gruppe wie 'Buch' oder 'Pizza' von einem Anwender eingegeben werden. Die Verwendung von Schlagworten ermöglicht es, die zulässigen Werte für ein Attribut auf eine Auswahl zu beschränken. In der objektorientierten Programmierung ist dies durch die Definition von *Enum*s möglich, welche die erlaubten Werte enthalten und in RDBMS können Schlagworte in speziellen Tabellen verwaltet werden.

In PODSL ist die Verwaltung von Schlagworten analog dazu über *Enum*s, welche für jeden Typ eines Schlagwortes eine Liste erlaubter Begriffe enthalten. Diese erlaubten Begriffe sind auch hier im Allgemeinen Strings, können nun aber einem bestimmten Schlagworttyp wie 'taxonomische Gruppe' zugeordnet werden. Dazu wird bei der Modellierung eines Attributs für eine Entität in PODSL der Typ des Schlagwortes als Datentyp angegeben. Auf diese Weise wird modelliert, dass ausschließlich Begriffe des kontrollierten Vokabulars als Werte für die Belegung des Attributs verwendet werden dürfen. Schlagworte werden dabei häufig für Modalitäten z.B. bei der Sicherheit einer Bestimmung verwendet.

5.4 PODSL

Der folgende Abschnitt beschreibt die Struktur und Modellelemente der verschiedenen Ebenen von PODSL und PODSL-Biodiv. Dazu wird in Abschnitt 5.4.1 die Metastruktur von PODSL vorgestellt. In Abschnitt 5.4.2 werden zentrale Komponenten des Metamodells von PODSL auf M_2-Ebene spezifiziert. In den Abschnitten 5.4.2 und 5.4.3 wird die M_1-Ebene des Metamodells beschrieben. Dabei enthält Abschnitt 5.4.2 generische Elemente von PODSL. In Abschnitt 5.4.3 wird mit PODSL-Biodiv eine domänenspezifische Erweiterung für die Biodiversitätsinformatik formuliert. Die vollständige Modellierung von PODSL auf allen Metaebenen mit der Modellierung von PODSL-Biodiv findet sich in Anhang C. Da OMME für die Spezifikation von Modellelementen den Begriff des 'Konzepts' verwendet, wird dieser auch für die Spezifikation von Elementen in PODSL verwendet. Die Visualisierung von Entitäten und Dokumenten zur Prozessausführung erfolgt dabei über die Darstellung der Ableitungshierarchien. Die einzelnen Attribute und Felder dieser Modellelemente werden aufgrund der Komplexität von PODSL nicht grafisch dargestellt, sondern sind der technischen Spezifikation zu entnehmen.

5.4.1 Metastruktur

PODSL wurde mit Hilfe von OMME erstellt. Die Einbettung von PODSL in eine Metastruktur erfolgt dabei über drei Ebenen. Dabei enthält die M_2-Ebene die grundlegenden Konzepte der Modellierungssprache. Auf der M_1-Ebene werden Konzepte erstellt, welche in verschiedenen Domänen zum Einsatz kommen können. Diese bilden den generischen Kern von PODSL. Auf der M_1-Ebene werden zusätzlich domänenspezifische Erweiterungen spezifiziert. Eine konkrete domänenspezifische Erweiterung ist dabei PODSL-Biodiv für die Domäne der Biodiversitätsinformatik (siehe Abschnitt 5.4.3). Auf der M_0-Ebene wird die M_1-Ebene durch konkrete Instanzen von Entitäten besiedelt. Die Besiedelung der M_0-Ebene erfolgt dabei stets auf Basis einer domänenspezifischen Erweiterung. Da OMME für die Erstellung von Modellen und Metamodellen und nicht für deren Besiedlung entwickelt wurde, ist die Besiedlung der M_0-Ebene über OMME zwar möglich aber unübersichtlich und mit hohem Aufwand verbunden. Dementsprechend ist es vorteilhaft für die Besiedlung von PODSL-Biodiv, Repräsentationen von PODSL-Biodiv in anderen Technologien zu verwenden (siehe Abschnitt 5.5.2). Damit verfügt PODSL über einen Aufbau wie in Abbildung 5.17 dargestellt ist. In der Spezifikation der Metastruktur von PODSL wird bei der Spezifikation einiger Elemente auf die in [250] beschriebene 'Deep Instantiation' zurückgegriffen, so dass in M_2 spezifizierte Eigenschaften auch auf M_0

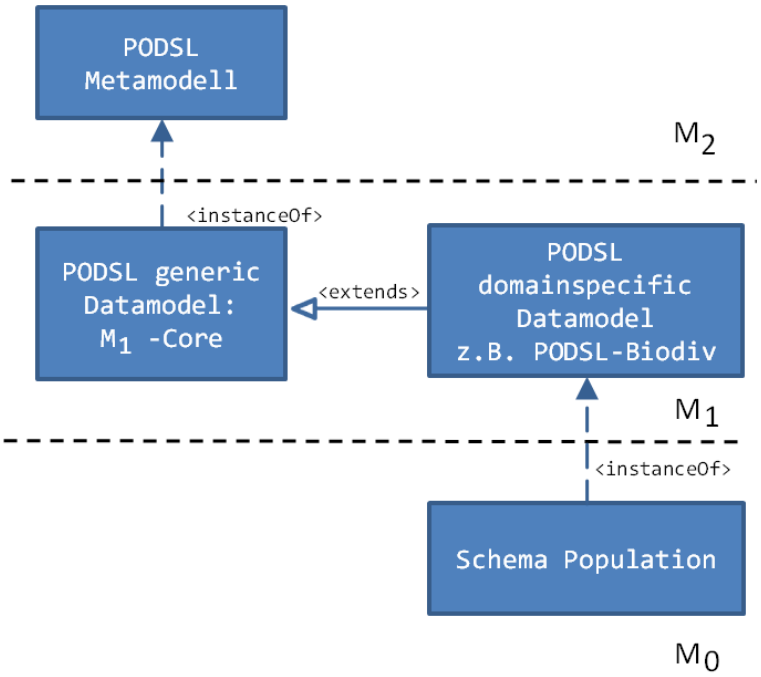

Abbildung 5.17: Metastruktur von PODSL

instanziiert werden können.

5.4.2 M2-Ebene

Auf M_2 wird eine allgemeine Modellierungssprache zur Erstellung von Modellen mit PODSL spezifiziert. In dieser wird festgelegt, auf welche Weise in der M_1-Ebene modelliert werden kann. Alle Modellierungselemente von PODSL werden eindeutig mit einem Identifier gekennzeichnet, so dass dieser stets bei der Verwendung eines Modellelements referenziert werden kann.

Wie in der ER-Modellierung wird auf die Entität als zentrales Konzept zurückgegriffen. Die Modellierung einer Entität in OMME ist als das Konzept 'EntityType' in Listing 5.1 dargestellt. Dabei wird einem EntityType ein Identifier zugewiesen, welcher auch durch Instanzen von Identifiern auf der M_0-Ebene belegt werden kann. Dies wird durch das Schlüsselwort 'defferdBy' angezeigt und ist eine Verwendung von 'Deep Instantiation' wie sie in [250] beschrieben wird. Dies hat den Vorteil, dass zum einen Konzepte, welche Instanzen von 'EntityType' sind, wie z.B. 'Person' auf der M_1-Ebene und Instanzen von Person wie z.B. 'Josef Simmel' auf der M_0-Ebene, eindeutig referenziert werden können. Die Bildung dieser Identifier erfolgt nach dem Prinzip, das in Abschnitt 5.3.4 vorgestellt wurde. Entitäten sind in PODSL primär aus Attributen aufgebaut und können Beziehungen mit anderen Entitäten eingehen.

```
1  concept EntityType
2      {
3        1..1 concept Identifier identifier deferredBy 2;
4        1..1 string name;
5        0..* concept Attribute attribute;
6        0..* concept Relation relation;
7      }
```

Listing 5.1: EntityType auf der M_2-Ebene

Das Konzept der 'Relation' muss dabei in PODSL mehr Anforderungen erfüllen als in dem in [250] vorgestellten Metamodell der ER-Modellierung. Diese zusätzlichen Anforderungen sind aus Listing 5.2 zu entnehmen. So werden Beziehungen zusätzlich durch Modalitäten beschrieben und können verschiedenen Dokumentenaspekten zugeordnet werden. Zunächst wird aber durch eine Beziehung ein Zusammenhang zwischen zwei Entitätstypen hergestellt. Analog zur ER-Modellierung werden Kardinalitäten verwendet, um zu modellieren mit wie vielen Entitäten eines Entitätstyps eine Beziehung eingegangen werden kann. Dabei werden 1:1 und 1:n-Beziehungen unterstützt, die als optional gekennzeichnet werden können. Durch die Belegung von 'role' wird durch ein Schlüsselwort modelliert, was für eine Beziehung besteht. Z.B. wird durch das Schlüsselwort 'FormOfCoexistence' in der 'FormOfCoExistenceRelation', welche eine Instanz von 'Relation' in PODSL-Biodiv ist, angezeigt, dass diese Beziehung die Art des Zusammenlebens zwischen zwei biologischen Organismen beschreibt.

```
1 concept Relation
2       {
```

```
3         0..1 string role;
4         0..1 concept Aspect aspect;
5         1..1 concept Identifier identifier;
6         1..1 concept EntityType source;
7         1..1 concept EntityType target;
8         1..1 enum Cardinality cardinality = Cardinality.one;
9         0..* concept Modality modality;
10      }
```

Listing 5.2: Relation auf der M_2-Ebene

Um diese Beziehungen genauer zu beschreiben, werden in PODSL Modalitäten verwendet. Diese stellen Attribute von Beziehungen dar. Der Aufbau einer 'Modality' ist in Listing 5.3 dargestellt. Sie werden dazu verwendet, die Art der Beziehung genauer zu beschreiben wie z.B. die Sicherheit der Identifikation eines biologischen Objektes. Um dies zu erfassen, verfügt die 'Modality' über ein Attribut.

```
1   concept Modality{
2         1..1 concept Identifier identifier;
3         1..1 string name;
4         1..1 concept Attribute attribute;
5       }
```

Listing 5.3: Modality auf der M_2-Ebene

Beziehungen, die bei der Dokumentation einer Prozessausführung in einem 'ProcessExecutionDocument' verwendet werden, werden zusätzlich einem Dokumentenaspekt zugeordnet. Die Spezifikation eines Dokumentenaspektes findet sich in Listing 5.4. In diesem wird beschrieben, welchem Dokumentenaspekt eine Beziehung bei der Prozessausführung zugeordnet ist. Dies ist z.B. der organisatorische Dokumentenaspekt für die Beziehung von dem 'ProcessExecutionDocument' zu der Person, welche für die Ausführung des Prozesses verantwortlich ist.

```
1 concept Aspect
2         {
3           1..1 concept Identifier identifier;
4           1..1 string name;
5         }
```

Listing 5.4: Aspect auf der M_2-Ebene

Die zentralen Dokumente zur Erfassung von Prozessen werden in den 'ProcessExecutionDocuments' in Listing 5.5 spezifiziert. Diese sind über Vererbung von 'Enti-

tyType' abgeleitet und verfügen somit über alle Eigenschaften eines 'EntityTypes'. In einem 'ProcessExecutionDocument' werden zur Erfassung der Prozessausführung zwei Arten von Beziehungen zwingend vorgeschrieben. Zum einen eine Beziehung auf das Objekt der Prozessausführung und zum anderen auf die Dokumentenaspekte, welche die Art der Prozessausführung genauer beschreiben. Die Beziehungen, die hierfür verwendet werden, müssen Dokumentenaspekten zugeordnet sein.

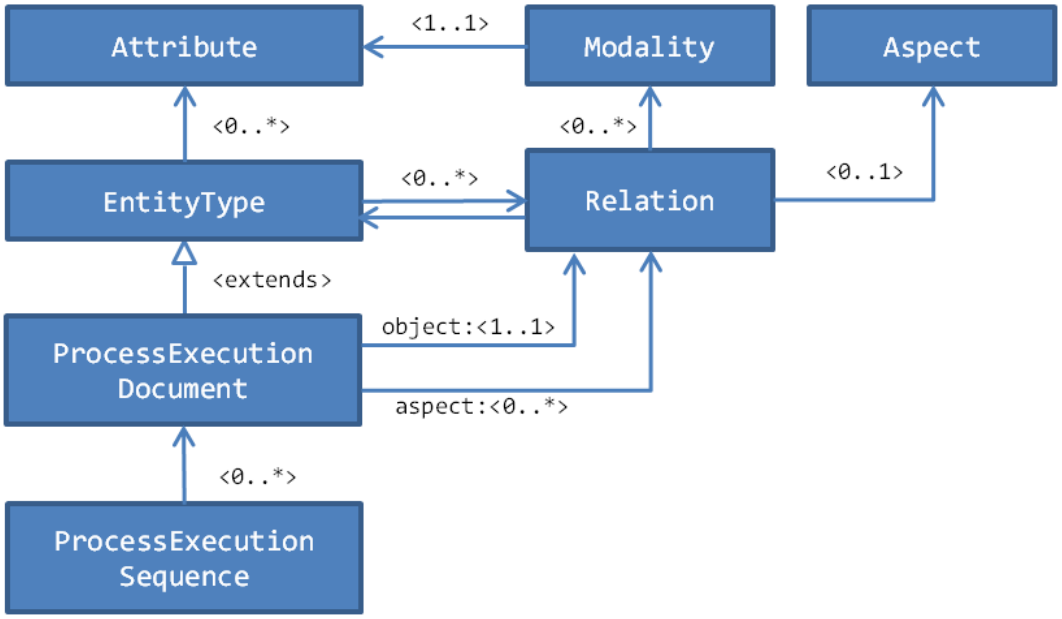

Abbildung 5.18: Zentrale Konzepte auf der M2-Ebene von PODSL und ihre Beziehungen zueinander

```
1  concept ProcessExecutionDocument extends EntityType
2      {
3        1..1 concept Relation object;
4        0..* concept Relation aspect;
5        0..* concept ProcessExecutionSequence sequences;
6      }
```

Listing 5.5: ProcessExecutionDocument auf der M_2-Ebene

Die Übersicht über die wesentlichen Konzepte der M_2-Ebene von PODSL ist in Abbildung 5.18 mit Kardinalitäten dargestellt. Bei der Darstellung von PODSL wurde auf die Aufnahme von Hilfskonzepten wie dem Identifier aus Gründen der Übersichtlichkeit verzichtet. Die Modellierung der M_2-Ebene von PODSL ist in Anhang C vollständig aufgelistet.

Das generische Datenmodell auf der M_1-Ebene: $M_1 - Core$

Im $M_1 - Core$ Modell von PODSL werden Instanzen von Konzepten der M_2-Ebene gebildet, welche von allgemeiner Natur sind, so dass diese in verschiedenen Domänen verwendet werden können. Da in PODSL auf der M_1-Ebene Entitäten, Beziehungen, Modalitäten, Dokumentenaspekte und 'ProcessExecutiondocuments' ausschließlich durch Vererbung aus bekannten Konzepten erweitert werden können, müssen in

$M_1 - Core$ zwingend alle Basiskonzepte als Grundlage einer Ableitungshierarchie enthalten sein. Darüber hinaus ist die Unterstützung von Data Provenance für alle Anwendungsdomänen relevant, so dass diese auch Teil von $M_1 - Core$ ist. Des Weiteren gibt es eine Vielzahl an Entitäten, Beziehungen und Prozessen, die in mehreren Domänen verwendet werden, die somit auch $M_1 - Core$ zugeordnet sind.

Für die Unterstützung von Vererbung werden in $M_1 - Core$ zunächst die Basiskonzepte 'BaseEntity', 'BaseRelation', 'BaseModality', 'BaseAspect' und 'BaseProcessExecutionDocument' als abstrakte Konzepte eingeführt, so dass diese nicht auf der M_0-Ebene instanziiert werden können. Diese stellen jeweils die Wurzel der Ableitungshierarchie dar. Als Beispiel ist die Spezifikation der 'BaseEntity' in Listing 5.6 dargestellt. Für diese werden grundlegende Werte für das Schema des 'EntityTypes' der M_2-Ebene festgelegt, über die alle Entitäten in PODSL verfügen sollen. Dies ist insbesondere die Unterstützung von Data Provenance, so dass für alle Entitäten in PODSL die 'ProvenanceRelation' vorgeschrieben ist. Darüber hinaus enthält die BaseEntity die Attribute 'Name' und 'Identifier'. Über den Identifier wird die 'BaseEntity' eindeutig referenziert. Diese Wertbelegungen sind dabei aus Gründen der Vollständigkeit angegeben und müssen von abgeleiteten Klassen überschrieben werden. In der objektorientierten Programmierung wird dies durch das Schlüsselwort 'virtual' modelliert. Dies kann allerdings aktuell in OMME nicht dargestellt werden, da dieses Schlüsselwort in den aktuellen Versionen von OMME nicht unterstützt wird.

```
1 abstract EntityType BaseEntity{
2         identifier=BaseEntityIdentifier;
3         name="BaseEntity";
4         relation=ProvenanceRelation, CreatorRelation,
              CreationTimeRelation;
5       }
```

Listing 5.6: BaseEntity auf der M_1-Ebene

Zur genaueren Modellierung von Entitäten werden ausgehend von BaseEntity speziellere Entitäten abgeleitet. In $M_1 - Core$ sind dies beispielsweise die Entitäten 'Person' und 'Institut', welches über die 'OrganizationalBaseEntity' mittelbar von 'BaseEntity' abgeleitet werden. Die Spezifikation von Person ist in Listing 5.7 dargestellt. Durch Ableitung von 'Person' lässt sich dieses Konzept weiter spezialisieren.

```
concept Person extends OrganizationalBaseEntity{
        name="Person";
        identifier=PersonIdentifier;
        relation=DateOfBirthRelation;
        attribute=firstName,lastName;
      }
```

Listing 5.7: Person auf der M_1-Ebene

Analog hierzu werden ausgehend von den anderen Basiskonzepten Spezialisierungen gebildet. An dieser Stelle zeigt sich auch die Vorteilhaftigkeit der Verwendung von Vererbung in PODSL durch das Prinzip der Ersetzbarkeit. Dazu soll die 'PartWholeRelation', wie sie in Listing 5.8 dargestellt ist, betrachtet werden. Diese erlaubt die Verknüpfung von zwei 'BaseEntities', so dass diese Beziehung zwischen allen Entitäten in PODSL bestehen kann. Die 'PartWholeRelation' beschreibt, ob zwischen zwei beliebigen Entitäten eine Teil-Ganzes Beziehung besteht.

```
concept PartWholeRelation extends BaseZeroOrMoreRelation{
        role="PartWholeRelation";
        identifier=PartWholeRelationIdentifier;
        source=BaseEntity;
        target=BaseEntity;
        modality=PartWholeModality;
      }
```

Listing 5.8: PartWholeRelation auf der M_1-Ebene

Im Gegensatz dazu stellt die 'OwnershipRelation' (Listing 5.9) höhere Anforderungen an die Spezialisierung der zu verknüpfenden Entitäten. Diese Beziehung besteht zwischen einer Entität 'BaseDataSet' und einer 'OrganizationalBaseEntity'. Es können somit nur diese Entitätstypen oder Spezialisierungen dieser verknüpft werden. Damit kann in einer 'OwnershipRelation' sowohl eine Person, als auch ein Institut als 'target' eingesetzt werden, da diese beiden Konzepte von der 'OrganizationalBaseEntity' als Basisklasse abgeleitet werden.

```
1 concept OwnershipRelation extends BaseOneRelation{
2         role="Ownership";
3         aspect=OrganizationalAspect;
4         identifier=OwnershipRelationIdentifier;
5         source=BaseDataSet;
6         target=OrganizationalBaseEntity;
7       }
```

Listing 5.9: OwnershipRelation auf der M_1-Ebene

Der Vorteil dieser Strukturierung liegt in der Anwendung von PODSL in Infrastrukturen zwischen heterogenen Datenspeichern (siehe Kapitel 8). Hier kann der Fall auftreten, dass ein Datenspeicher Daten nicht in derselben Genauigkeit erfassen kann, wie ein anderer Datenspeicher, mit dem dieser Daten austauscht. Da allerdings beide Datenspeicher PODSL bis zu einem gewissen Spezialisierungsgrad unterstützen, kann durch Rückgriff auf Basiskonzepte der Datenverlust minimiert werden.

$M_1 - Core$ enthält als generisches Modell eine umfangreiche Liste von Entitäten, Beziehungen, Modalitäten, Dokumentenaspekten und 'ProcessExecutionDocuments'. Aus Gründen der Übersichtlichkeit werden diese an dieser Stelle nicht vollständig aufgeführt, sondern eine Auswahl angegeben. All diesen Konzepten ist gemein, dass sie über allgemeinen Charakter verfügen, so dass diese für alle Anwendungsdomänen relevant sind. Für das vollständige Modell von $M_1 - Core$ wird auf den Anhang C verwiesen.

- Entitäten: BaseEntity, OrganizationalBaseEntity, LocalBaseEntity, Person, Institute, Locality, Device, BaseDataSet, Date, Time, DateTime, ProvenanceTable
- Beziehungen: BaseRelation, BaseOneRelation, BaseOneOrMoreRelation, StartEnd, PartWhole, Responsible, ExecutionTime, ExecutionLocality
- Modalitäten: BaseModality, PartWholeModality, BaseConversionModality
- Dokumentenaspekte: BaseAspect, FunctionalAspect, OrganizationalAspect, OperationalAspect
- ProcessExecutionDocuments: BaseProcessExecutionDocument, BaseConversion, Conversion

Zuletzt soll ein besonderes Augenmerk auf die Unterstützung von Data Provenance gelegt werden. Dieses ist in PODSL über das Konzept 'ProvenanceTable' in $M_1 - Core$ angesiedelt. Der Aufbau der 'ProvenanceTable' ist in Listing 5.10 dargestellt. Dabei wird über die 'OriginalRelation' eine Referenz über eine 1:1 Beziehung auf den Originaldatensatz gesetzt. In der 'SiblingsRelation' werden alle anderen Versionen des Originaldatensatzes über eine 1:n Beziehung referenziert. Mit der 'ConversionRelation' werden die Dokumente referenziert, welche den Prozess der Konvertierung eines Datensatzes beschreiben. Dazu ist in $M_1 - Core$ 'Conversion' ein spezialisiertes 'ProcessExecutionDocument' enthalten, welches genau diesen Prozess erfasst.

```
1 concept ProvenanceTable extends DataOrientedBaseEntity{
2         name="ProvenaceTable";
3         identifier=ProvenanceTableIdentifier;
4         relation=OriginalRelation, SiblingsRelation, ConversionRelation;
5       }
```

Listing 5.10: ProvenanceTable auf der M_1-Ebene

Damit enthält die 'ProvenanceTable' alle Eigenschaften, die für die Unterstützung von Data Provenance – wie in Abschnitt 5.3.7 beschrieben – benötigt werden.

5.4.3 Das domänenspezifische Modell für die Biodiversitätsinformatik: PODSL-Biodiv

PODSL-Biodiv ist eine domänenspezifische Erweiterung von $M_1 - Core$ zur Erfassung der Biodiversitätsforschung. Für die Modellierung von PODSL-Biodiv wurden die in Kapitel 4.4 vorgestellten Datenstandards und die Anwendungsfälle UC1-UC5 aus Abschnitt 4.2.5 herangezogen. Ziel der Modellierung von PODSL-Biodiv war es, die Domäne der Biodiversitätsforschung möglichst vollständig zu beschreiben und die Grundlagen zu Erweiterung zu legen. Da DwC als Datenstandard in der Biodiversitätsinformatik von besonderer Bedeutung und auch zu einem gewissen Teil mit ABCD kompatibel ist,wurde für die Berücksichtigung von etablierten Datenstandards in PODSL-Biodiv der Datenstandard DwC gewählt. Im aktuellen Schema von PODSL-Biodiv ist DwC bis auf die Berücksichtigung von geologischen Bezügen vollständig integriert [10] und es wurde ein Mapping zwischen DwC und PODSL-Biodiv

[10] Auf die Abbildung geologischer Bezüge wurde verzichtet, da diese kein unmittelbarer Bestandteil der Domäne Biodiversitätsforschung sind. Diese können aber in PODSL-Biodiv integriert werden.

erstellt. Das Mapping ist in Anhang B aufgelistet.

Der Kern von PODSL-Biodiv besteht aus Dokumenten zur Erfassung von Prozessen, welche in der Biodiversitätsfosrchung von besonderer Bedeutung sind. Grundlage für die Auswahl der Prozesses sind die Anwendungsfälle UC1-UC5. Die Rechtfertigung für dieses Auswahl liegt zum einen in den praktischen Erfahrungen aus dem IBF-Projekt, in welchem eine Vielzahl von Prozessen in diese Domäne analysiert werden konnte. Zum anderen liegt diese in der Begründung der Repräsentativität der Anwendungsfälle UC1-UC5 in Abschnitt 4.2.5. Zusätzlich werden in PODSL-Biodiv Entitäten spezifiziert und in eine Ableitungshierarchie eingebettet. Die aktuelle Mächtigkeit von PODSL-Biodiv umfasst damit die Anforderungen der Anwendungsfälle UC1-UC5 und die Mächtigkeit von DwC mit Ausnahme der geologischen Bezüge (siehe Anhang B). Da die Anforderungen des IBF-Projektes auf Grundlage von DiversityCollection formuliert wurden, kann PODSL-Biodiv außerdem alle relevanten Fälle aus DiversityCollection abbilden. Damit bildet PODSL-Biodiv die wichtigsten Anwendungsfälle der Biodiversitätswissenschaft ab und ist als domänenspezifisches Datenmodell für die Biodiversitätsinformatik geeignet.

Über PODSL-Biodiv wird damit für die Domäne der Biodiversitätsforschung über eine prozessorientierte Sichtweise ein umfangreiches Datenmodell zur Verfügung gestellt. Ein zentrales Element ist hierbei die Unterteilung eines Prozesses in Subprozesse, so dass für komposite Prozesse wie Begehungen, Kartierungen oder den Umgang mit Belegen eine strukturierte Möglichkeit der Erfassung in spezialisierten Dokumenten der Prozesserfassung ermöglicht wird. Teil von PODSL-Biodiv ist auch ein kontrolliertes Vokabular in der Form, wie dies in Abschnitt 5.3.8 beschrieben wurde.

Aufgrund des großen Umfangs von PODSL-Biodiv kann diese an dieser Stelle nicht vollständig vorgestellt werden. Der interessierte Leser sei hierzu auf Anhang C verwiesen. Es werden Modellelemente und ihre Beziehungen zueinander in diesem Anschnitt anhand von Beispielen vorgestellt, so dass dem Leser ein Überblick über die Möglichkeiten der Datenstrukturierung mit PODSL-Biodiv gegeben wird.

Ein zentraler Begriff in der Biodiversitätsinformatik ist der Begriff 'Taxon'. Ein Taxon beschreibt die Zuordnung eines biologischen Objektes zu einem Element der Taxonomie in der Biologie [226] – in der Praxis der Biodiversitätsinformatik meistens zu einer Art. Es gibt verschiedene Taxonomien nach denen ein biologisches Objekt einem Taxon zugeordnet werden kann. Dementsprechend ist es von entscheidender Bedeutung nicht nur den wissenschaftlichen Namen zu dokumentieren, sondern auch die Taxonomie zu erfassen nach der die Klassifikation erfolgt. Aufgrund der herausragenden Bedeutung der Taxonomie, ist der Basisklasse Taxon in DwC eine Vielzahl

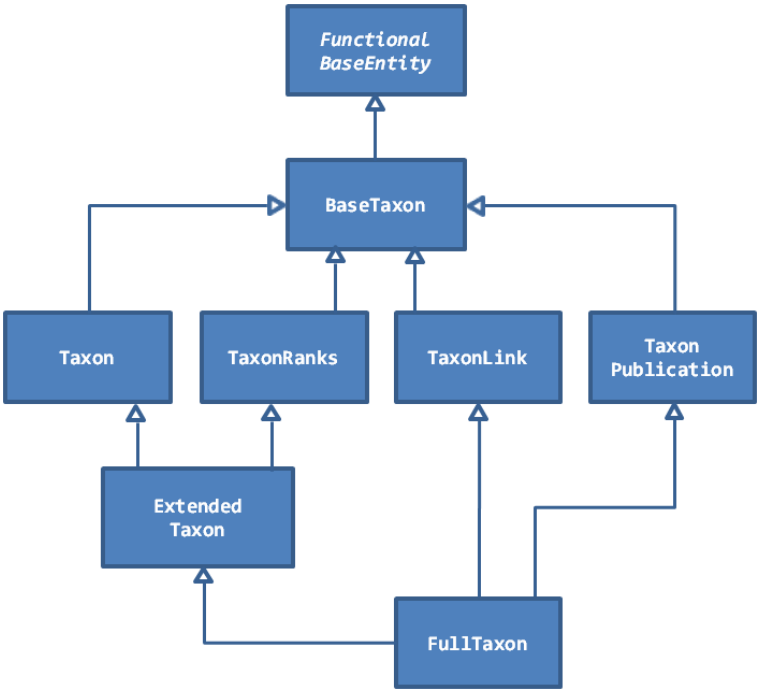

Abbildung 5.19: Ableitungshierarchie für Taxon-Konzepte

an Attributen zugeordnet. In PODSL-Biodiv werden aber nicht alle Felder in jedem Kontext benötigt. Dementsprechend wird die taxonomische Klassifikation über eine Ableitungshierarchie beschreiben. Dabei enthält das Konzept 'BaseTaxon' die grundlegenden Attribute zur Arbeit mit Taxa und das Konzept 'FullTaxon' alle Attribute der Taxonspezifikation aus DwC. Die PODSL-Ableitungshierarchie für Taxa ist in Abbildung 5.19 dargestellt.

Die zentrale Aufgabe bei der Felddatenerfassung ist es über den Prozess der 'Identification' ein biologisches Objekt einem Taxon zuzuordnen. In PODSL wird dieser Prozess über das 'ProcessExecutionDocument' 'Identification' erfasst, wie in Listing 5.11 dargestellt ist. Im funktionalen Dokumentenaspekt ist die Zuordnung des biologischen Objektes zu einem Taxon beschrieben. Im organisatorischen Dokumentenaspekt wird der Verantwortliche der Bestimmung und im operationalen Dokumentenaspekt werden die verwendeten Hilfsmittel erfasst. Damit ist das 'ProcessExecutionDocument' 'Identification' dazu geeignet, den Prozess der Taxonbestimmung zu dokumentieren.

```
1 concept Identification extends BaseProcessExecutionDocument{
2     name="Identification";
3     identifier=IdentificationIdentifier;
4     object=BiologicalObjectRelation;
```

```
5       relation=ResponsibleScientistRelation, TaxonDeterminationRelation,
6       ExecutionTimeRelation, WorkOfReferenceRelation;
7   }
```

Listing 5.11: Prozessausführung einer Identification in PODSL-Biodiv

Ein weiterer Prozess ist der Prozess der Georeferenzierung. Dieser wird immer dann ausgeführt, wenn es die Bestimmung des Ortes nicht nur in beschreibender Form erfolgen soll, sondern als Messung ausgeführt wird und somit auch die Ausführungszeit und der Verantwortliche erfasst werden soll. Dieser Prozess ist in Listing 5.13 dargestellt. Da die Georeferenzierung eine Messung darstellt, wird das 'ProcessExecutionDocument' 'Georeferencing' von 'BaseMeasuring' (Abbildung 5.12) abgeleitet. Dieses enthält bereits eine Vielzahl von Dokumentenaspekten, die im Rahmen einer Messung aufgenommen werden müssen. Zusätzlich wird im Rahmen einer Georeferenzierung eine Spezialisierung der Messung vorgenommen, so dass diesem Prozess im funktionalen Dokumentenaspekt die Bestimmung der Koordinaten zugeordnet ist. Die anderen Dokumentenaspekte, welche bei der Ausführung des Prozesses dokumentiert werden sollen, werden von 'BaseMeasuring' geerbt.

```
1 concept BaseMeasuring extends BaseProcessExecutionDocument{
2       name="BaseMeasuring";
3       identifier=BaseMeasuringIdentifier;
4       object=ObservationObjectRelation;
5       relation=ExecutionTimeRelation, ExecutionLocalityRelation,
            ResponsibleScientistRelation, MeasurementMethodRelation,
6       WorkOfReferenceRelation, MeasurementValueRelation;
7   }
```

Listing 5.12: Prozessausführung einer allgemeinen Messung in PODSL-Biodiv

```
1 concept GeoReferencing extends BaseMeasuring{
2       name="GeoReferencing";
3       identifier=GeoReferencingIdentifier;
4       relation=CoordinateDeterminationRelation, UsedDeviceRelation,
            MethodRelation;
5   }
```

Listing 5.13: Prozessausführung der Georeferenzierung in PODSL-Biodiv

Die Prozesse der Georeferenzierung und der Identifikation werden bei der Kartierung eines biologischen Objektes als Subprozesse ausgeführt. Dies ist in Listing 5.14 dargestellt. Dabei enthält das 'ProcessExecutionDocument' für die Kartierung (Ob-

servation) im verhaltensorientierten Dokumentenaspekt eine Beziehung auf dieses Subprozesse. Die Dokumentenaspekte, die für die Erfassung einer Kartierung benötigt werden wie z.B. der Verantwortliche oder die Ausführungszeit, werden über die Ableitung aus dem Basiskonzept 'BaseMonitoring' geerbt.

```
1  concept BaseMonitoring extends BaseProcessExecutionDocument{
2        name="BaseMonitoring";
3        identifier=BaseMonitoringIdentifier;
4        object=ObservationObjectRelation;
5        relation= ResponsibleScientistRelation, ExecutionTimeRelation,
             ExecutionLocalityRelation, WorkOfReferenceRelation,
6        RelatedProcessDocumentsRelation,MethodRelation;
7    }
8
9  concept Observation extends BaseMonitoring{
10     name="Observation";
11     identifier=ObservationIdentifier;
12     object=BiologicalObjectRelation;
13     relation=IdentificationSubProcess,GeoReferencingSubProcess;
14   }
```

Listing 5.14: Prozessausführung einer Kartierung in PODSL-Biodiv

Durch die Strukturierung in Prozessen erfasst PODSL-Biodiv die Biodiversitätsinformatik in Form von Dokumenten zur Prozessausführung und Ableitungshierarchien von Entitäten und ermöglicht so die Beschreibung der Domäne der Biodiversitätsforschung durch eine Vielzahl an Konzepten. Aufgrund des großen Umfangs von PODSL-Biodiv ist eine vollständige Darstellung nicht möglich. Für die vollständige Spezifikation von PODSL-Biodiv wird der Leser auf Anhang C verwiesen.

5.4.4 M0-Ebene

Auf der M_0-Ebene werden konkrete Datensätze als Instanzen der Entitäten von PODSL-Biodiv gebildet. Dabei ist zu beachten, dass PODSL-Biodiv vor allem vor dem Hintergrund entwickelt wurde, als eine logische Struktur zu dienen, so dass Daten möglichst leicht in andere Formate konvertiert werden können. Damit liegt das Einsatzgebiet von PODSL-Biodiv in der Definition eines Datenschemas für die Domäne der Biodiversitätsinformatik, welches Abbildungen in anderen Datenstrukturen ermöglicht und somit als Grundlage von Mappings dient. Die eigentliche Persistenz der Daten findet damit in der Praxis in Datenbanken, XML, Strukturen der objektorientierten Programmierung und Ontologien statt. Die Übertragung von Daten im

logischen Format von PODSL-Biodiv in andere Technologien wird in Abschnitt 5.5.2 beschrieben. So ist es zwar möglich mit PODSL-Biodiv Daten auf der M_0-Ebene zu instanziieren, aufgrund der geringen Relevanz in der Praxis wird an dieser Stelle darauf verzichtet.

5.5 Eigenschaften von PODSL

Da ein Datenstandard einem steten Wandel unterworfen ist, wird in Abschnitt 5.5.1 ein Prozess zur Anpassung eines Datenstandards an neue Herausforderungen vorgestellt, welcher dem Kriterium der Flexibilität aus Abschnitt 4.3 genügt. Da sich PODSL, wie in Abbildung 5.1 demonstriert ist, im Kontext von RDBMS, der objektorientierten Programmierung, XML und Ontologien befindet, wird in Abschnitt 5.5.2 die Erstellung eines Mappings von PODSL in die entsprechenden Technologien dargestellt. Die Verwendung von PODSL bei der Datenübertragung wird in Abschnitt 5.5.3 besprochen.

5.5.1 Prozess zur Sicherung der Flexibilität

Flexibilität hat die Anpassung von Schemata an neue Anforderungen zum Ziel und ist nach [32] ein wesentliches Element einer Modellierungssprache. Nach POSE ist die Definition eines Prozesses zur Integration neuer Anforderungen in ein bestehendes Dokument ein entscheidendes Kriterium. Das heißt, eine Modellierungssprache muss neue Elemente im Kontext zu den bisherigen Elementen aufnehmen können. Diese Anforderung kann durch die Einbettung in eine Metastruktur erreicht werden. Mit Hilfe dieses Metamodells wird ein Prozess definiert, mit dem neue Elemente in ein bestehendes Schema hinzugefügt werden. Gleichzeitig ist definiert, wie diese neuen Elemente im Kontext des Metamodells zu interpretieren sind.

In PODSL-Biodiv wird die Flexibilität durch die Erweiterung der Konzepte der M_1-Ebene realisiert. Ausgangspunkt für die Modellerweiterung sind stets bestehende Modellelemente, die als Basiskonzept für neue Konzepte dienen. Neue Modellelemente können damit nur durch Vererbung aus bestehenden Konzepten abgeleitet werden. Auf diese Weise ist ein neues Element nicht gänzlich unbekannt, sondern kann auf ein bekanntes Basiskonzept zurückgeführt werden. Soll ein neues Konzept nicht aus existierenden Konzepten von PODSL-Biodiv abgeleitet werden, da es sich fundamental von den existierenden Konzepten unterscheidet, muss $M_1 - Core$ darauf untersucht werden, ob $M_1 - Core$ ein geeignetes Basiskonzept enthält. Dieses enthält alle generischen Implementierungen der M_2-Ebene als Basiskonzepte, wie z.B. das 'BasePro-

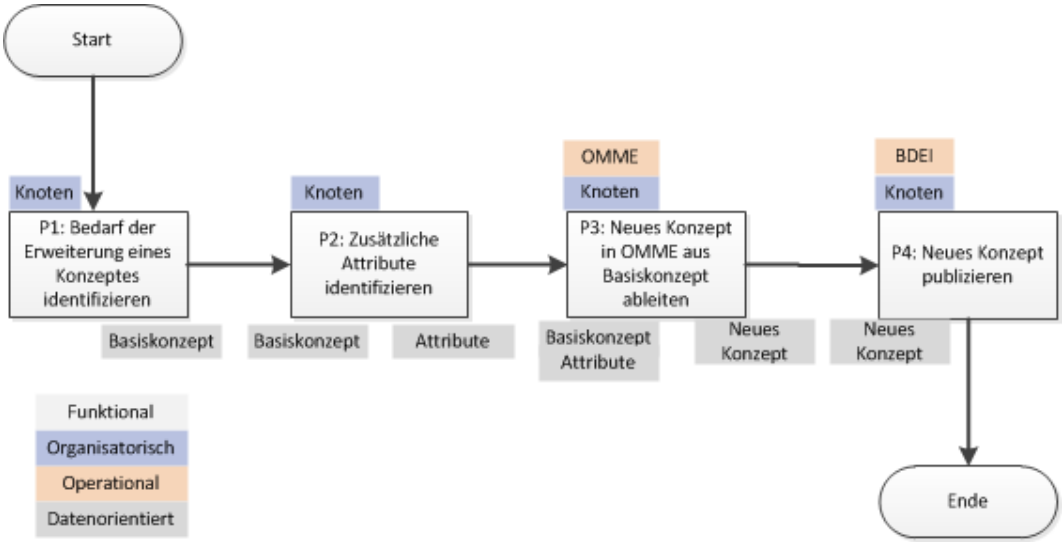

Abbildung 5.20: Prozess der Erweiterung von PODSL-Biodiv

cessExecutionDocument' oder die 'BaseEntity'. Ist in $M_1 - Core$ kein geeigneteres Konzept spezifiziert, kann somit zumindest immer auf diese Konzepte zurückgegriffen werden und ein neues Konzept kann damit immer in eine Ableitungshierarchie eingebettet werden.

Damit stellt die Erweiterung des Modells an sich selbst einen Prozess dar, wie in Abbildung 5.20 dargestellt ist. Zunächst muss von den Verantwortlichen in P1 die Entscheidung getroffen werden, dass das Modell erweitert werden muss, um die Anforderungen der Anwendungsdomäne abzubilden. Dieses sind in einer Infrastruktur wie BDEI (siehe Kapitel 8) die Knoten, welche als Schnittstelle zwischen Wissenschaftlern und den zentralen Organisationseinheiten fungieren. Anschließend wird in P2 ein existierendes Konzept als Basiskonzept ausgewählt und dieses um zusätzliche Attribute erweitert. In P3 wird schließlich das neue Konzept in OMME von der Basisklasse abgeleitet. Wenn das neue Konzept in einer Infrastruktur verwendet werden soll, muss dieses in der Infrastruktur publiziert werden. In P4 ist dies am Beispiel eine Publikation über BDEI modelliert.

Diese Art der Sicherung der Flexibilität hat einen enormen Vorteil bei der Anwendung in Infrastrukturen wie BDEI. In diesen Infrastrukturen ist es normalerweise nicht möglich, Modellerweiterungen synchronisiert mit allen Teilnehmern auszuführen. Somit wird es stets Teilnehmer an der Infrastruktur geben, die mit ihren Datenstrukturen bestimmte Modellelemente nicht unterstützen. Allerdings ist diesen Teilnehmern stets ein Basiskonzept eines neuen Elements bekannt. So kann durch Rückgriff auf das Basiskonzept mit den neuen Elementen gearbeitet werden.

Damit wird durch den Prozess aus Abbildung 5.20 die Flexibilität von PODSL-Biodiv garantiert. Der Prozess kann dabei für lokale Erweiterungen im Prinzip von jedem Teilnehmer einer Infrastruktur ausgeführt werden. Im Rahmen der Biodiversitätsinformatik empfiehlt es sich, diesen für offizielle Erweiterungen auf den Tagungen der TDWG durch ein Fachgremium auszuführen. Aufgrund der Flexibilität ist allerdings jeder Datenspeicher dazu in der Lage eine lokale Erweiterung zu erstellen und weiterhin auf der Grundlage der Basiskonzepte mit anderen Teilnehmern der Infrastruktur zu kommunizieren. Somit ist die Flexibilität eines Datenmodells in PODSL-Biodiv auch auf Ebene der lokalen Datenspeicher garantiert.

5.5.2 Mapping auf andere Technologien

Innerhalb einer Infrastruktur werden Datenspeicher auf Basis verschiedener Technologien eingesetzt. PODSL-Biodiv dient in einer solchen Infrastruktur als logisches Datenmodell, welche die Abbildungen von Daten aus verschiedenen Schemata und Technologien moderiert. Damit dies erfolgen kann, muss die logische Struktur von PODSL-Biodiv in verschiedenen Technologien repräsentiert werden können. Dementsprechend wird im folgenden Abschnitt gezeigt, wie die logische Struktur von PODSL-Biodiv zur Erstellung von Datenbanken, XML-Dokumenten, Ontologien und in der objektorientierten Programmierung dargestellt werden kann.

Dabei wird für jede dieser Technologien eine Repräsentation von PODSL-Biodiv erzeugt. Innerhalb einer Infrastruktur muss ein Datenspeicher diese nicht unterstützen, sondern kann auch über einen Wrapper an PODSL-Biodiv angebunden werden. Die Repräsentationen ermöglichen es aber, einen Datenspeicher direkt nach dem logischen Schema von PODSL-Biodiv zu erzeugen. Hat ein Datenspeicher keine zusätzlichen Aufgaben ist aufgrund der Einfachheit die interne Strukturierung des Datenspeichers nach diesen Repräsentationen empfehlenswert.

Mapping auf Datenbanken

Die Nähe des M_2-Metamodells zur ER-Modellierung ermöglicht eine einfache Umsetzung von PODSL-Biodiv in eine Datenbank. Dabei wird ausgenutzt, dass die M_2-Ebene von PODSL mit dem M_2-Modell der ER-Modellierung nach [250] korreliert und somit über Entitäten und Beziehungen verfügt. Kern der Repräsentation von PODSL-Biodiv als Datenbankstruktur ist die M_1-Ebene von PODSL-Biodiv. Dazu werden analog zur ER-Modellierung für alle Entitäten Tabellen gebildet, welche die Attribute des jeweiligen Konzepts aus PODSL-Biodiv enthalten. Da das 'ProcessExecutionDocument' auf der M_2-Ebene von 'EntityType' abgeleitet wird,

werden für diese ebenfalls Tabellen erstellt. Die Identifier in PODSL dienen dabei als Schlüssel für die Datensätze in den Tabellen. Die Abbildung von Beziehungen erfolgt analog zur ER-Modellierung über ein Attribut in den verknüpften Tabellen. In der ER-Modellierung ist es dabei freigestellt, in welcher Entität die andere über einen Fremdschlüssel referenziert wird. Dies soll auch für die Repräsentation von PODSL-Biodiv als Datenbank möglich sein. In der klassischen ER-Modellierung sind allerdings PODSL-spezifische Elemente wie Dokumentenaspekte oder Modalitäten nicht enthalten. Diese beiden Elemente beschreiben in PODSL die Art und Weise einer Beziehung und entsprechen damit den Attributen einer Relation in der ER-Modellierung. Diese werden bei der Umsetzung in eine Datenbankstruktur der 'source'-Entity nach der M_2-Spezifikation von PODSL als zusätzliches Attribut zugewiesen (siehe Listing 5.2). Ein weiteres Element der M_2-Ebene von PODSL, welche nicht in der klassischen ER-Modellierung enthalten ist, ist die 'ProcessExecutionSequence'. Diese stellt eine Erweiterung von 'EntityType' dar und wird dementsprechend in der Datenbank über eine separate Tabelle realisiert.

Ein wichtige Aufgabe beim Mapping zwischen Datenbanken und PODSL-Biodiv ist die Berücksichtigung der Ableitungshierarchien aus PODSL. Diese werden in klassischen RDBMS nicht unterstützt. Somit stellt sich die Frage, an welchem Punkt in der Vererbungshierarchie eine spezielle Tabelle für die Erfassung der Daten gebildet werden soll. Dabei soll gelten, dass für die Erfassung eines Prozesses jeweils eine neue Tabelle erstellt wird, da in diesen verschiedene Sachverhalte abgebildet werden. Für die Erstellung von Tabellen von Entitäten aus PODSL liegt es im Entscheidungsbereich des Datenbankdesigners, inwieweit er die Spezialisierung einer Entität unterstützen möchte. Dabei müssen allerdings zentrale Konzepte von PODSL-Biodiv wie 'Locality', 'Person', 'Institute' oder 'Taxon' in Form einer separaten Tabelle realisiert werden. Für diese Konzepte bietet PODSL-Biodiv verschiedene Spezialisierungsgrade an. Soll PODSL-Biodiv vollständig von der Datenbankstruktur unterstützt werden, müssen alle Attribute dieser Spezialisierungen in der entsprechenden Tabelle als Attribute vorhanden sein.

Mapping auf XML

Über XML werden Daten in einer Baumstruktur gespeichert. Dazu wird für die Repräsentation von PODSL-Biodiv eine XSD mit dem Wurzelelement 'PODSL-Biodiv-Schema' gebildet. In diesem werden alle Konzepte von PODSL-Biodiv als komplexe Typen in XSD spezifiziert. Somit stellen die Instanzen von 'ProcessExecutionDocument' und EntityType separate komplexe Typen dar. Diese enthalten alle Attribute

der jeweiligen Konzepte in PODSL. Die Vererbung von Konzepten wird dabei analog zum Mapping auf Datenbanken vorgenommen. Die Identifier aus PODSL werden dazu verwendet, um Konzepte und Datensätze zu identifizieren.

Eine wesentliche Aufgabe der XML-Repräsentation von PODSL-Biodiv ist die Übertragung von Daten auf der M_0-Ebene innerhalb einer Infrastruktur. Dazu müssen verschiedene Instanzen der M_1-Ebene von PODSL-Biodiv in einer gemeinsamen Struktur übertragen werden können. Diese werden dazu analog zur Datenübertragung mit DwC in einem Container zusammengefasst, welcher als 'DataSet' bezeichnet wird. Ein 'DataSet' kann dabei Instanzen von beliebigen Entitäten von PODSL-Biodiv enthalten. Da 'ProcessExecutionDocuments' und 'ProcessExecutionSequences' nach der Spezifikation von PODSL auf M_2-Ebene ebenfalls 'EntityTypes' sind, können diese auch innerhalb eines DataSets übertragen werden.

Mapping auf Ontologien

Ein zentrales Element von Ontologien ist die Identifikation von Modellelementen über URNs und die Zerlegung der Datenstruktur in Begriffe und Literale. Da PODSL-Biodiv über Identifier die Referenzierung von Modellelementen bis auf Attributebene unterstützt, kann PODSL-Biodiv vollständig als Ontologie repräsentiert werden. Dazu wird für jedes Konzept aus PODSL-Biodiv in der Repräsentation als Ontologie ein separater Begriff gebildet. Da der Identifier von PODSL innerhalb einer Infrastruktur eindeutig sind, können die PODSL-Identifier in eine URN überführt und zur Identifikation in der Ontologie verwendet werden. Ableitungshierarchien werden hierbei analog zu dem ER-Mapping unterstützt.

Die Grundstruktur eine Ontologie in RDF besteht aus der Zerlegung eines Sachverhalts in Tripel des Typs 'Subjekt' - 'Prädikat' - 'Objekt', wobei diese jeweils eindeutig über eine URN identifiziert werden. Damit wird ein 'EntityType' aus PODSL in der Repräsentation als Ontologie über eine Menge dieser Tripel beschrieben. Dabei wird in einem solchen RDF-Statement ein 'EntityType' als Subjekt referenziert und eine spezifische Eigenschaft dieses 'EntityTypes' als Objekt angegeben. Die Funktion dieses Objekts bei der Beschreibung des 'EntityTypes' wird durch das Prädikat angegeben. Auf diese Weise lassen sich sowohl Beziehungen zu anderen Entitäten oder Attribute einer Entität spezifizieren. Da Vererbung und Mehrfachvererbung in Ontologien unterstützt wird, kann das Vererbungsprinzip von PODSL in Ontologien direkt umgesetzt werden.

Mapping auf Klassen der objektorientierten Programmierung

Für die objektorientierte Programmierung ist die Strukturierung eines Datenmodells in Form von Klassen erforderlich. Dabei dienen die Entitäten von PODSL-Biodiv als Grundlage für die Erstellung von Klassen. Vererbung ist ein zentrales Element der objektorientierten Programmierung. Mehrfachvererbung wird hingegen in den meisten objektorientierten Programmiersprachen (wie z.B. Java oder C#) nicht unterstützt. In diesen Programmiersprachen existiert das Konzept von Schnittstellen, welche auch als 'Interfaces' bezeichnet werden. Schnittstellen geben dabei für eine Klasse eine Struktur vor, ohne eine konkrete Implementierung vorzuschreiben. Dadurch kann eine Klasse mehrere Interfaces implementieren [249]. Dabei wird sich in den folgenden Ausführungen, sofern nicht explizit anders erwähnt auf die Programmiersprache C# bezogen.

Die Strukturierung der Datenstruktur von PODSL-Biodiv in C# erfolgt dabei über 'Properties', welche in Interfaces spezifiziert werden[11]. Methoden und Attribute werden zur Spezifikation von PODSL nicht benötigt. Dabei wird für jede Entität und jede Spezialisierung einer Entität aus PODSL ein Interface angelegt. Da eine Klasse mehrere Interfaces implementieren kann, kann die Vererbungshierarchie von PODSL-Biodiv über Interfaces abgebildet werden. Zur Anwendung in Programmen werden auf Basis dieser Interfaces Klassen erzeugt, welche diese Interfaces implementieren. Beziehungen werden dabei auch über Properties abgebildet. Dazu wird für die Repräsentation einer 1:1-Beziehung ein Property vom Typ der referenzierten Entität gebildet. 1:n-Beziehungen werden über eine Collection des entsprechenden Datentyps wie 'List<EntityType>' erzeugt, wobei für 'EntityType' die referenzierte Entität einzusetzen ist. Da 'ProcessExecutionDocuments' und 'ProcessExecutionSequences' auf der M_2-Ebene von PODSL von 'EntityType' abgeleitet werden, werden diese in der objektorientierten Programmierung ebenfalls über 'Interfaces' und Klassen repräsentiert.

5.5.3 PODSL-Biodiv in der Datenübertragung

Ein wesentliche Aufgabe eines Datenstandards in der Biodiversitätsinformatik ist die Anwendung des Datenstandards zum Datenaustausch in Infrastrukturen. Diese Thematik wird speziell für BDEI in Kapitel 8 ausführlich besprochen. Aufgrund der enormen Bedeutung soll der Datenaustausch mit PODSL-Biodiv an dieser Stelle kurz skizziert werden. Durch das Metamodell von PODSL-Biodiv wird eine flexible Struktur für Daten in der Biodiversitätsforschung spezifiziert. Allerdings kann diese

[11] Das korrelierende Konzept in Java ist die Spezifikation über 'Getter' und 'Setter'.

Struktur nicht direkt zur Datenübertragung verwendet werden, sondern dient dafür als logische Grundlage. Für die Datenübertragung müssen Daten in serieller Form existieren, welche zwischen den einzelnen Datenspeichern ausgetauscht werden.

Der Datenaustausch in der Biodiversitätsinformatik erfolgt überwiegend in XML. Deswegen soll für den Austausch von Daten in PODSL-Biodiv auch die in Abschnitt 5.5.2 vorgestellte Methode zur Datenübertragung nach XML genutzt werden. Ein Vorteil von der Verwendung von XML als Datenaustauschformat ist, dass die Nutzung von XML in Webservices (siehe Kapitel 6) weit verbreitet ist. Datenspeicher innerhalb einer Infrastruktur wie BDEI sind dabei im Allgemeinen heterogener Natur und verfügen über jeweils verschiedene interne Schemata und Speichertechnologien. Diese internen Schemata werden über ein Mapping mit einem Wrapper an das lokale Schema angebunden. Für die Kommunikation zwischen den verschiedenen Datenspeichern werden die Daten für den Austausch in die XML-Repräsentation von PODSL-Biodiv übertragen. Damit dient die Repräsentation von PODSL-Biodiv in XML als Austauschformat für den Datenexport und -import eines Datenspeichers.

5.6 Mapping von PODSL-Biodiv auf DwC und andere Datenstandards

Eine wesentliche Eigenschaft einer Modellierungssprache ist nach [32] die Abbildung auf andere Sprachen. In der Biodiversitätsinformatik ist insbesondere die Abbildung nach DwC und ABCD von Bedeutung, da diese zentrale Standards der Biodiversitätsinformatik darstellen. Aufgrund der Zielsetzung von PODSL-Biodiv, die Mächtigkeit von DwC vollständig abzubilden, wird sich im Folgenden auf das Mapping von PODSL-Biodiv nach DwC konzentriert. Dabei wird indirekt eine Abbildung von PODSL-Biodiv nach ABCD erstellt, da mit [15] eine Abbildung zwischen ABCD und DwC besteht. Darüber hinaus umfasst PODSL-Biodiv aufgrund der prozessorientierten Struktur und der Erfassung von Messungen als Prozessen OBOE vollständig.

Bei dem Mapping nach DwC wurde eine Abbildung auf den allgemeinen DwC mit Ausnahme des Konzepts 'GeologicalContext' vorgenommen [12]. Ein explizites Mapping auf den Simple-DwC ist nicht erforderlich, da dieser lediglich eine Teilmenge der Konzepte des allgemeinen DwC enthält. PODSL-Biodiv ist bis auf den geologischen Kontext mächtiger als DwC und kann somit in der Domäne der Biodiversitätsforschung mehr Sachverhalte erfassen als DwC. Dementsprechend muss sich die

[12] Dieses kann leicht nachgeholt werden, sobald PODSL-Biodiv geologische Zusammenhänge berücksichtigt.

Abbildung von PODSL-Biodiv auf DwC auf Elemente beschränken, welche in DwC existieren. Grundlage für die Abbildung ist die Struktur von DwC in Basisklassen. Dazu wurde den Basisklassen in DwC im Mapping in Anhang B die entsprechenden Modellierungselemente von PODSL-Biodiv gegenübergestellt. Das vollständige Mapping ist in Anhang B aufgelistet.

Im Mapping zwischen PODSL-Biodiv und DwC ist zu erkennen, dass in DwC nicht sauber zwischen Entitäten und der Erfassung einer Prozessausführung getrennt wird. Darüber hinaus erlaubt DwC die Belegung eines Attributs mit sehr unterschiedlichen Werten, so dass die Vergleichbarkeit der Daten in DwC nicht gewährleistet ist. So umfasst z.B. die Basisklasse 'Location' aus DwC genauso die allgemeine Beschreibung eines Ortes wie auch den Prozess der Georeferenzierung mit der Bestimmung von Koordinaten durch einen Verantwortlichen. Diese Daten sind in PODSL-Biodiv strikt voneinander getrennt. Ein weiteres Beispiel für die vage Spezifikation von DwC findet sich in der Basisklasse 'Occurence'. Hier können z.B. über den 'sex', wie in [263] beschrieben, nicht nur das Geschlecht eines Beobachtungsobjekts sondern auch Zählungen von Individuen mit Geschlechtsangabe in Freitextform eingegeben werden. Dies macht eine strukturierte Auswertung der Daten unmöglich. In PODSL-Biodiv ist dieses Problem dahingehend gelöst, dass dieses Feld nicht einer Entität zugeordnet ist, sondern als ein Prozess zur Messung des Geschlechts eines biologischen Objektes aufgefasst wird. Dadurch ist klar, auf welches biologische Objekt sich diese Messung bezieht. Da für Messergebnisse ein kontrolliertes Vokabular besteht, in welchem die Schlagworte über Identifier eindeutig referenzierbar sind, kann in PODSL-Biodiv genau spezifiziert werden, was gemessen wurde. Die Vergleichbarkeit von verschiedenen Messungen ist damit in PODSL-Biodiv gewährleistet.

5.7 Evaluation

Vor der Entwicklung von PODSL-Biodiv wurden existierende Datenstandards aus der Biodiversitätsinformatik evaluiert und es konnte gezeigt werden, dass diese die Anforderungen der Domäne der Biodiversitätsinformatik nur unzureichend erfüllen. Mit PODSL-Biodiv wurde in diesem Kapitel ein neuer Datenstandard für dieses Domäne eingeführt, der die identifizierten Mängel beseitigen soll. Um diese zu verifizieren soll in diesem Abschnitte PODSL-Biodiv nach den Kriterien aus Abschnitt 4.3 evaluiert werden.

Vollständigkeit: In PODSL-Biodiv werden die Anwendungsfälle UC1-UC5 vollständig erfasst. Zusätzlich umfasst PODSL-Biodiv (bis auf die Basisklasse 'GeologicalContext') DwC vollständig. Das Kriterium der Vollständigkeit ist damit erfüllt.

Flexibilität: In Abschnitt 5.5.1 wird ein Prozess zur Erweiterung von PODSL-Biodiv vorgestellt, welcher die individuelle Anpassung von PODSL-Biodiv ermöglicht und gleichzeitig die Kompatibilität mit PODSL-Biodiv über Vererbung erhält. Der Prozess ist in Abbildung 5.20 beschrieben und auch ohne die Konsultation eines Expertengremiums ausführbar. Das Kriterium der Flexibilität ist damit erfüllt.

Data Provenance: Die Unterstützung von Data Provenance ist in PODSL über das Konzept der 'ProvenanceTable', 'BaseEntity' und 'BaseProcessExecutionDocument' realisiert. Von 'BaseEntity' müssen alle anderen Entitäten zumindest mittelbar abgeleitet werden, so dass alle Entitäten in PODSL über alle Attribute der 'BaseEntity' verfügen müssen. Gleich gilt für das 'BaseProcessExecutionDocument' im Bezug auf 'ProcessExecutionDocuments'. Sowohl in 'BaseEntity' als auch im 'BaseProcessExecutionDocument' ist die Speicherung des Erstellungszeitpunkts sowie des Erstellers explizit vorgesehen. DP1 und DP2 sind damit erfüllt. Die Speicherung von Versionen von Datensätzen und der Bezug zum Original wird in der 'ProvenanceTable' gespeichert. Diese enthält auch Verweise auf Prozesse, in denen Datensätze konvertiert werden. Damit können Veränderungen von Datensätzen verfolgt werden. DP3 und DP4 sind somit erfüllt. Das Kriterium 'Data Provenance' ist damit insgesamt erfüllt.

Referenzierbarkeit: Durch die Spezifikation aller Konzepte in der M_2-Ebene wird in PODSL gewährleistet, dass jedes Konzept in PODSL und in allen domänenspezifischen Erweiterungen über eine eindeutige Referenz verfügen muss. Dies gilt somit auch für alle Konzepte aus PODSL-Biodiv. Das Kriterium der Referenzierbarkeit ist damit erfüllt.

Redundanzfreiheit: Konzepte sind in PODSL-Biodiv immer genau einmal angelegt. Die Wiederholung von Konzepten tritt in anderen Datenstandards häufig durch Spezialisierungen von Konzepten auf, in denen Attribute redundant erfasst werden. In PODSL-Biodiv wird diese durch die Spezialisierung von existierenden Konzepten durch Vererbung vermieden. Das Kriterium der Redundanzfreiheit ist damit erfüllt.

Damit erfüllt PODSL-Biodiv alle Kriterien aus Abschnitt 4.3. Darüber hinaus basiert PODSL-Biodiv zu einem großen Teil auf der Übertragung der Prozessmodelle zu den Anwendungsfällen UC1-UC5 in Datenstrukturen. Da diese Prozessmodelle die Visualisierung wichtigsten Prozesse einer komplexen Anwendungsdomäne darstellen, wird über diese auch die Verständlichkeit von PODSL-Biodiv erhöht. Dies ist ein Vorteil gegenüber etablierten Datenstandards, da diese lediglich auf eine abstrakten Spezifikation beruhen, die nicht visualisiert werden kann.

5.8 Fazit

Mit PODSL-Biodiv wird ein erweiterbares Datenmodell für die Biodiversitätsinformatik spezifiziert, welches im Bezug auf die Vollständigkeit den Anforderungen dieser Domäne in umfassender Weise gerecht wird. Dabei wird PODSL-Biodiv als eine logische Datenstruktur zu verstehen, aus welcher Abbildungen in andere Technologien und Datenschemata leicht möglich sind. Grundlage hierfür ist die Metastruktur von PODSL und die in PODSL-Biodiv erfassten Prozesse. Damit liegt das Einsatzgebiet von PODSL-Biodiv als moderierendes Schema bei der Datenübertragung. Daten können von einem Datenstandard in die logische Struktur von PODSL-Biodiv übertragen werden. Genauso können Daten im Schema von PODSL-Biodiv leicht in andere Datenstandards übertragen werden. Somit bietet sich PODSL-Biodiv als Zwischenschema für die Datenübertragung an.

PODSL-Biodiv ermöglicht darüber hinaus die Konvertierung der Daten in verschiedene Technologien von Datenspeichern wie XML, Ontologien oder Datenbanken ermöglicht. Dies ist notwendig, da die Instanziierung der M_0-Ebene im OMME sehr umständlich ist und die Daten in Netzwerken in serieller Form übertragen werden müssen. Dazu ist die Konvertierung von Daten aus PODSL-Biodiv in eine XML-Struktur besonders gut geeignet. Der Prozess der flexiblen Erweiterung des Datenmodells bietet außerdem Vorteile gegenüber die Erweiterung eines Datenstandards durch Expertengremien. Diese können neue Anforderungen an ein Datenmodell nicht zeitnah umsetzen und die Entscheidungen sind häufig nicht transparent. Durch die Möglichkeit der Erstellung von proprietären Erweiterungen, kann PODSL-Biodiv schnell an neue Anforderungen und auch individuelle Anforderungen angepasst werden. Die Kompatibilität der Erweiterungen zu PODSL-Biodiv wird dabei durch das Prinzip der Vererbung gewährleistet.

Da PODSL-Biodiv das Kriterium der Vollständigkeit für die Domäne der Biodiversitätsinformatik erfüllt, ist PODSL-Biodiv für die Strukturierung der Daten in der Biodiversitätsforschung sehr gut geeignet. Über das Prinzip der Vererbung wird in PODSL-Biodiv eine wichtige Anforderung für den Einsatz in Infrastrukturen erfüllt. Deshalb dient PODSL-Biodiv als Datenstandard für die Datenspeicherung und -übertragung in der Infrastruktur BDEI für die Biodiversitätsinformatik (siehe Kapitel 8). Folglich erfüllt PODSL-Biodiv bereits heute alle technischen Voraussetzungen um einen Datenstandard wie DwC oder ABCD abzulösen. Dies wird aber vermutlich aus organisatorischen Gründen nicht zeitnah geschehen, da enorme Datenmengen in den etablierten Datenstandard wie DwC oder ABCD gespeichert sind und diese Standards auch von eine Vielzahl von Anwendungen unterstützt werden. Dieses

Anwendungen sind häufig nicht so programmiert, dass das Datenmodell leicht ausgetauscht werden könnte. Die Einführung eines neuen Datenstandards wie PODSL-Biodiv erfordert deshalb einen enormen Programmieraufwand zur Anpassung aller etablierten Anwendungsprogramme. Dies kann nicht ohne die Unterstützung der Entscheidungsträger in der Biodiversitätsinformatik erfolgen, die häufig eigene Lösungen propagieren. Für die Einführung von PODSL-Biodiv auf breiter Basis ist deshalb primär Überzeugungsarbeit zu leisten.

Kapitel 6

Daten- und Informationsintegration

Bevor in Kapitel 7 ein Framework zur Evaluation von Infrastrukturen spezifiziert werden kann, sollen zunächst die Grundlagen der Daten- und Informationsintegration eingeführt werden. Dazu werden in Abschnitt 6.1 grundlegende Definitionen, Begriffe und Arten der Datenintegration vorgestellt. In Abschnitt (6.2) werden Techniken zum Datenaustausch in verteilten Systemen besprochen, welche die Grundlage zum Datenaustausch über eine Infrastruktur bilden. Ein wesentliches Element in diesen Infrastrukturen sind Webservices und verteilte Datenbanken, auf die explizit eingegangen wird.

Anschließend werden existierende Ansätze zur Integration von heterogenen Datenquellen aus dem verschiedenen Bereichen in Abschnitt 6.3 vorgestellt. Ansätze im kommerziellen Bereich werden z.B. von Oracle, SAP, IBM und Informatica angeboten. Im wissenschaftlichen Bereich sind Taverna, Kepler und DaltON von besonderem Interesse. Besonderes Augenmerk wird hierbei auf das DaltON-Framework gelegt (Abschnitt 6.3). Die Besonderheit von DaltON ist die Kombination eines prozessbasierten Ansatzes mit der Möglichkeit der semantischen Integration von Daten. DaltON ist das zentrale Element der Dissertation [201]. Die Gedanken dieser Arbeit werden in Kapitel 8 aufgegriffen und zu einem eigenen Framework zum Datenaustausch in der Biodiversitätsinformatik weiterentwickelt.

6.1 Grundlagen

Die Begriffe **Daten- und Informationsintegration** werden synonym verwendet [193]. Dabei scheint im deutschen Sprachraum der Begriff der Informationsintegrati-

on geläufiger zu sein, wohingegen im Englischen der Begriff 'Data Integration' vermutlich von größerer Bedeutung ist. Ausgangspunkt für die synonyme Verwendung von 'Datenintegration' und 'Informationsintegration' sind die Begriffe 'Daten' und 'Information'. Die Unterscheidung dieser Begriffe im Deutschen geht auf die Repräsentation von Daten in Rechnern zurück. Informationen werden im Rechner durch Nullen und Einen repräsentiert [89]. Die so repräsentierten Informationen werden als Daten bezeichnet [89]. Die Repräsentation der Daten muss dabei so gewählt sein, dass die Informationen durch den Vorgang der Abstraktion wiederhergestellt werden können [89]. Bei der Unterscheidung der Begriffe 'Datenintegration' und 'Informationsintegration' wird dementsprechend darauf abgezielt, auf welcher Ebene die Integration stattfindet. Da die Repräsentation in Daten dabei in der Praxis von geringerem Interesse ist als die Integration der tatsächlichen Information, wird im Folgendem im Einklang mit der üblichen Sprachregelung im Deutschen der Begriff Informationsintegration verwendet. Dieser wird in [93, 138, 139, 244] sinngemäß wie folgt definiert:

Definition 6.1: Informationsintegration

Informations- oder Datenintegration ist das Problem der Kombination von Daten, die in unterschiedlichen Quellen gespeichert sind. Ziel der Informationsintegration ist es dabei, dem Anwender eine einheitliche Sicht auf die Datenquellen zur Verfügung zu stellen.

Da die Ursprünge der Informationsintegration im Datenbankbereich liegen, ist es naheliegend den Begriff 'Sicht' im datenbanktechnischen Sinne zu verstehen. Das primäre Ziel der Informationsintegration ist es, einen einheitlichen Zugriff auf autonome und heterogen Quellen zu erhalten [93]. Dabei liegt der Fokus im Allgemeinen auf der Anfragestellung, wobei auch Änderungsmechanismen von Interesse sind [93]. Die Informationsintegration erfolgt über Systeme, die als'integrierte' oder 'integrierende Informationssysteme' bezeichnet werden [139] und wie folgt definiert sind:

Definition 6.2: Integriertes Informationssystem nach Leser [139]

Ein Programm, welches den Zugriff auf Daten aus verschiedenen Datenquellen ermöglicht, wird als ein 'integriertes Informationssystem' bezeichnet.

Der Begriff 'integrierendes Informationssystem' wird dabei synonym zu dem Begriff 'integriertes Informationssystem' verwendet [139]. Im Wortsinne ist der Unterschied zwischen diesen beiden Begriffen der, dass ein integrierendes System erst beim Zugriff integriert wohingegen in einem integrierten System die Daten zum Zeitpunkt

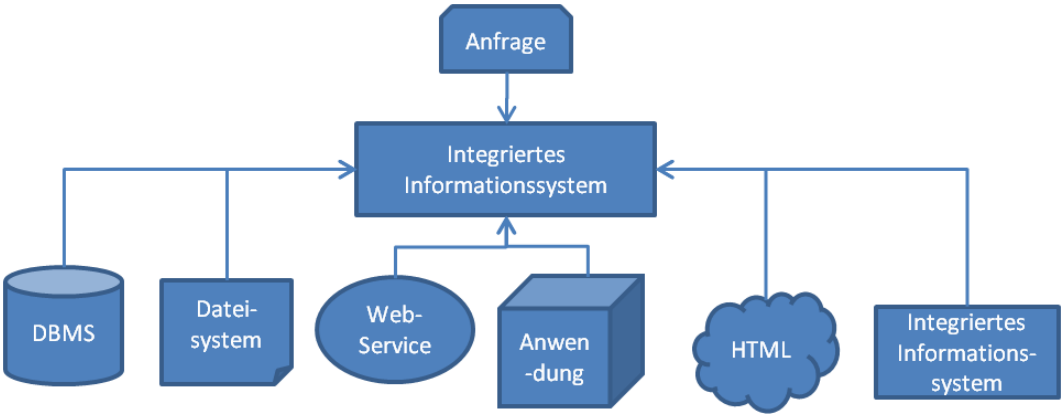

Abbildung 6.1: Informationsintegration über heterogene Datenquellen [139]

des Zugriffs bereits in integrierter Form vorliegen [139]. Dieser Unterschied ist aber für die Praxis nicht relevant.

Eine **Datenquelle** ist ein beliebig aufgebauter Informationsspeicher, dessen Daten in das System integriert werden sollen [139]. Die wichtigste Klasse an Informationsspeichern ist die Klasse der Datenbanken [139]. Dabei beschränkt sich Informationsintegration nicht ausschließlich auf Datenbanken, sondern bildet eine einheitliche Sicht für Datenquellen über verschiedenen Technologien hinweg (siehe Abbildung 6.1). Zu diesen können wiederum Integrationssysteme gehören.

Das Problem der Informationsintegration ist nicht neu und wird in [95, 139] als eines der beständigsten Probleme der Datenbankforschung beschrieben, welches sowohl in der Industrie als auch in der Forschung von immenser Bedeutung ist. Dabei wird nach [93, 139] zwischen **materialisierter** ([93] verwendet den Begriff 'Warehousing') und **virtueller Integration** unterschieden. Bei der 'materialisierten Integration' werden Daten zur Integration in einen gemeinsamen Speicher kopiert und so eine einheitliche Sicht auf die Daten erzeugt. Das wichtigste Beispiel für die materialisierte Integration sind Data Warehouses. In Data Warehouses werden Daten aus verschiedenen Quellen über den sogenannten ETL-Prozess (Extract, Transform, Load) in ein einheitliches Schema überführt [11]. Im Gegensatz dazu verbleiben in der 'virtuellen Integration' die Daten in ihrer ursprünglichen Datenquelle und es wird für diese verschiedenen Datenquellen ein globales Schema definiert [139], an welches Anfragen gestellt werden. Diese Anfragen werden zum Zeitpunkt der Ausführung in Anfragen an die angeschlossenen Datenquellen übersetzt. Ein Beispiel für virtuelle Integration sind 'föderierte Datenbanken' [139].

Der Umstand, welcher die Integration von Daten überhaupt erst notwendig macht ist, dass Daten in **verteilten und heterogenen Datenquellen** gespeichert sind.

Bei der Verteilung von Daten wird von zwei unterschiedlichen Arten ausgegangen, der physischen und der logischen Verteilung [139]. Nach [139] sind Daten physisch verteilt, wenn diese auf physisch getrennten Systemen gespeichert werden, wohingegen eine logische Verteilung vorliegt, wenn es für ein Datum mehrere mögliche Orte zu seiner Speicherung (auch innerhalb einer Datenbank) gibt. Ein weiteres Problem der Informationsintegration ist die **Heterogenität** der Daten, welches in Relation zur Verteilung der Daten als ein orthogonales Konzept betrachtet werden kann [139]. [139] unterscheidet hier zwischen verschiedenen Formen der Heterogenität mit folgenden zugehörigen Problembereichen:

- Technische Heterogenität: Probleme in der Realisierung des technischen Zugriffs
- Syntaktische Heterogenität: Probleme bei der Darstellung von Informationen
- Datenmodellheterogenität: Probleme in den zur Präsentation verwendeten Datenmodellen
- Strukturelle Heterogenität: Unterschiede in der strukturellen Repräsentation von Information
- Schematische Heterogenität: Spezialfall der strukturellen Heterogenität, wobei die Unterschiede in den verwendeten Datenmodellelementen liegen
- Semantische Heterogenität: Probleme mit der Bedeutung der verwendeten Begriffe und Konzepte

Eine Möglichkeit um Heterogenität zu begrenzen ist Standardisierung [139]. Standards sind im Allgemeinen domänenspezifisch. Dadurch entsteht das Problem, dass die Autonomie der Informationssystem in der Anwendungsdomäne eingeschränkt wird [139]. Dafür wird der Informationsaustausch erleichtert [139]. In der Biodiversitätsinformatik werden Daten in heterogenen Datenspeichern über das BioCase-Protokoll in das standardisierte ABCD-Format zum Datenaustausch übertragen. Die Datenquellen können dabei intern alle Arten der Heterogenität aufweisen, werden aber an eine standardisierte Architektur gebunden, die es ermöglicht die für GBIF relevanten Informationen aus einer Datenquelle zu entnehmen (siehe Abschnitt 7.4.1).

Eine weitere wichtige Eigenschaft von integrierten Systemen ist der **transparente Zugriff**. Unter transparenten Zugriff ist zu verstehen, dass die interne Informationsstruktur eines integrierten Systems für den Anwender unsichtbar bleibt und deshalb keine Kenntnis über diese besitzen muss [139]. Transparenz kann dabei auf

den verschiedensten Ebenen und in unterschiedlichsten Formen vorhanden sein, z.B. dadurch, dass der Anwender keine Kenntnis über die Datenquellen eines integrierten Systems benötigt, da diese Informationen intern vom integrierten System verwaltet werden [139].

Die Freiheit einer Datenquelle, unabhängig über die von ihr verwalteten Daten, ihre Struktur, und die Zugriffsmöglichkeiten auf diese zu entscheiden [139] wird als **Autonomie** bezeichnet. In [139] wird zwischen vier verschiedenen Arten von Autonomie unterschieden:

- Designautonomie: Eine Datenquelle besitzt Designautonomie, wenn sie frei entscheiden kann, in welcher Art und Weise sie ihre Daten zur Verfügung stellt.

- Schnittstellenautonomie: Schnittstellenautonomie bezeichnet die Freiheit jeder Datenquelle, selber zu bestimmen mit welchen technischen Verfahren auf die von ihr verwalteten Daten zugegriffen werden kann.

- Zugriffsautonomie: Zugriffsautonomie ist gegeben, wenn eine Datenquelle frei entscheiden kann, wer auf welche der von ihr verwalteten Daten zugreifen kann.

- Juristische Autonomie: Mit juristischer Autonomie wird das Recht einer Datenquelle bezeichnet, die Integration ihrer Daten in ein Informationsintegrationssystem zu verbieten.

Die Autonomie von Datenquellen stellt ein integriertes System vor besondere Herausforderungen, da nicht vorausgesetzt werden, dass ein integriertes System zu jeder Zeit vollen Zugriff auf die Daten einer Quelle hat [93]. Die Datenquellen können ihr Datenformat und ihre Zugriffsmöglichkeiten jederzeit ändern [93]. Dies ist Ursache von Heterogenitätsproblemen. Aufgrund des allgemeinen menschliches Strebens, sich größtmögliche Autonomie zu sichern [139], tritt in einem System mit vielen Teilnehmern das Heterogenitätsproblem mit hoher Wahrscheinlichkeit auf.

Die Hauptkomponenten eines virtuellen integrierten Informationssystem I sind das **globale Schema** G, die **Datenquellen** (sources) S, und die **Abbildungen** (mappings) M zwischen den Datenquellen mit dem globalen Schema [138], so dass ein virtuelles integriertes System durch ein Tripel $I =< G, S, M >$ dargestellt werden kann. Somit ist das Finden von entsprechenden Abbildungen M und die Optimierung von Anfragen an diese ein entscheidender Schritt bei der Anwendung von virtuellen integrierten Informationssystemen. Um Informationen aus den Datenquellen S zu bekommen muss damit eine Anfrage in der jeweiligen Sprache der Datenquelle formuliert werden. Dabei wird zwischen den folgenden, grundlegenden Strategien unterschieden:

- 'Global as view' (GAV): Das globale Schema G wird in Begriffen der Quellen S ausgedrückt [138]. Dabei werden im Datenbankbereich im globalen Schema Sichten erzeugt, welche konkrete Anfragen in den Quellen darstellen.

- 'Local as view' (LAV): Das globale Schema G ist von den Quellen S unabhängig und die Abbildungen zwischen G und S werden dadurch hergestellt, dass jede Quelle S als eine Sicht auf dem globalen Schema G definiert wird [138]. Das globale Schema bleibt dadurch beim Hinzufügen einer Quelle unverändert, da die Quellen S mit Hilfe von Sichten an das globale Schema angepasst werden.

Die Unterschiede und Vorteile von LAV und GAV werden in [138, 139] ausführlich diskutiert. Ein wesentlicher Nachteil bei der Verwendung von LAV ist, dass die Planung von Anfragen komplizierter als bei GAV [139] ist. Darüber hinaus ist LAV in Situationen besser, in denen das globale Schema allgemeiner ist als das lokale Schema, da mit GAV-Regeln keine entsprechenden Bedingungen formuliert werden können [139]. Ist umgekehrt das lokale Schema allgemeiner als das globale Schema, kann dies nicht mit LAV-Regeln dargestellt werden und der GAV-Ansatz ist überlegen. Um die Vorteile der beiden Ansätze zu kombinieren, wurde mit 'Global-Local as view' (GLAV) eine Kombination von GAV und LAV erschaffen, in welchem die Anfrageplanung wie in einer LAV und die Anfrageausführung wie in einer GAV ausgeführt wird [139]. Der interessierte Leser wird für eine genauere Beschreibung auf [94, 138, 139] verwiesen.

Eine besondere Methode der Informationsintegration stellt die **Semantische Integration** dar, in welcher insbesondere Probleme der semantischen Heterogenität gelöst werden sollen. In diesem Ansatz wird die Informationsintegration über die Anwendung von Ontologien realisiert (vgl. Abschnitt 3.1.2) [139, 217]). Eine übliche Methode ist es hierbei, eine zentralen Ontologie domänenspezifisch zu erweitern

[182]. Semantische Integration hat sich in den letzten Jahren stark weiterentwickelt. Der Leser sei für einen historischen Überblick auf [182] und für aktuelle Ansätze auf [217] verwiesen. Da DaltON auf semantischer Integration beruht wird in Abschnitt 6.3.3 auf die Anwendung der semantischen Integration in DaltON eingegangen.

6.2 Technische Grundlagen des Datenaustauschs

Ein integriertes Informationssystem benötigt eine **technische Infrastruktur**, über welche dieses mit seinen Quellen kommuniziert. Dabei werden in den Abschnitten 6.2.1 - 6.2.3 Technologien vorgestellt, welche den Zugriff auf verteilte Datenquellen ermöglichen. Die Grundlage des Datenaustauschs beruht damit auf einer IT-Infrastruktur, welche nach [194] aus Hardware, Software sowie baulichen Einrichtungen für den Betrieb von Software besteht. Eine umfassende Diskussion des Begriffs 'Infrastruktur' findet sich in Abschnitt 7.1. Ziel der technischen Infrastruktur ist die Überwindung von technischer Heterogenität [139]. Die Aufgaben einer solchen Infrastruktur sind nach [139] wie folgt:

- Finden von Quellen entweder über einen eindeutigen Bezeichner (IP oder URI) oder über die Suche in Verzeichnissen
- Senden von Anfragen der Integrationsschicht an die Quellen
- Antworten der Quellen an die Integrationsschicht

Um diese Aufgaben zu erfüllen, können verschiedene technische Ansätze gewählt werden. Nach [139] sind die folgenden Ansätze am wichtigsten:

- Verteilte Datenbanken (Abschnitt 6.2.1)
- Web-Services (Abschnitt 6.2.2)
- (Objektorientierte) Middleware (Abschnitt 6.2.3)

Diese Ansätze werden im Folgenden kurz vorgestellt.

6.2.1 Verteilte Datenbanken

Wenn es sich bei den Datenquellen ausschließlich um relationale Datenbanken handelt, ist es möglich eine Infrastruktur zum Datenaustausch über verteilte Datenbanken zu erschaffen [139]. [139] unterscheidet vier verschiedene Typen von verteilten Datenbanken, deren Grundaufgabe immer der Zugriff auf Daten in einer externen Datenbank ist:

- Homogen: Homogene verteilte Datenbanken basieren auf einem 'relationalen Datenbank Management Systems' (RDBMS) eines einzigen Herstellers, deren Instanzen physisch verteilt sind. Die Kommunikation beruht hierbei häufig auf proprietäre Protokolle.
- Heterogen: Alle Quellen des integrierten Systems sind RDBMS – aber von unterschiedlichen Herstellern. Hersteller mit ausreichender Verbreitung bieten dabei produktspezifische Datenbank-Gateways als Schnittstelle zu Systemen anderer Hersteller an.
- Heterogen/generisch: Wird mit einem RDBMS als Quelle gearbeitet, für das kein Gateway existiert, kann man mit diesen häufig über generische Schnittstellen wie die 'Open Database Connectivity' (ODBC), 'Java Database Connectivity' (JDBC) oder 'Object Linking and Embedding Database' (OLEDB) zugreifen.
- Nicht relational: Hiermit werden Systeme bezeichnet, die über Wrapper auf Daten in nicht relationalen Datenbanken (Datenspeicher im XML-Format, NoSQL-Datenbanken) zugreifen.

Der einfachste Fall von verteilten Datenbanken ist sicherlich der homogene Fall. Dieser wird aber in der Praxis bei autonomen Datenquellen eher selten anzutreffen sein, da jeder Betreiber einer Datenquelle eigene Anforderungen an sein System stellen kann und es so zwangsläufig zur Auswahl unterschiedlicher Anbieter von RDBMS kommt. Auch eine durchgängigen Auswahl von Anbietern, die über Datenbank-Gateways kommunizieren, kann garantiert werden. Der Zugriff auf generische Schnittstellen kann im Allgemeinen vorausgesetzt werden. Dieser verfügt über den Nachteil, dass lokale und entfernte Tabellen nicht gleichzeitig angesprochen werden können [139]. Wrapper ermöglichen im Kontext der verteilten Datenbanken an eine nichtrelationale Datenquelle eine SQL-Anfrage wie an eine lokale Tabelle zu richten [139]. Bei dem Einsatz von Wrappern kann dies mit Anfragen an eine relationale Datenquellen kombiniert werden.

6.2.2 Web-Services

Mit Web-Services wird ein systematisches und erweiterbares Framework zur Kommunikation zwischen Anwendungen zur Verfügung gestellt [39]. Über diese ist auch der Austausch von Daten möglich. Dabei bauen Webservices auf existierenden Web-Protokollen auf und basieren auf offenen XML Standards [39]. Eine wichtige Eigenschaft von Webservices ist die Unabhängigkeit von einer bestimmten Plattform

[39]. Das W3C definiert in der Spezifikation der Architektur eines Web-Service [254] diesen wie folgt:

Definition 6.3: Web-Service nach W3C [254]

„A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.“

Demnach ist ein Web-Service ein Softwaresystem, um die vollständig kompatible Kommunikation von Maschine zu Maschine zu unterstützen. Er muss dabei über eine maschinenlesbare Schnittstelle in WSDL verfügen und mit anderen Systemen über SOAP kommunizieren. WSDL und SOAP werden folglich per Definition durch das W3C vorgeschrieben. Eine weitere in diesem Zusammenhang häufig erwähnte Technologie ist UDDI. Über diese Technologien wird in [39] eine umfassende Einführung gegeben, welche an dieser Stelle kurz nach [39] vorgestellt werden:

- Simple Object Access Protocol (SOAP): SOAP ermöglicht die Kommunikation zwischen Web-Services. SOAP basiert auf einer XSD, welche als zwingende Element einen Header und einen Body vorschreibt. Die eigentlichen Informationen werden damit über einen Umschlag (envelope) gekapselt. Über diesen Grundaufbau hinaus verfügt SOAP über die Möglichkeit konsumierenden Services mitzuteilen, wie diese SOAP-Nachrichten verabeiten sollen.

- Web Service Description Language (WSDL): WSDL stellt eine maschinenlesbare Sprache zur Beschreibung von Web-Services dar. In dieser werden Web-Services in XML als Mengen von Kommunikationsendpunkten beschrieben, die Nachrichten austauschen können.

- Universal Description, Discovery and Integration (UDDI): UDDI ist ein Verzeichnisdienst zum Registrieren und Auffinden von Web-Services.

SOAP wird neben der Verwendung zum Senden von Nachrichten auch im Kontext von 'Remote Procedure Calls' (RPC) angewendet [39]. Diese sind Teil der objektorientierten Middleware (siehe Abschnitt 6.2.3).

Es ist anzumerken, dass die frühere Definitionen des W3C [252] für einen Web-Service keine spezielle Implementierung über SOAP und WSDL vorgeschrieben hat. Die Gründe für diese restriktivere Definition sind allerdings unklar. Die weniger restriktive Definition [252] schreibt für Definition und Beschreibung von Schnittstellen die Verwendung von XML vor und schließt damit eine wichtige Technologie aus, deren Bedeutung in den letzten Jahren seit ihrer Einführung bei Amazon [70] stark zugenommen hat: Den Services, die auf dem 'REpresentational State Transfer' (REST) nach [71] basieren. REST ist eine spezielle Architektur für verteilte Systeme (für Hypertext und Multimedia=Hypermedia), welche auf Befehlen des HTTP-Protokoll beruht [71]. Diese ist nach [71] unter anderem durch folgende Eigenschaften gekennzeichnet:

- Client-Server-Architektur
- Zustandslosigkeit: Jede Anfrage des Servers an den Client muss alle Informationen enthalten, die notwendig ist, um die Anfrage zu verstehen.
- Cache: Antworten müssen explizit als pufferfähig (cacheable) oder nicht pufferfähig gekennzeichnet sein. Wenn eine Antwort als pufferfähig gekennzeichnet ist, darf der Client diese zu einem späteren Zeitpunkt für ähnliche Anfragen verwenden.
- Anforderungen an ein standardisiertes bzw. einheitliches Interface (zur Spezifikation der Anforderung siehe [71])
- Systemarchitektur in Schichten, welche die Kapselung von Services erlaubt.
- Code-On-Demand: Die Funktionalität der Clients kann über den Download von Applets oder Skripten erweitert werden.

Web-Services, die nach diesen Prinzipien aufgebaut sind, werden als RESTful bezeichnet [204]. Die Architektur und der Datenaustausch eines RESTful Services können deutlich einfacher gehalten werden als die eines großen Web-Service, der auf SOAP basiert [204]. Für eine umfassende Beschreibung von RESTFul Services sei auf [204] verwiesen.

Da sich in den letzten Jahren haben sich die RESTFul Services bewährt und zunehmend etabliert haben, werden diese im Rahmen dieser Arbeit als Web-Services betrachtet, auch wenn diese nicht der aktuellen Definition des W3C [254] entsprechen. Dies entspricht auch dem allgemeinen Sprachgebrauch in der IT-Branche, was durch Bücher, die 'RESTful' und 'Web Services' im Titel enthalten [204, 2, 208],

belegt wird. Damit stellen Web-Services auf SOAP-Basis und RESTful Services die wichtigsten Vertreter in diesem Bereich dar.

6.2.3 Middleware

Middleware organisiert die Interaktion zwischen Anwendungen über verschiedene Plattformen hinweg. Middleware ist damit eine Lösung für das Problem der Integration von mehreren Servern und Anwendungen unter einer gemeinsamen Schnittstelle [3]. Dabei bietet Middleware die Möglichkeit der Abstraktion, so dass die Komplexität der Entwicklung einer verteilten Anwendung vor dem Programmierer teilweise verborgen werden kann [3]. Ziel der Anwendung von Middleware ist die Anwendungsunabhängigkeit und die Kapselung des physikalischen Orts eines Objekts, so dass der Entwickler überhaupt keine Kenntnis mehr davon haben muss, wo die Objekte und Methodenaufrufe, mit denen er programmiert, zur Laufzeit ausgeführt werden [139]. [3] listet verschiedene Arten von Middleware auf, von denen hier nur 'RPC-basierte Systeme' und 'Object broker' sowie 'Application Server' betrachtet werden sollen. Der interessierte Leser sei für weitere Arten von Middleware auf [3] verwiesen.

'Remote Procedure Calls' (RPC) sind eine Technologie für die Kommunikation über Netzwerke und wurden bereits in den 80er Jahren durch [20] eingeführt [3]. Der Grundgedanke von RPC nach [20] ist der Aufruf einer Prozedur an einem physisch entfernt liegenden Computer. RPC´s sind in diesem historischen Kontext nicht objektorientiert, können aber in objektorientierten Architekturen verwendet werden. Das Grundprinzip der RPC´s ist in vielen Architekturen von Middleware vorhanden [3]. Für die genaue Darstellung der Arbeitsweise von RPC´s wird der Leser auf [20] verwiesen. Die Verwendung von RPC´s ist im Allgemeinen mit der Verwendung von SOAP als Datenaustauschformat verbunden wohingegen RESTful Services einen Gegenentwurf zur RPC darstellen [204]. Die Vor- und Nachteile entsprechender Architekturen wurden in den letzten Jahren ausführlich diskutiert [69, 195]. Ein wesentlicher Vorteil von RESTful Services ist die bessere Skalierbarkeit und Performance [69], sowie die einfachere Architektur [195]. RPC-basierte Webservice-Architekturen verfügen auf der anderen Seite über eine höhere Flexibilität in der Architektur, so dass diese auch unabhängig von HTTP geschaffen werden können [195]. Darüber sind RPC-basierte Webservice-Architekturen leichter zu programmieren [195].

Ein weit verbreiteter Ansatz zum Datenaustausch über Middleware sind 'Object broker'. 'Object broker' sind Middleware-Infrastrukturen, welche in das Paradigma von RPC den Gedanken der Objektorientierung aufnehmen und so die Interoperabilität von Objekten auf verschiedenen Systemen unterstützen [3]. Der wichtigste

Vertreter eines Object brokers ist die 'Common Object Request Broker Architecture' (CORBA) [3, 139], welcher von der OMG in [187] spezifiziert wird. Für die Darstellung der Architektur von von CORBA wird auf [3, 139] verwiesen. Aufgrund seiner Komplexität hat CORBA mittlerweile stark an Bedeutung verloren [139]. Der Grund hierfür liegt in der hohen Komplexität von CORBA-Anwendungen und in Sicherheitsmängeln, die in der Architektur begründet sind [101]. Zusätzlich werden in [101] Mängel in der Versionsverwaltung von Anwendungen und weitere technische Probleme moniert.

Wichtige Vertreter der objektorientierten Middleware finden sich im Bereich der sprachspezifischen Anwendungsserver. In diesen geht allerdings die Eigenschaft der Plattformunabhängikeit verloren. Die beiden wichtigsten Programmiersprachen für die sprachspezifische Middleware sind Java und C# [139]. In dem Umfeld dieser beiden Programmiersprachen hat sich eine Vielzahl an Lösungen am Markt etablieren können, welche für die jeweilige Programmiersprache spezifisch sind. Für Java wird mit der 'Java Enterprise Edition' (Java EE) in [191] von Oracle eine Lösung spezifiziert. Der Gegenentwurf zu Java EE von Microsoft ist das '.NET Framework' [165]. Hierbei ist für die netzbasierte Kommunikation insbesondere die 'Internet Information Services' (IIS) als Webserver (Spezifikation: [164]) und die 'Windows Communication Foundation' (WCF) als Framework zum Austausch von Objekten (Spezifikation: [166]) von Bedeutung. Neben Java EE und .NET wurde eine Vielzahl von Lösungen im Bereich der Anwendungsservern entwickelt, die nicht zwingend auf diesen Ansätzen beruhen müssen.

Der Nachteil einer sprachgebundenen Lösung ist, dass jeder Teilnehmer an einem Datenaustausch an eine bestimmte Architektur und an bestimmte Anbieter gebunden wird. So muss in einer Infrastruktur jeder Teilnehmer dieser Infrastruktur die Middleware auch bei sich lokal unterstützen. Bei der Verwendung von Java EE ist es nicht ohne Weiteres möglich, eine neue Datenquelle zu integrieren, welche mit .NET arbeitet. Prinzipiell kann diese neue Datenquelle nur dann in eine Java EE-Infrastruktur eingebettet werden, wenn diese auf Java EE umgestellt wird oder zusätzlich einen Java EE-Zugang anbietet.

Java EE und das '.NET-Framework' sind aktuell die wichtigsten Vertreter im Bereich der objektorientierten Middleware. Diese weisen zwar gegenüber CORBA den Nachteil auf, dass diese plattformspezifisch arbeiten. CORBA hat sich aufgrund der Komplexität von CORBA-Anwendungen nicht als praxistauglich erwiesen und ist heute von untergeordneter Bedeutung. Die Wahl der Anwendung von Jave EE oder '.NET' sollte aber gut überlegt sein, da aufgrund der Plattformabhängigkeit sprach-

gebundener Lösungen eine solche Entscheidung nur mit hohem Aufwand verändert werden kann.

6.3 Software zur Informationsintegration

Software zur Informationsintegration wird sowohl von kommerzieller Seite wie von wissenschaftlicher Seite angeboten. Die Ansätze basieren überwiegend auf den in Abschnitt 6.1 und 6.2 vorgestellten Grundlagen. Diese Software ist im kommerziellen Bereich an einen bestimmten Hersteller gebunden. In Abschnitt 6.3.1 und 6.3.2 wird ein kurzer Überblick über die wichtigsten Anbieter gegeben. Ein besonderes Framework zur Informationsintegration aus dem wissenschaftlichen Bereich ist das DaltON-Framework. Dieses wird in Abschnitt 6.3.3 vorgestellt.

6.3.1 Kommerzielle Produkte

Im kommerziellen Bereich gibt es ein weites Spektrum an Anbietern, die auch eine enorme wirtschaftliche Bedeutung erlangt haben – was an den in [238] veröffentlichen Umsätzen ersichtlich ist. Die Marktführer sind dabei nach dem 'Magic Quadrant for Data Integration Tools' von Gartner [238]:

- Informatica mit der Informatica Platform [109]
- IBM z.B. mit IBM InfoSphere [107]
- Oracle z.B. mit dem Oracle Data Integrator [190]
- SAP z.B. mit dem SAP Data Integrator [209]
- SAS-DataFlux z.B. mit dem SAS Enterprise Data Integration Server [210]

Eine umfassend Analyse über die Marktführer in diesem Bereich und konkurrierende Produkte findet sich in [238, 239]. Dabei wurde im Magic Quadrant for Data Integration Tools' [239] Informatica als der Anbieter mit der vollständigsten Vision und den besten Fähigkeiten ausgezeichnet.

Diese Lösungen wurden im Allgemeinen vor dem Hintergrund des Einsatzes in einem Unternehmen entwickelt und werden in Kombination mit anderen Produkten und umfassender Beratung angeboten. Sie sind aber für den Einsatz in offenen Infrastrukturen aus Kostengründen ungeeignet. Auch einfachere Lösungen wie von Altova [5] sind für einen flächendeckenden Einsatz in offenen Infrastrukturen immer noch zu kostspielig.

6.3.2 Produkte aus der Wissenschaft

Auch im wissenschaftlichen Bereich existieren verschiedene Ansätze und Frameworks zum Datenaustausch über heterogene Quellen. Dabei ist die Unterstützung von Prozessen von besonderem Interesse. Dementsprechend werden an dieser Stelle vor allem Lösungen betrachtet, in denen ein Prozessgedanke verfolgt wird. Darüber hinaus werden Lösungen analysiert, welche einen ähnlichen Anwendungshintergrund wie die kommerziellen Lösungen haben, aber dafür meistens kostenfrei sind. Ein Nachteil der Lösungen in wissenschaftlichen Bereich ist aber, dass die Lösungen im Allgemeinen nicht so weit entwickelt sind und die Entwicklung auch teilweise eingestellt wird. Darüber hinaus besteht im wissenschaftlichen Bereich im Allgemeinen keine Betreuung (Support) durch die Anbieter von Lösungen.

Von besonderem Interesse im wissenschaftlichen Bereich ist Kepler, da in Kepler über 'Scientific Workflows' (SWF) ein Prozessgedanke verfolgt wird [4] und Kepler einen konkurrierenden Ansatz zu DaltON (siehe Abschnitt 6.3.3) darstellt. Der SWF ist nach [4] der komplette Prozess, den ein Wissenschaftler von der Erhebung der Rohdaten bis zur Publikation ausführt. Dabei orientiert sich der Einsatz von SWF´s an Businessprozessen, wobei ein wesentlicher Unterschied darin besteht, dass in SWF´s komplexe und heterogene Daten auftreten. Dementsprechend tritt in Vergleich zu Businessprozessen in SWF´s der Datenfluss in den Vordergrund [4]. Informationsintegration ist in Kepler nur indirekt durch den Aufruf von externen Diensten in einem SWF verfügbar [205]. Ein weiterer Ansatz aus diesem Bereich ist Taverna, in welchem der Workflowgedanke in die Bioinformatik übertragen wird [183]. Taverna basiert auf der Koordination von Webservices und stellt elementare Möglichkeiten zur Informationsintegration sowie Data Provenance auf Basis von Workflows zur Verfügung [205]. Sowohl Kepler als auch Taverna arbeiten mit materialisierter Informationsintegration [205].

Bioflow, Fusionplex und HumMer basieren hingegen auf virtuelle Informationsintegration[1] [205]. Ein Vergleich dieser Frameworks mit Kepler und Taverna findet sich in Abbildung 6.1. Dabei fällt auf, dass BioFlow, Fusionplex und HumMer in [205] nicht über eine prototypische Entwicklung hinaus verfügbar waren. Auch aktuell konnten keine frei verfügbaren Versionen dieser Frameworks zum Test gefunden werden. Das Ergebnis der Evaluation aus [205] ist in Tabelle 6.1 dargestellt. Für die vollständige Evaluation wird der interessierte Leser auf [205] verwiesen.

Die in [205] vorgestellten Frameworks bieten keine Gesamtarchitektur zum Austausch von Daten über ein Netzwerk an. Sie können deswegen nur für die Lösung

[1] In [205] wird zusätzlich noch SnapLogic betrachtet. Dieses ist aber ein kommerzielles Produkt.

	Taverna	Kepler	Bioflow	Fusionplex	HumMer
Schema Matching	No	No	Based on ontologies	No	Yes
Schema Mapping	No	No	Yes	Yes	Yes
Extensibility	Through webservices	Through webservices and applications	Yes	Possible, but not supported	Possible, but not supported
Variety of data sources and sinks	Through webservices	Yes	Database, XML files, web forms	Yes	Yes
Support of composite data	Based on XML	Based on XML	Based on XML	No	No
Shared computation	Webservices	Webservices	Tools can be outsourced	No	Possible, but not supported
Data Provenace	Workflows	Only with extensions	No	No	Rudimental
Implementation	Yes	Yes	Prototype	Prototype	Prototype

Tabelle 6.1: Vergleich von Frameworks zur Informationsintegration nach [205]

des Teilproblems der Informationsintegration bei der Architektur einer Infrastruktur für die Biodiversitätsinformatik herangezogen werden. Besonders erwähnenswert sind aber Kepler und Taverna, da diese Workflowunterstützung anbieten. Kepler und Taverna werden in Projekten wie DataONE eingesetzt (siehe Abschnitt 7.4.4).

6.3.3 DaltON

Das 'Data Logistics with Ontologies' (DaltON) Framework geht auf [117] zurück und liegt der Doktorarbeit [201] zu Grunde. Es wurde in leichten Variationen in [40, 115, 118, 202] veröffentlicht und basiert auf eine Zusammenarbeit der Universität Bayreuth mit Universität Paris Est. DaltON stellt dabei insbesondere durch die Anwendung der datenorientierten Perspektive aus POPM eine wesentliche Vorarbeit bei der Entwicklung einer eigenen Infrastruktur für die Biodiversitätsinformatik (siehe Kapitel 8) dar und wird deshalb kurz vorgestellt. Für eine umfassende Beschreibung von DaltON wird auf [201] verwiesen. Die Ausführungen in diesen Abschnitt beziehen sich wenn nicht anders vermerkt auf die Veröffentlichung in [40], in welchem die semantische Integration bei DaltON besonders detailliert beschrieben ist.

Grundlage von DaltON sind Scientific Workflows mit POPM (siehe Abschnitt 4.1) als Modellierungssprache. Innerhalb eines Prozessmodells wird von DaltON se-

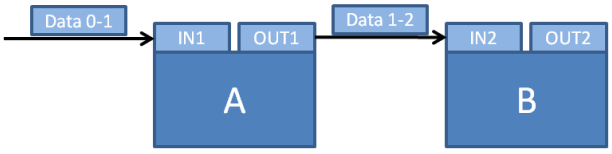

Abbildung 6.2: Datenaustausch zwischen zwei Prozessschritten in DaltON nach [117]

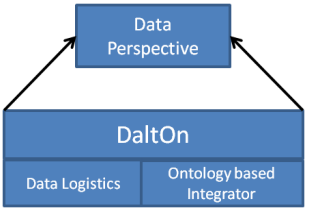

Abbildung 6.3: Komponenten der datenorientierten Perspektive in DaltON nach [117]

mantische Informationsintegration auf Basis von Ontologien (siehe Abschnitt 6.1) betrieben. Hierbei wird ein wesentlicher Fokus auf die datenorientierte Perspektive von POPM gelegt. In dieser wird der Datenaustausch zwischen zwei Prozessschritten modelliert. Jeder Prozessschritt produziert hierbei Daten in einem bestimmten Format und mit einer bestimmten Bedeutung. Ein Beispiel für den Datenaustausch zwischen zwei Prozessschritten findet sich in Abbildung 6.2. In dieser Abbildung ist der Prozess 'B' Konsument der Daten 'OUT1' des Prozesses 'A'. Wenn das Datenmodell 'OUT1' zu dem Datenmodell von 'IN2' nicht syntaktisch und semantisch kompatibel ist, wird Informationsintegration benötigt.

Ziel von DaltON ist nun, diesen Datentransport syntaktisch und semantisch zu bewältigen. Dabei treten mit dem Datenaustausch an sich und der Integration der übertragenen Daten zwei Problembereiche auf [117]:

- Die Datenlogistikkomponente ist verantwortlich für die technische Datenübertragung.

- Die Informationsintegrationskomponente stellt die semantische Kompatibilität über Ontologien her.

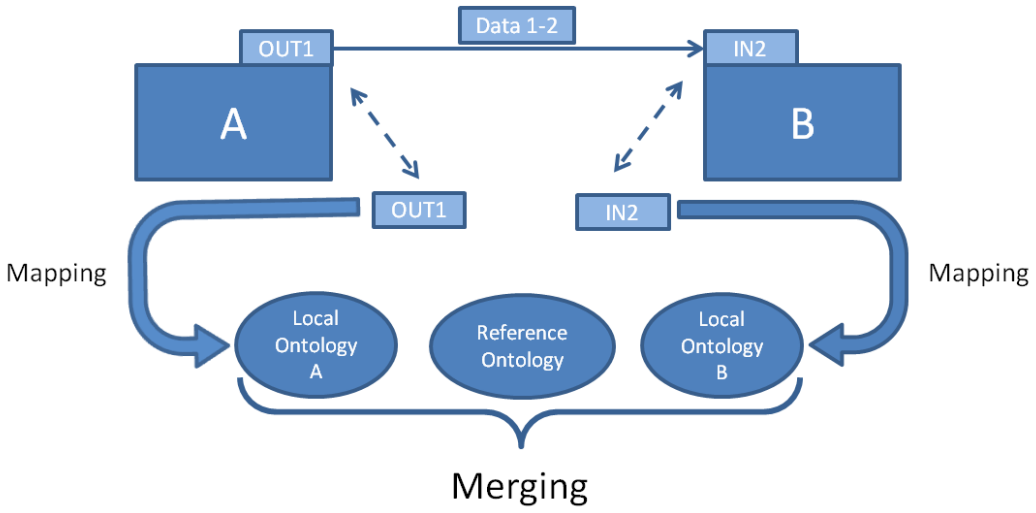

Abbildung 6.4: Semantische Integration mit DaltON nach [117]

Dementsprechend wird die datenorientierte Perspektive in zwei Komponenten unterteilt (Abbildung 6.3). Die semantische Integration wird auf Basis von Ontologien ausgeführt. Dabei werden für eine Datentransformation drei Ontologien benötigt:

- Die Referenzontologie, welche ein Vokabular für eine Domäne enthält, auf das sich alle Teilnehmer verständigt haben
- Zwei lokale Ontologien zur Repräsentation des Wissens im exportierenden und importierenden System, welche von der Referenzontologie abgeleitet sind oder diese erweitern

Bei der semantischen Integration der Daten werden diese drei Ontologien zu einer gemeinsamen Ontologie vereinigt. Da beide lokale Ontologien auf der Basis der Referenzontologie spezifiziert sind, sind diese zueinander kompatibel. Die Konzepte der Ontologie des exportierenden Prozesses A können in die Konzepte des importierenden Prozess B übertragen werden (siehe Abbildung 6.4). Dabei muss zwischen den Konzepten von A und B nicht zwingend semantische Äquivalenz bestehen. Es kann vielmehr auch der Fall auftreten, dass Konzepte von A Spezialisierungen oder Generalisierungen von Konzepten von B sind bzw. sich Konzepte von A und B semantisch überlappen. Hierbei stellen Generalisierung und semantische Überlappung problematische Fälle dar, die im Allgemeinen nicht mehr automatisch gelöst werden können.

Da DaltON keine einheitliche Sicht auf die Daten sondern eine Kette von Integrationsschritten mit Datentransport liefert, ist DaltON kein Integrationssystem von

heterogenen Quellen im Sinne von Definition 1. Dies würde erfordern, dass DaltON eine einheitliche Sicht für die datenorientierte Perspektive im gesamten Prozessmodell erzeugen kann. Da die semantische Integration immer nur zwischen zwei Prozessen und nicht mit allen Prozessen des Prozessmodells erfolgt, wird allerdings keine einheitliche Sicht für alle beteiligten Datenspeicher erzeugt.

Die Referenzontologie liefert ein gemeinsames Vokabular, auf das sich alle Teilnehmer verständigen müssen. Da Daten überwiegend in relationalen Datenbanken gespeichert werden ist zunächst das Mapping der Datenbank in eine lokale Ontologie erforderlich. Erst anschließend kann ein Mapping in die Referenzontologie erfolgen. Die Daten in der Referenzontologie werden dann in einem nächsten Schritt in die lokale Ontologie des Austauschpartners übertragen. Die Transformation der Daten in das relationale Datenbankschema des Austauschpartners stellt einen weiteren Integrationsschritt dar. Damit sind für eine Datenübertragung in DaltON faktisch 4 Mappings notwendig und die Daten sind insgesamt in fünf Formaten repräsentiert. Zusätzlich besteht der Nachteil, dass damit nur die Datenübertragung für zwei spezielle Teilnehmer einer Infrastruktur ermöglicht wird. Für den Datenaustausch mit einem dritten Teilnehmer sind zusätzliche Schritte notwendig. Damit stellt DaltON eine strukturierte Vorgehensweise zum Datenaustausch in einer Infrastruktur dar. Es sind aber viele aufwändige Abbildungs- und Transformationsschritte erforderlich.

Dies wird deutlich, wenn man sich die Anwendung von DaltON zur Informationsintegration in einem Netzwerk wie GBIF vorstellt. Hier müsste für jeden Datenaustausch zwischen zwei Teilnehmern des Netzwerks zunächst semiautomatisch mit Hilfe der Referenzontologie die semantische Integration zwischen exakt diesen speziellen Teilnehmern erfolgen. Sobald einer dieser Teilnehmer mit einem dritten Teilnehmer kommuniziert, muss die semantische Integration erneut für diese Teilnehmer ausgeführt werden. Somit müsste jeder Teilnehmer zunächst für sich eine lokale Ontologie auf Basis der Referenzontologie erstellen. Dies ist in der Praxis eine ungünstige Konstellation. Darüber hinaus erfordert die Arbeit mit Ontologien bei Datenquellen, die auf relationalen Datenbanken beruhen, zunächst eine Transformation der relationalen Daten in die Ontologie. Aufgrund dieser Erfordernisse ist die praktische Anwendung von DaltON im Bereich der Biodiversitätsforschung nicht direkt möglich, kann aber als Basis für den Datenaustausch und die Informationsintegration in einer Infrastruktur verwendet werden.

In DaltON sind wichtige Grundlagen für den Aufbau einer Infrastruktur enthalten. An DaltON wird die Bedeutung von Prozessen für den Datenaustausch und die semantische Integration deutlich. Es ist somit nicht nur wichtig, Daten von verschie-

denen Orten zu übertragen und in passende Formate zu konvertieren, sondern auch zwingend erforderlich, dass sich die Bedeutung der Daten bei dieser Übertragung nicht verändert. Dazu ist ein gemeinsames Verständnis der Anwendungsdomäne erforderlich, welche in DaltON als Referenzontologie integriert ist. Ein entsprechendes gemeinsames Verständnis muss aber nicht auf der technischen Grundlage einer Ontologie erfolgen, sondern kann auch durch ein gemeinsames Datenaustauschformat wie ABCD oder DwC gebildet werden.

6.4 Zusammenfassung

In diesem Kapitel wurden die elementaren Klassifikationen aus dem Bereich der Informationsintegration vorgestellt. Mit diesen Begriffen konnten anschließend grundlegende Technologien zum Datenaustausch über verteilte, heterogene Quellen erklärt werden. Dabei wurde deutlich, dass Informationsintegration und -austausch wesentliche Grundlagen einer Infrastruktur sind. Allerdings reichen diese noch nicht, aus um den Datenaustausch in einer Infrastruktur vollständig zu beschreiben, da dadurch nur die technische Sichtweise einer Infrastruktur beschrieben werden kann und zusätzlich personelle und institutionelle Gegebenheiten berücksichtigt werden müssen (vgl. Abschnitt 7.1). Existierende Softwarelösungen für die Informationsintegration konnten also nur vor dem Hintergrund des Einsatzes als Teilkomponente in einer Infrastruktur betrachtet werden. Hierbei scheiden kommerzielle Anbieter in der Biodiversitätsinformatik aus Kostengründen aus. Im wissenschaftlichen Bereich finden sich hingegen einige interessante Ansätze, die als Teilelemente einer Infrastruktur verwendet werden können. Diese sind aber nicht bis zur Marktreife implementiert. Darüber hinaus müssten diese allgemeinen Lösungen domänenspezifisch angepasst werden, damit die Konfiguration der Software die Anwender in der Biodiversitätsforschung nicht überfordert. Mit DaltON wurde ein Framework vorgestellt, welches wie Kepler auf Basis von Prozessen arbeitet. Kernelement von DaltON ist die semantische Integration, die aber in ihrer Implementierung und mit der Abhängigkeit von lokalen Ontologien und einer Referenzontologie für die Anwendung in einer offenen Infrastruktur ungeeignet ist. Trotzdem bietet DaltON interessante Ansätze, welche in Kapitel 8 weiterverfolgt werden.

Kapitel 7

Evaluation von Infrastrukturen

Daten aus heterogenen Quellen werden mit Hilfe von Infrastrukturen zwischen diesen ausgetauscht. Die Grundlagen, auf denen diese Infrastrukturen aufbauen, wurden in Kapitel 6 vorgestellt. Auf Basis dieser Grundlagen werden im folgenden Kapitel Infrastrukturen betrachtet. Dazu wird in Abschnitt 7.1 zunächst definiert, was unter einer Infrastruktur verstanden wird und es werden allgemeine Eigenschaften beschrieben, nach denen sich Infrastrukturen klassifizieren lassen. Als ein wesentlicher Teil einer Infrastruktur wird ihr Anwendungshintergrund betrachtet, auf den an dieser Stelle explizit eingegangen wird. In Abschnitt 7.2 werden die Eigenschaften von Infrastrukturen in verschiedene Bereiche unterteilt, welche als Ebenen bezeichnet werden.

Es werden zusätzlich Qualitätskriterien benötigt, die darüber Auskunft geben, wie gut Datenaustausch und die Informationsintegration in einem Netzwerk gelingen. Für die Bewertung von Infrastrukturen konnten in der Literatur keine existierenden Frameworks gefunden werden, so dass deren Entwicklung dieser Kriterien und Klassifikation eine neue Entwicklung darstellt. Die Entwicklung eines Frameworks zur Evaluation wird deshalb als neue Entwicklung in Abschnitt 7.3 vorgestellt. Dazu wird zunächst ein Katalog für Qualitätskriterien in Infrastrukturen in Abschnitt 7.3.1 vorgestellt. Die Kriterien sind generisch formuliert, so dass dieser Katalog in verschiedenen Domänen angewendet werden kann. Anschließend wird die Grundstruktur der Evaluation mit dem 'Infrastructure Evaluation Framework' (IEF) in Abschnitt 7.3.2 eingeführt. Dieses wird in Abschnitt 7.3.3 an die Domäne der Biodiversitätsinformatik angepasst und zu dem domänenspezifischen Framework IEF-Biodiv ausgebaut.

In Abschnitt 7.4 wird IEF-Biodiv zur Evaluation von Infrastrukturen in der Biodiversitätsinformatik und in verwandten, wissenschaftlichen Bereichen genutzt. Aufgrund der besonderen Bedeutung wird hierbei das GBIF-Netzwerk besonders detailliert vorgestellt und evaluiert. Neben der GBIF-Architektur im Allgemeinen wird

auch die spezielle Nutzung der GBIF-Infrastruktur im Kontext des IBF-Projekts (siehe Abschnitt 2.9) angewendet und dazu separat evaluiert. Damit wird die GBIF-Infrastruktur in zwei Varianten besonders ausführlich diskutiert. Dies liegt an der besonderen Bedeutung der GBIF-Infrastruktur für die Biodiversitätsinformatik. Um dies konsequent weiterzuverfolgen, wurde die Anordnung der Infrastrukturen in Abschnitt 7.4 nach ihrer Bedeutung für die Biodiversitätsinformatik gewählt. Die wichtigeren Infrastrukturen werden dabei detaillierter vorgestellt und evaluiert. In Abschnitt 7.5 werden die Ergebnisse der Evaluation zusammengefasst und das Fazit als Ausgangspunkt für die Entwicklung einer eigenen Infrastruktur in Kapitel 8 verwendet.

7.1 Infrastrukturen zum Datenaustausch

Infrastrukturen zum Austausch von Daten können nicht willkürlich entstehen. Sie werden zu einem bestimmten Zweck vor einem bestimmten **Anwendungshintergrund** erschaffen. Durch diesen Anwendungshintergrund wird festgelegt, welche technischen und organisatorischen Elemente Teil dieser Infrastruktur sein müssen und welche Art von Daten über die Infrastruktur transportiert werden. Dazu soll zunächst der Begriff der **Infrastruktur** definiert werden. Der Begriff 'Infrastruktur' wird je nach Kontext mit verschiedener Bedeutung genutzt. Ursprünglich entstammt der Begriff dem NATO-Vokabular und umfasst erdgebundene Anlagen, welche der Mobilität dienen [194]. Ausgehend von dieser Bedeutung hat sich der Begriff 'Infrastruktur' weiterentwickelt und wird insbesondere in der IT dazu verwendet, um das Zusammenspiel verschiedener Komponenten innerhalb eines Systems zu beschreiben. Im folgenden Abschnitt wird 'Infrastruktur' für den Kontext des Datenaustauschs definiert.

Dazu soll zunächst auf den Begriff der 'IT-Infrastruktur' zurückgegriffen werden. Dieser ist nach Patig [194] nicht einheitlich definiert. Es gibt nach [194] zwei Sichtweisen auf den Begriff 'IT-Infrastruktur':

- Technische Sicht: Hardware, Software, baulichen Einrichtungen zum Betrieb von Software (siehe Abschnitt 6.2)
- Sicht des Informationsmanagements: In dieser Sichtweise sind zusätzlich personelle und institutionelle Gegebenheiten enthalten [99, 194]. Die IT-Infrastruktur umfasst damit auch die Anwendung von Gesetzen und Normen, Datenschutzrichtlinien und das Wissen von Mitarbeitern. Diese Sichtweise der IT-Infrastruktur wird nach [100] auch 'Informationsinfrastruktur' bezeichnet.

Beide Sichtweisen gehen nach [194] davon aus, dass der Zweck einer Infrastruktur der Betrieb von Anwendungssoftware ist. Patig [194] betrachtet in ihrer Definition von 'IT-Infrastruktur' diesen Begriff aus dem Blickwinkel eines einzelnen Anbieters wie z.B. eines Unternehmens. Infrastrukturen zum Datenaustausch hingegen haben den reibungslosen Datenaustausch zwischen verschiedenen Datenspeichern und die Präsentation der ausgetauschten Daten als Aufgabe. Dabei kann zwar auch gemeinsame Anwendungssoftware zum Einsatz kommen – dies ist aber nicht der einzige Aspekt, aus dem diese Infrastruktur besteht.

Damit können beide Sichtweisen der Definition einer IT-Infrastruktur nicht direkt auf eine **Infrastruktur zum Datenaustausch** angewendet werden. Innerhalb einer Dateninfrastruktur treten Teilnehmer dieser Infrastruktur in unterschiedlicher Form auf. So ist ein Datenanbieter innerhalb dieser Infrastruktur nicht nur die Datenbank, die dieser in die Infrastruktur einbringt. Er verfügt vielmehr auch über eine organisatorische Struktur mit Entscheidungsträgern, welche über die Teilnahme an der Infrastruktur bestimmen. Folglich ist eine rein technische Sichtweise nicht ausreichend.

Es soll für die Definition einer **Dateninfrastruktur** die Sichtweise des Informationsmanagements als Ausgangspunkt dienen, da diese auch personelle und organisatorische Elemente berücksichtigt. Dabei sind die Daten mehr als ihre physische Repräsentation in einer Datenbank. Daten verfügen vielmehr über ein konzeptuelles Schema, nach welchem diese logisch strukturiert sind. Darüber hinaus entsprechen Daten Aussagen über die reale Welt und haben damit eine bestimmte Bedeutung. Diese Semantik der Daten ist aber mit Sicherheit weder ein Teil der organisatorischen Struktur eines Datenanbieters noch ein Teil des technischen Datenspeichers, in welchem diese Informationen nur physisch hinterlegt ist.

Ein weiteres wesentliches Element einer Infrastruktur zum Datenaustausch ist ein **gemeinsames Verständnis** über die Ziele dieser Infrastruktur und die Art der Daten, die über diese Infrastruktur ausgetauscht und publiziert werden. So muss eine Datensenke Daten einer anderen Datenquelle bei materialisierter Integration in ihre eigenen Datenbestände integrieren können. Dies ist neben technischen Voraussetzungen nur möglich, wenn sowohl die Datenquelle als auch die Datensenke über ein gemeinsames Verständnis über die Struktur und die Semantik der auszutauschenden Daten besitzen. Folglich muss die Sicht des Informationsmanagements um dieses gemeinsame Verständnis der Daten und die Datenbestände erweitert werden.

Neben **Datenanbietern** existieren weitere Teilnehmer in einer Dateninfrastruktur, wie **Datenkonsumenten** und **Koordinationsstellen** für den Datenaustauschs

oder zur **Publikation** der Daten. Diese verfügen neben rein technischen Elementen auch über organisatorische Strukturen und müssen das gemeinsame Verständnis teilen. Damit lässt sich der Begriff **Dateninfrastruktur** wie folgt definieren:

Definition 7.1: Dateninfrastruktur

In einer Dateninfrastruktur ist der Datenaustausch zwischen verschiedenen Datenspeichern organisiert. Teile dieser Infrastruktur sind neben den Datenspeichern an sich auch Einheiten zum Auffinden und zur Präsentation der Daten, sowie Datenkonsumenten. Alle Teilnehmer verfügen dabei neben einer technischen Infrastruktur aus der zum Datenaustausch benötigten Hardware, Software und baulichen Einrichtungen über eine organisatorische Repräsentation. Darüber hinaus muss unter allen Teilnehmern ein gemeinsames Verständnis über den Zweck und Umfang des Datenaustauschs und die Semantik der Daten bestehen.

Weil sich diese Arbeit schwerpunktmäßig mit Dateninfrastrukturen befasst, wird im Folgenden unter dem Begriff 'Infrastruktur' eine Dateninfrastruktur verstanden, sofern dies nicht explizit anders vermerkt ist. Mit dieser Definition können Infrastrukturen mit unabhängigen Teilnehmern genauso wie zentrale Strukturen erfasst werden. Wichtig ist, dass die Infrastruktur nicht nur aus ihren technischen Elementen besteht, sondern auch aus den organisatorischen Einheiten und einem gemeinsamen Verständnis der Art der Daten. Die organisatorischen Einheiten müssen berücksichtigt werden, da eine IT-Infrastruktur ohne gemeinsame (Quasi-)Standards, Vereinbarungen und Protokolle nicht arbeiten kann. Die wichtigste Vereinbarung ist hierbei über die **Art und den Umfang des Datenaustauschs**. Dazu gehört neben der Vereinbarung eines gemeinsamen technischen Austauschformats auch eine Vereinbarung darüber, welche Art von Daten in welchem Umfang über das Netzwerk ausgetauscht werden können.

So versteht sich z.B. GBIF als dezentrales Netzwerk zum Austausch von Biodiversitätsdaten aller Art. Implizit wird allerdings eine Formatierung der Daten in DwC oder ABCD vorausgesetzt. Mit der Wahl von DwC und ABCD ist nicht nur ein bestimmtes technisches Austauschformat (XML) vorgegeben, sondern es werden auch die semantischen Beschränkungen von ABCD und DwC akzeptiert. Informationen, die nicht in DwC oder ABCD ausgedrückt werden, können nicht über das Netzwerk übertragen werden. Infrastrukturen zum Datenaustausch werden im Allgemeinen aufgrund der Notwendigkeit eines gemeinsamen Anwendungshintergrund innerhalb einer bestimmten Domäne angewendet.

7.2 Ebenen einer Infrastruktur

Wie im vorangegangenen Abschnitt gezeigt wurde, treten die Elemente einer Infrastruktur in verschiedenen Rollen auf. Ein Datenanbieter verfügt zum einen über den Datenbestand an sich, einen physischen Datenspeicher und eine organisatorische Struktur. Ein einzelner Datenanbieter muss dabei als **Einheit** betrachtet werden. Diese Einheit verfügt allerdings je nach Blickwinkel über unterschiedliche Rollen in dieser Infrastruktur. Um diesen verschiedenen Sichtweisen gerecht zu werden, muss eine Infrastruktur in verschiedene Bereiche unterteilt werden, die als **Ebenen** bezeichnet werden. Der Begriff 'Ebene' wurde gewählt, da hierbei ein Element einer Infrastruktur innerhalb dieser Infrastruktur in verschiedenen Rollen auftreten kann. So verfügt z.B. ein Repositorium neben einem Datenspeicher auch über eine organisatorische Struktur. Beide Aspekte sind für die Evaluation einer Infrastruktur relevant – fließen aber auf unterschiedlichen Ebenen in die Analyse ein. Die Ebenen einer Infrastruktur sind in folgender Weise definiert:

Definition 7.2: Ebene

Eine Ebene ist eine bestimmte thematische Sichtweise auf eine Infrastruktur, in der eine Einheit dieser Infrastruktur gemäß ihrer Rollen separat analysiert werden kann.

Die Ebenen werden in Anlehnung an den Sprachgebrauch bei POPM in orthogonale Bereiche unterteilt, welche die Analyse einer Einheit in einer Infrastruktur jeweils aus einem speziellen, thematischen Blickwinkel ermöglicht:

- **operationale Ebene** (Wie?): Elemente zum technischen Austausch der Daten analog zur technischen Sicht der IT-Infrastruktur wie Hardware, Software, Services
- **organisatorische Ebene** (Wer?): Organisatorische Einheiten wie Organisationen, die Daten halten oder eine zentrale Koordinationsstelle, sowie Personen, die für diese arbeiten
- **funktionale Ebene** (Was?): Globales Schema, Austauschschema, Datenbestand, Multimediadaten

Die Aufzählung der Ebenen ist nicht abschließend und kann bei Bedarf erweitert werden.

Eine **Einheit** in einer Infrastruktur ist im Allgemeinen auf verschiedenen Ebenen einer Infrastruktur repräsentiert und kann auch mehrfach in diversen Rollen auf einer

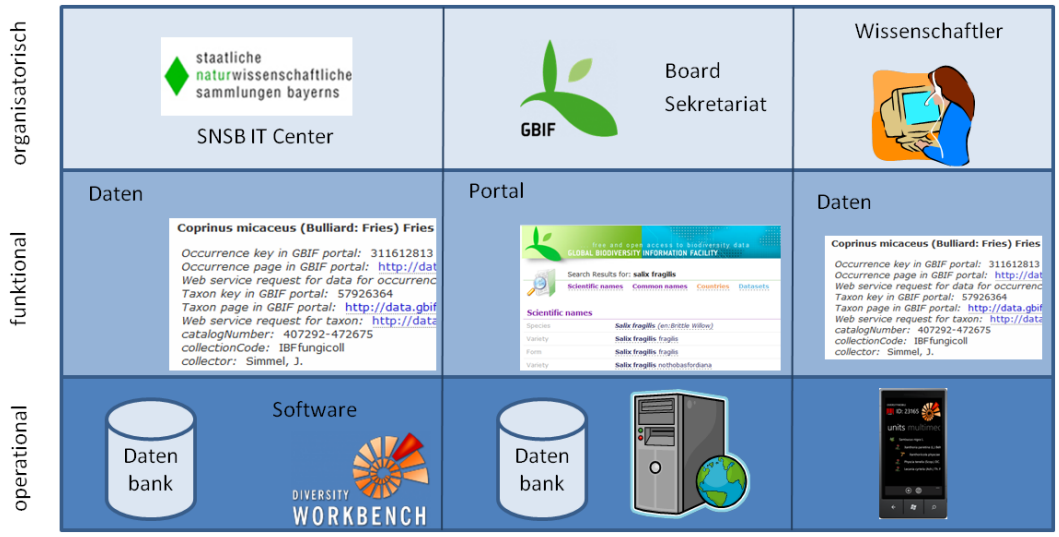

Abbildung 7.1: Beispiel für Einheiten einer Infrastruktur mit Ebenenzuordnung

Ebene auftreten. Dies ist der Fall, wenn eine Organisation neben einer Datenbank auf der operationalen Ebene auch Services anbietet. Sie muss allerdings nicht auf allen Ebenen repräsentiert sein, wenn eine Organisation zwar ein Datenportal aber keinen eigenen Datenbestand unterhält.

Ein **Repositorium** kann beispielsweise auf drei Ebenen auftreten:

- auf der **funktionalen Ebene** als reine Datenquelle, die über einen gewissen Datenbestand verfügt
- auf der **organisatorischen** Ebene als Einheit, in der Menschen arbeiten und auch Entscheidungen treffen
- auf der **operationalen Ebene** in Form der technischen Anlagen, die dieses betreibt

Wenn auf der organisatorischen Ebene entschieden wird, die Infrastruktur zu verlassen, gehen neben den institutionellen Ressourcen auch die Datenbestände und die angeschlossenen technischen Einrichtungen verloren. Damit ist der physischen Datenspeicher untrennbar mit den Daten und der Organisation verbunden.

Abbildung 7.1 beschreibt die verschiedenen Rollen von Einheiten innerhalb einer Infrastruktur am Beispiel von Elementen in der GBIF-Infrastruktur. In der 1. Spalte ist die das SNSB-IT-Centers als Teil des GBIF-Netzwerks in verschiedenen Ebenen dargestellt. Auf der organisatorischen Ebene verfügt das SNSB-IT-Center über einen internen Aufbau aus Abteilungen und Personen die am SNSB-IT-Center

arbeiten. Auf der funktionalen Ebene verfügt da SNSB-IT-Center über einen eigenen Datenbestand. Dieser in Datenbanken physisch gespeichert. Da die Datenbankserver technische Anlagen sind, sind diese auf der operationalen Ebene angesiedelt. Darüber hinaus stellt das SNSB Software, wie DiversityCollection zur Verfügung über die Daten bearbeitet und geladen werden können. DiversityCollection ist damit auch ein Teil der operationalen Ebene des SNB IT-Centers. Insgesamt bildet das SNSB-IT-Center aber eine Einheit. Wenn dieses nicht mehr existieren würde, wäre sowohl die Organisationsstruktur des SNSB-IT-Centers nicht mehr existent als auch der Datenbestand und die technischen Anlagen. Allerdings kann der Beitrag des SNSB-IT-Centers in den jeweiligen Bereichen nach Ebenen separat analysiert werden.

Analog dazu verfügt in der 2. Spalte GBIF mit dem GBIF-Board und dem GBIF-Sekretariat über eine Organisationsstruktur (organisatorische Ebene) und stellt über das GBIF-Portal seinen eigenen Datenbestand zur Verfügung (funktionale Ebene). Dazu muss GBIF in der operationalen Ebene technische Anlagen wie Datenbanken und Webserver unterhalten. In der 3. Spalte ist ein einzelner Wissenschaftler dargestellt, der Daten mit einem eigenen mobilen Endgerät und DiversityMobile erhebt. Dieser besteht auf der organisatorischen Ebene nur durch seine Person. In der funktionalen Ebene verfügt dieser über den Datenbestand, den er erhebt und auf der operationalen Ebene benötigt diese ein Mobilgerät,auf dem DiversityMobile installiert ist.

Die Aufteilung in Ebenen ermöglicht es auch es eine Ebene einer Infrastruktur isoliert zu betrachten und so wichtige Elemente separat darzustellen. In Abbildung 7.2 ist die organisatorische Ebene von GBIF schematisch dargestellt. Eine detaillierter Erfassung der organisatorischen Ebene könnte über ein vollständiges Organigramm erfolgen. In Abbildung 7.2 ist dargestellt, dass GBIF über das GBIF-Board zentral gesteuert wird. Diesem sind die regionalen GBIF-Organisationen untergeordnet, welche nach Ländern aufgeteilt sind. Die direkte Unterorganisation von GBIF für Deutschland ist z.B. GBIF-D, welche für die direkte Kommunikation mit GBIF zuständig ist [84]. Diesen sind wiederum in Knoten und Institute untergliedert, die sich die Aufgaben innerhalb eines Landes selbständig aufteilen. Aus den Knoten wird auch das Führung für die Ländervertretungen bestimmt. So besteht GBIF-D aus den acht deutschen GBIF-Knoten und die Koordinatoren der Knoten bilden das Leitungsgremium von GBIF-D [84]. Einzelne Wissenschaftler können sich nicht direkt dem GBIF-Board oder einer Länderorganisation anschließen sondern partizipieren indirekt über den Anschluss an einen Knoten.

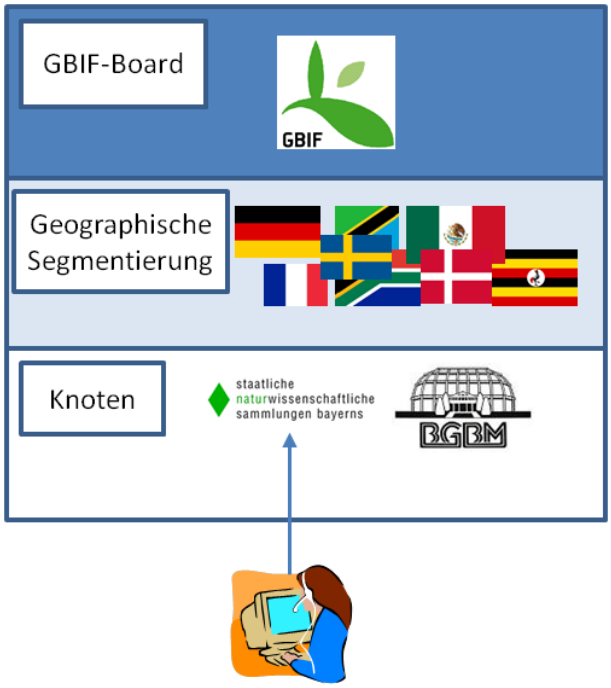

Abbildung 7.2: Aufbau der organisatorischer Ebene einer Infrastruktur am Beispiel von GBIF

Kommt es hingegen auf eine Repräsentation der technischen Infrastruktur an, ist die organisatorischen Ebene nicht von Interesse und es kann eine reine Betrachtung der operationalen Ebene erfolgen (siehe Abbildung 7.3). In dieser Abbildung ist die reine technische Infrastruktur von GBIF dargestellt. In dieser technischen Betrachtung wird deutlich, dass Datenanbieter in GBIF entweder über das 'Intergrated Publishing Toolkit' (IPT) oder andere Protokolle wie das 'TDWG Access Protocol for Information Retrieval' (TAPiR) angeschlossen werden. Die Art des Protokolls determiniert dabei den Datenfluss bis zur Publikation der Daten im GBIF-Portal. Somit wird in Abbildung 7.3 ausschließlich die operationale Ebene visualisiert. Die organisatorische und die funktionale Ebene spielt in dieser Abbildung keine Rolle. Damit kann über die Ebenen einer Infrastruktur die Infrastruktur thematisch isoliert analysiert werden.

Durch die Betrachtung in Ebenen können Teilaspekte einer Infrastruktur dargestellt werden. Die Aufteilung in Ebenen ermöglicht somit eine zielgerichtete Analyse einer Infrastruktur. Dies ist eine notwendige Voraussetzung für die Identifikation von Kriterien zur Evaluation einer Infrastruktur.

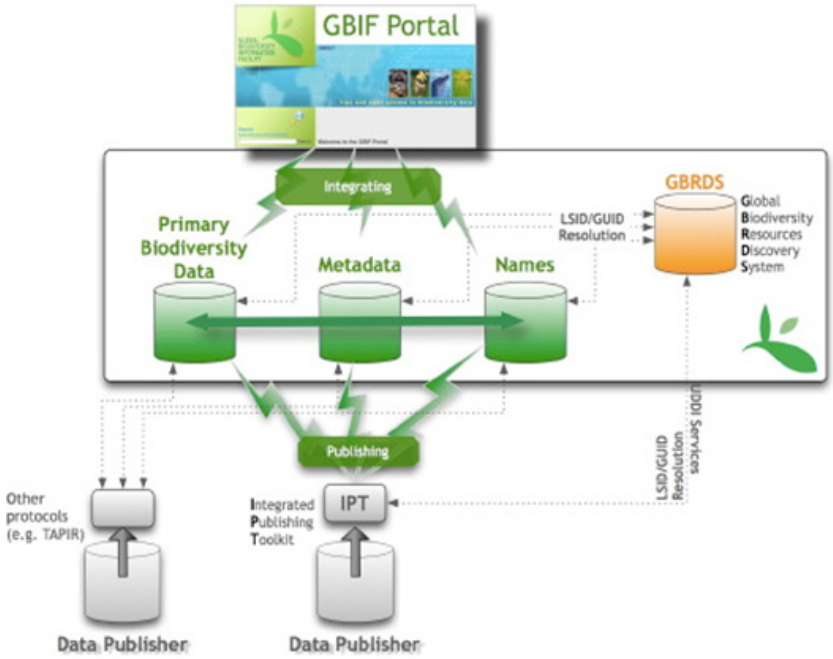

Abbildung 7.3: Aufbau der operationalen Ebene einer Infrastruktur am Beispiel des GBIF-Portals nach [81]

Operationale Ebene

In der operationale Ebene einer Infrastruktur werden die technischen Komponenten des Datenaustauschs analysiert. Sie umfasst die gesamte Hardware, sowie Software und Services, die zum Datenaustausch erforderlich sind. In dieser Ebene werden ausschließlich technische Gesichtspunkte betrachtet. Ein Algorithmus, der bei der Erstellung eines Schemamappings unterstützt, ist Teil dieser Ebene – das Schemamapping an sich nicht. Dieses ist in der funktionalen Ebene angesiedelt. Genauso ist die Hardware von Datenbanken in der operationalen Ebene enthalten, wohingegen der Datenbestand an sich der funktionalen Ebene zugeordnet ist. Analog dazu ist die Software zur Distribution von Daten Teil der operationalen Ebene, wohingegen die betreibende Organisation ein Element der organisatorischen Ebene ist und die publizierten Daten Teil der funktionalen Ebene sind.

Organisatorische Ebene

In einem Netzwerk kommunizieren verschiedene Teilnehmer mit unterschiedlichen Interessen. Dies kann aber nicht allein auf technischer Ebene erfolgen, insbesondere wenn die Teilnehmer des Netzwerkes über eine gewisse Autonomie verfügen. Dementsprechend müssen Eigentümer von Datenquellen und Entscheidungsträger in

Organisationen, die einen Datenspeicher verwalten, von der Teilnahme an einer Infrastruktur überzeugt werden. Genauso muss eine Infrastruktur in der Community der Anwendungsdomäne akzeptiert sein, da dieses sonst nicht verwendet und mit Daten versorgt wird. All diese Themen werden auf der organisatorischen Ebene einer Infrastruktur betrachtet. Innerhalb der organisatorischen Ebenen werden rechtliche Themen behandelt, welche die Zusammenarbeit der Organe einer Infrastruktur betreffen. So ist beispielsweise das Entsenden von Abgeordneten eines Instituts auf eine höhere Ebene der Infrastruktur oder die Rechte der Teilnehmer einer Infrastruktur Teil der organisatorischen Ebene. Die Eigentümerschaft an Daten bezieht sich allerdings auf den Datenbestand an sich und ist somit der funktionalen Ebene zugeordnet.

Ein wesentliches Element auf dieser Ebene ist eine zentrale, koordinierende Stelle. Aufgabe dieser ist es, Standards und Protokolle zu spezifizieren und Software zum Datenaustausch zu Verfügung zu stellen. Auch wenn es prinzipiell wünschenswert wäre, auf diese in einem Netzwerk von gleichberechtigten Teilnehmern zu verzichten, können diese Aufgaben in der Praxis nicht der Selbstorganisation der einzelnen Teilnehmer überlassen werden, da es ohne Moderation von einer zentralen Stelle unmöglich wäre mit einer theoretisch unbegrenzten Anzahl an Teilnehmern eine konsensfähige Lösung zu finden. Darüber hinaus kann eine zentrale Koordinationstelle durch ihre Entscheidungen die Akzeptanz in der Community erhöhen und so eine Vertrauensbasis schaffen.

Funktionale Ebene

Die Datenanbieter innerhalb eines Netzwerk müssen zudem unabhängig voneinander arbeiten können. Sie bilden eine gemeinsame Infrastruktur, weil sie ein bestimmtes, gemeinsames Ziel verfolgen, der meistens im Austausch von Daten einer bestimmten Domäne liegt. Über den Anwendungsbereich einer Infrastruktur muss zwischen den Teilnehmern Einigkeit herrschen. Dies wird in der funktionalen Ebene festgelegt. Kernpunkt ist ein gemeinsames Verständnis über die Art der auszutauschenden Daten. Dabei ist unter Art zum einen das globale Schema für den Datenaustausch zu verstehen, zum anderen aber auch ein grundlegendes Verständnis über die Grenzen der Anwendungsdomäne. So ist es z.B. für eine Infrastruktur in der Biodiversitätsinformatik notwendig, sich von benachbarten Disziplinen wie der Bioinformatik abzugrenzen. Folglich sollen in einem Austauschschema der Biodiversitätsinformatik die speziellen Belange der Bioinformatik wie Genetik nicht berücksichtigt werden. Es wird festgelegt, welche Daten über die Infrastruktur übertragen werden können

und welche nicht. Damit ist das Austauschschema Teil der funktionalen Ebene. Darüber hinaus sind die Datenbestände an sich Teil dieser Ebene zugeordnet. Dabei ist zu beachten, dass unter dem Datenbestand nicht nur Daten aus relationalen Daten oder in einem XML-Schema verstanden werden, sondern auch Multimediadaten wie Bilder oder Videoaufzeichnungen. Dabei ist auf der funktionalen Ebene festgelegt, wer Urheber dieser Daten ist und wer Zugriffsrechte auf die Daten hat.

7.3 Evaluation von Infrastrukturen

Das Ziel der Evaluation einer Dateninfrastruktur ist festzustellen, ob sie für ihren Anwendungszweck gut oder schlecht geeignet ist. So kann eine Infrastruktur für einen spezifischen Anwendungszweck sehr gut geeignet sein – wohingehend dieselbe Infrastruktur in einem anderen Fall völlig ungeeignet ist. Folglich ist die Analyse des Anwendungszwecks ein zwingendes Element des Evaluationsframeworks IEF, welches in Abschnitt 7.3.2 vorgestellt wird.

Für IEF müssen verschiedene Kriterien identifiziert werden, nach denen eine Infrastruktur analysiert werden kann. Infrastrukturen sind im Allgemeinen große Strukturen mit vielen Teilnehmern und für eine spezielle Aufgabe konzipiert. Damit hat eine Infrastruktur einen stark individuellen Charakter, der die Entwicklung eines generischen Evaluationsframeworks erschwert. Trotzdem lassen sich allgemeine Kriterien für Infrastrukturen aufstellen (siehe Abschnitt 7.3.1), welche den verschiedenen ebenen zugeordnet werden können. Diese werden in Abschnitt 7.3.2 zu dem generischen Framework IEF zusammengefasst. Die Evaluation kann vor einem bestimmten Anwendungshintergrund genauer erfolgen, da die speziellen Anforderungen der Anwendungsdomäne dabei berücksichtigt werden. Deswegen wird IEF in Abschnitt 7.3.3 zu IEF-Biodiv ausgebaut und berücksichtigt damit den speziellen Anwendungshintergrund der Biodiversitätsinformatik. Die Konzeption des generischen Evaluationsframework IEF und der domänenspezifischen Anpassung IEF-Biodiv stellen neue Entwicklungen dar und ermöglichen erstmals den nach Ebenen strukturierten Vergleich von Infrastrukturen vor einem spezifischen Anwendungshintergrund.

7.3.1 Kriterien

Im folgenden Abschnitt werden Kriterien zur Evaluation einer Infrastruktur vorgestellt. Die Sammlung der Kriterien erfolgt dabei nicht frei, sondern über die Zuordnung der Kriterien zu den Ebenen (siehe Abschnitt 7.2). Auf diese Weise lässt sich eine Infrastruktur nach isolierten Kriterien für die operationale, organisatorische und

funktionale Ebene evaluieren.

So kann es möglich sein, dass in einer Infrastruktur zwar die technischen Voraussetzungen gut an den Anwendungshintergrund angepasst sind, das Austauschschema aber die Anforderungen des Anwendungsbereichs nur unzureichend ist oder aber die organisatorische Struktur für den Anwendungszweck nicht gut geeignet ist. Die Evaluation einer Infrastruktur erfordert eine genaue Kenntnis der Anwendungsdomäne und der Interessen der Stakeholder einer Infrastruktur. Dabei ist zu beachten, dass die notwendigen Kenntnisse zur Evaluation einer Infrastruktur teilweise nicht veröffentlicht sind. Eine Infrastruktur kann deshalb für einen Außenstehenden auch nur soweit beurteilt werden, wie die für die Evaluation notwendigen Eigenschaften publiziert sind.

Kriterien der operationalen Ebene

Kriterien der operationalen Ebene bewerten die Hardware, Software und baulichen Einrichtungen einer Infrastruktur. Teil dieser Kriterien ist damit auch die grundlegende Struktur eine Infrastruktur, die aber nicht per se als gut oder als schlecht bezeichnet werden kann. Da in Abschnitt 6.1 primär die technische Sicht der Informationsintegration betrachtet wurden, sind die in diesem Abschnitt vorgestellten Merkmale primär der operationalen Ebene zuzuordnen.

Folgende Kriterien konnten für die operationale Ebene identifiziert werden:

- **Infrastrukturtyp**: Charakterisierung, ob es sich um eine offene oder geschlossenen Infrastruktur handelt
- **technische Partizipation**: Möglichkeit von potentiellen Datenanbietern und Datenkonsumenten Teil der technischen Infrastruktur zu werden
- **Integrationsart**: Beinhaltet, ob über die Infrastruktur materialisierte oder virtuelle Informationsintegration oder aber beides möglich ist
- **Zentralität**: Ausrichtung der Datenspeicherung auf einen zentralen Datenspeicher oder verteilte Datenspeicherung
- **technische Realisierung**: Umfasst wie der Datenaustausch auf technischer Ebene realisiert ist und welche Technologie zum Einsatz kommt z.B. verteilte Datenbanken, objektorientierte Middleware oder ein anderer Ansatz (siehe Abschnitt 6.2)
- **Software- und Servicequalität**: Misst die Nutzbarkeit und Qualität der Softwarekomponenten, die gemeinsam genutzt werden

- **Autonomie**: Autonomie der Teilnehmer im Sinne von Abschnitt 6.1, welche innerhalb einer Infrastruktur zulässig ist
- **Heterogenität**: Heterogenität im Sinne von Abschnitt 6.1, welche innerhalb einer Infrastruktur möglich ist

Das erste Kriterium ist der **Infrastrukturtyp**. Dieser kann entweder *offen* oder *geschlossen* sein. In einer geschlossen Infrastruktur sind die teilnehmenden Datenspeicher zum Zeitpunkt des Designs der Infrastruktur bekannt und ändern sich im Betrieb der Infrastruktur nicht oder nur kaum. Dies ermöglicht es, eine spezialisierte Architektur anzuwenden und führt im Allgemeinen dazu, dass bei der Informationsintegration der Grad der Homogenität (siehe Abschnitt 6.1) höher ist. Bei einer offenen Infrastruktur ist das Design dieser Infrastruktur im Gegensatz dazu darauf ausgelegt, jederzeit einen neuen Datenspeicher zu integrieren. Es können also nur geringe Annahmen über die Homogenität der Datenspeicher in das Design aufgenommen werden, welche harte Anforderungen für die Aufnahme einer Datenquelle in die Infrastruktur darstellen.

Eine weitere Eigenschaft ist die **Partizipationsmöglichkeit**, welche *direkt* oder *indirekt* ist. Bei direkter Partizipation wird ein Datenanbieter vollständig in die Infrastruktur integriert und ermöglicht direkten Zugriff auf seinen Datenbestand. Bei indirekter Partizipation wird ein neuer Teilnehmer über einen Datenspeicher angebunden, der bereits Teil der Infrastruktur ist. Alle Daten dieses Teilnehmers können nur indirekt über den angebundenen Datenspeicher in die Infrastruktur gelangen. Dies ist z.B. regelmäßig bei Einzelpersonen der Fall, die sich zunächst an ein Repositorium anschließen müssen. Bei der Evaluation muss dementsprechend auch nach Partizipationsmöglichkeiten bezüglich der potentiellen Teilnehmergruppen unterschieden werden.

Das nächste Kriterium ist die **Zentralität** einer Infrastruktur. Eine Infrastruktur kann *zentral*, *dezentral* oder in Mischformen organisiert sein. In einem dezentralen Design tauschen Teilnehmer direkt miteinander Daten aus, wohingegen in einem zentralen Design der Datenaustausch über einen zentralen Datenspeicher erfolgt, mit dem alle Teilnehmer kommunizieren. Auch in dezentralen Architekturen wird aber im Allgemeinen eine zentrale, koordinierende Stelle existieren, welche grundlegende Spezifikationen (Austauschformat, Austauschprotokoll,...) trifft und möglicherweise Services zur Verfügung stellt.

Die **technische Realisierung** sowie die **Heterogenität** und **Autonomie** beziehen sich auf die in Kapitel 6 vorgestellten Merkmale der Informationsintegration. Ein weiteres Kriterium ist die **Qualität der gemeinsam genutzten Software**

und Services. Hierzu soll kein eigenes Framework entworfen werden, weil die Messung von Softwarequalität eine eigenständige Disziplin ist, für die mit [112] eigene Standards existieren. Der interessierte Leser sei hierzu auf [141] verwiesen.

Kriterien der organisatorischen Ebene

Für die organisatorische Ebene sind insbesondere Kriterien aus dem sozialen Bereich wichtig, da diese die Akzeptanz der Infrastruktur in der Anwendungsdomäne maßgeblich bestimmen. Sind diese Kriterien nicht erfüllt, wird die Infrastruktur von den Nutzern in der Anwendungsdomäne nicht angenommen und die Infrastruktur wird nicht genutzt. Dadurch ist verfügbare Datenbestand in einer solchen Infrastruktur geringer als in einer akzeptierten Infrastruktur. Dabei ist von Interesse, wie viel Aufwand eine Teilnehmer betreiben muss, um an der Infrastruktur teilnehmen zu können. Dies beinhaltet zum einen den personellen Aufwand, der betrieben werden muss, um die technischen Voraussetzungen zu erfüllen – zum anderen auch den Aufwand, der für die organisatorische Partizipation von Teilnehmern betrieben werden muss. Hierbei sind der Anwendungszweck und die persönlichen Interessen der Teilnehmer von Bedeutung und bilden den Kontext der Evaluation.

Kriterien auf der organisatorischen Ebene sind damit im Folgenden:

- **Zitierfähigkeit** der Daten: Bedeutung der Infrastruktur für die eigene Arbeit und Ruf der Infrastruktur für Veröffentlichungen von eigenen Daten in Fachzeitschriften
- **organisatorische Partizipation**: Möglichkeit von potentiellen Datenanbietern und -konsumenten an Entscheidungen teilzuhaben
- **Ressourcenaufwand**: Aufwand den ein Teilnehmer der Infrastruktur z.B. für Softwareinstallation, Trainingsangebote oder Teilnahme an Gremien leisten muss
- **Ressourcenangebot**: Beinhaltet wie und in welcher Form die Infrastruktur personelle Ressourcen für die Teilnehmer wie Schulungen und Beratung in Wikis zur Verfügung stellt

Mit der **organisatorischen Partizipation** ist gemeint, in welcher Form und ob die Infrastruktur neue Mitglieder in die Organisationsstruktur aufnimmt und an Entscheidungsprozessen beteiligt. Die Teilnahme an einer Infrastruktur kann auch an gewisse Voraussetzungen geknüpft sein. So kann die organisatorische Teilnahme an einer Infrastruktur ausschließlich Institutionen vorbehalten sein und Privatpersonen

von Entscheidungsprozessen ausschließen (siehe z.B. GBIF in Abschnitt 7.4.1). Auf dieser Ebene ist auch relevant, ob für Teilnehmer geeignete Schulungsmaßnahmen existieren. Diese werden im Allgemeinen von einer zentralen Koordinationsstelle oder aber auch von speziellen Teilnehmern angeboten, die in der Organisationsstruktur eine spezielle Funktion ausüben.

Kriterien der funktionalen Ebene

In der funktionalen Ebene einer Infrastruktur ist spezifiziert, welche Inhalte in den Daten über die Infrastruktur übertragen werden können. Sie ist damit das zentrale Element der Evaluation einer Infrastruktur. Wenn die Kriterien der funktionalen Ebene nicht erfüllt sind, ist der Datenaustausch über die Infrastruktur nur unzureichend möglich. Ein weiteres Kriterium ist der Datenbestand an sich. Das heißt, inwieweit eine Infrastruktur relevante Daten für die Teilnehmer liefern kann.

Dabei ist von Interesse, welches Austauschschema eine Infrastruktur verwendet. Das Austauschschema bildet die Basis der Integration und ist die primäre Ursache für Einschränkungen bei der Informationsintegration. Es kann zwischen verschiedene Datenarten unterscheiden werden. Neben den wissenschaftlichen Primärdaten werden über Infrastrukturen personen- und projektbezogene Daten ausgetauscht. Diese dienen der Beschreibung und werden als 'Metadaten' bezeichnet[1]. Eine weitere Art von Daten sind Multimediadaten, deren Austausch in Infrastrukturen spezielle Berücksichtigung erfordert. Die Kriterien der funktionalen Ebene sind:

- **integrierte Datenarten**: Unterstützung der Integration von Primär-, Meta- oder Multimediadaten
- unterstützte **Datenstandards und Austauschschemata**
- **Datenverluste**: Stark von den Austauschschemata und den Möglichkeiten zum Transport von Multimediadaten abhängig
- **Identitätssicherung**: Ein Datensatz muss in einem Netzwerk eindeutig gekennzeichnet sein
- **Aktualisierbarkeit**: Die Infrastruktur muss es ermöglichen, Aktualisierungen in Datenspeichern zu übertragen

[1] Der Begriff 'Metadaten' ist ungünstig gewählt, da kein Bezug zur Metamodellierung (siehe Abschnitt 5.2) besteht. Er soll aber auch im Rahmen dieser Arbeit verwendet werden, da dies der allgemeine Sprachregelung im Kontext der Biodiversitätsinformatik entspricht.

- **Vergleichbarkeit**: Vergleichbarkeit der Bedeutung der über die Infrastruktur angebotenen Daten
- **Data Provenance**: Speicherung der Herkunft der Daten und der mit Daten ausgeführten Transformationen.

Dabei ist vor allem die **verlustfreie Datenübertragung** von Bedeutung, was primär von dem Austauschschema und den Möglichkeiten Multimediadaten zu übertragen abhängt. Das Austauschschema wird hierbei im Kontext des Anwendungshintergrunds evaluiert. Das ABCD-Format mag z.B. für den Datenaustausch in der Biodiversitätsinformatik geeignet sein (vgl. Abschnitt 4.4.1). Daten aus dem Finanzwesen können über ABCD aber nicht übertragen werden. Es können selbstverständlich nicht alle individuellen Anforderungen erfüllt werden. Der Anwendungshintergrund stellt vielmehr einen Kompromiss unter den Teilnehmern über den kleinsten gemeinsamen Nenner dar. Die Qualität des Austauschschemas erfolgt im Bezug auf diesen Anwendungshintergrund nach den Kriterien aus Kapitel 4. Diesbezüglich ist zu beachten, dass in einer Infrastruktur mit mehreren Austauschschritten auch mehr als ein Datenaustauschschema verwendet werden kann. Ein Datenverlust kann aber, wie in Abbildung 7.4 gezeigt wird, bei jeder Datenübertragung auftreten. Die erste Quelle für Datenverluste ist dabei, dass die Primärdatenaufnahme in der Natur nur unzureichend vom Medium der primären digitalen Speicherung unterstützt wird. Es können in diesem Szenario nicht alle erforderlichen Sachverhalte in digitaler Form gespeichert werden und es findet bereits an dieser Stelle ein Datenverlust statt. Anschließend werden die Daten zweimal über Wrapper übertragen: Das erste Mal vom Medium der primären Erfassung in ein institutionelles Repositorium. Das zweite Mal vom institutionellen Repositorium in das globale Austauschschema eine Infrastruktur. An jedem Schemaübergang besteht dabei die Gefahr von Datenverlusten.

Für die Messung der Qualität ist damit nicht nur der Datenverlust in jedem einzelnen Austauschschritt relevant, sondern auch welche Daten bei einer Übertragung über eine ganze Kette von Integrationsschritten (siehe Abbildung 7.5) verloren gehen. Dabei ist von besonderen Interesse, ob in der Infrastruktur Vorkehrungen getroffen wurden, die es erlauben Datenverluste über die Infrastruktur an einer anderen Stelle auszugleichen, so dass eine Infrastruktur eine gewisse Robustheit gegenüber Datenverlusten bietet (siehe Aktualität und Data Provenance). Probleme bezüglich der Aktualität von Daten können dabei ausschließlich bei materialisierter Informationsintegration auftreten.

Als Beispiel hierfür kann die Rolle des ABCD-Standards im GBIF-Netzwerk herangezogen werden. Auch wenn ABCD explizit für Beobachtungsdaten in der Biodi-

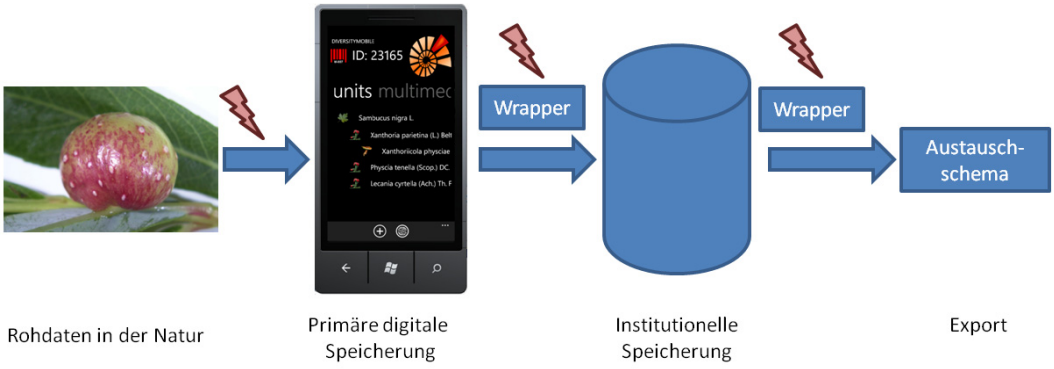

Abbildung 7.4: Potentieller Datenverlust bei Datenübertragung in einer Infrastruktur

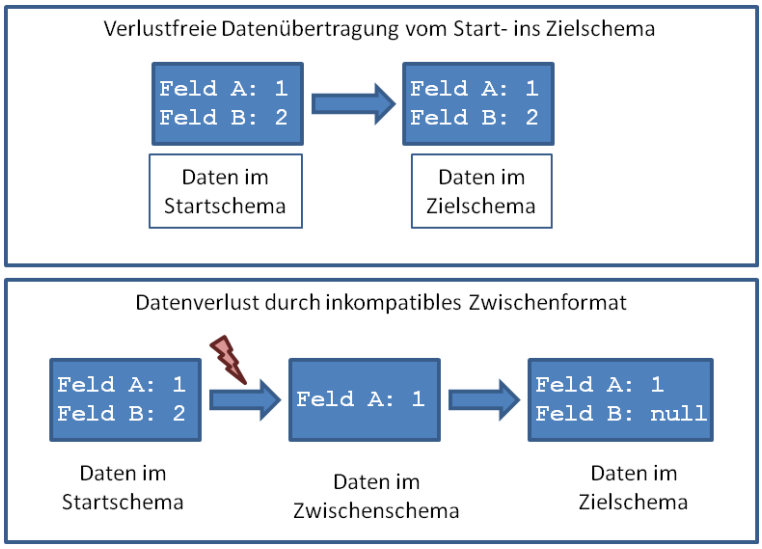

Abbildung 7.5: Datenverlust bei Kompatibilität von Start- und Zielschema durch ein inkompatibles Zwischenschema

versitätsforschung entwickelt wurde, kann daraus noch nicht geschlossen werden, dass alle Sachverhalte dieser Domäne in ABCD adäquat abgebildet werden können. Dementsprechend ist es möglich, dass ein Wissenschaftler nicht alle Aussagen, die er über sein Forschungsvorhaben abbilden möchte, in ABCD abbilden kann und es entstehen Datenverluste. Arbeitet der Wissenschaftler nicht direkt mit ABCD, sondern exportiert seine Daten über eine Datenbank, die an das GBIF-Netzwerk angeschlossen ist, nach ABCD, finden mehrere Übergänge statt. Zunächst muss der Wissenschaftler seine Rohdaten in das Schema der Datenbank übertragen, was die erste Quelle für einen Datenverlust darstellt. Anschließend werden die Daten über einen Wrapper in das ABCD-Format transformiert. Dabei kann das Mapping des lokalen Datenschemas nach ABCD selbst unzureichend sein, so dass mit der Konfiguration des Wrappers eine weitere Verlustquelle gegeben ist. Zuletzt kann der Fall eintreten, dass das lokale Datenschema nicht vollständig von ABCD erfasst werden kann. Damit sind in dieser Beispielinfrastruktur bereits drei Quellen für Datenverluste identifiziert. Hierbei ist zu beachten, dass in der beschriebenen Infrastruktur selbst dann Datenverluste auftreten können, wenn die Primärdaten vollständig mit ABCD kompatibel sind, da bei der Übertragung in das lokale Datenschema und von diesem nach ABCD Datenverluste auftreten können (siehe Abbildung 7.5).

Ein weiteres wichtiges Qualitätskriterium der funktionalen Ebene ist die Unterstützung von **Data Provenance**, da mit der Veränderung eines Datensatzes auch eine Veränderung der Bedeutung der Daten verbunden ist. Dies ist relevant bei der kuratorischen Pflege von Daten in einer Infrastruktur, bei welcher die Originaldaten verändert werden. Durch Data Provenance wird zum einen die Urheberschaft an den Daten dokumentiert, zum anderen kann stets der Originaldatensatz wiederhergestellt werden. Die Grundlagen für die Unterstützung von Data Provenance sind Eigenschaften eines Datenstandards und werden in diesem Kontext bewertet (siehe Abschnitt 4.3). Bei der Evaluation der Unterstützung von Data Provenance für Infrastrukturen wird ermittelt, ob die Möglichkeiten des Datenstandards tatsächlich genutzt werden.

In engen Zusammenhang mit Data Provenance in einer Infrastruktur stehen die Aufgaben, die die **Aktualisierung** von Datensätzen ermöglichen und die **Identität** eines Datensatzes garantieren. So kann durch den Austausch desselben Datensatzes in einem Netzwerk dieser in verschiedenen Datenbanken materialisiert integriert werden. Wenn eine dritte Datenbank mit den ersten beiden Datenbanken kommuniziert, muss gewährleistet sein, dass der Datensatz nur einmal kopiert wird. In Abbildung 7.6 wird die Duplikation eines Datensatzes in einem Netzwerk bei materialisierter In-

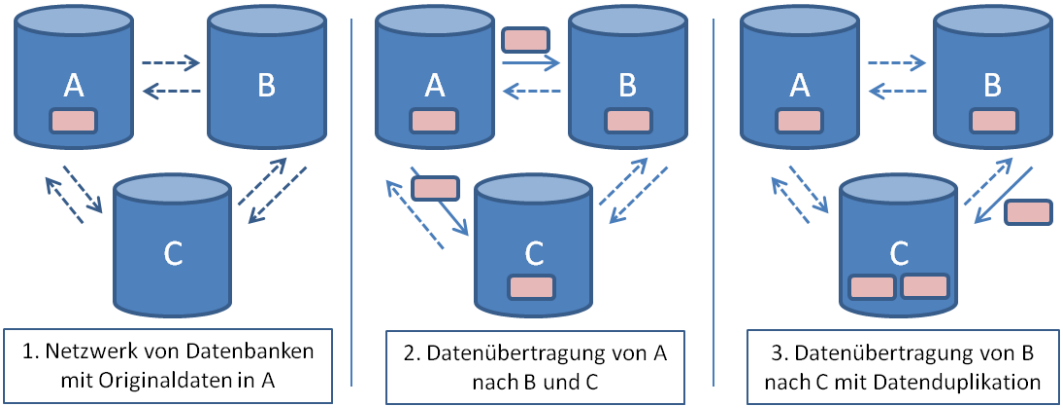

Abbildung 7.6: Identitätsproblem in Netzwerken

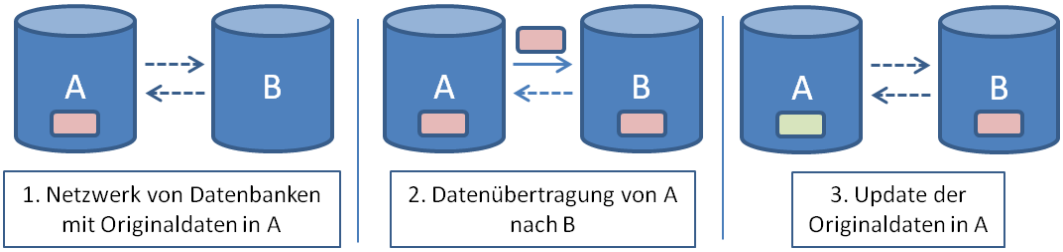

Abbildung 7.7: Aktualitätsproblem in Infrastrukturen

tegration beschrieben. Ausgehend von Datenbank A wird derselbe Datensatz an die Datenbanken B und C übertragen. Anschließend wird ein Datensatz von Datenbank B nach Datenbank C übertragen. Da der Datensatz im Netzwerk nicht eindeutig gekennzeichnet ist, kann dieser nicht als bekannt identifiziert werden und es entsteht ein Duplikat des Datensatzes in Datenbank C. Um das zu vermeiden, ist eine eindeutige Kennzeichnung eines Datensatzes für das gesamte Netzwerk erforderlich.

Ein anderes Problem liegt in der Garantie der **Aktualität von Daten bei materialisierter Informationsintegration**. In diesem Fall wird ein Originaldatensatz an seinem ursprünglichen Speicherort verändert, nachdem dieser an einen anderen Datenspeicher des Netzwerks übertragen wird (siehe Abbildung 7.7). Die Infrastruktur muss nun ermöglichen, diesen Datensatz auch in allen anderen Datenbanken zu aktualisiert. Ob eine Aktualisierung erwünscht ist, hängt von der Art der Daten und dem Anwendungshintergrund ab. Entsprechende Datenaktualisierungen müssen darüber hinaus über Data Provenance verfolgt werden.

Unter der Vergleichbarkeit der Daten ist die **Homogenität** der integrierten Daten zu verstehen. Durch eine Vielzahl von optionalen Feldern wie z.B. in ABCD können sehr verschiedenartige Daten über eine Infrastruktur ausgetauscht werden.

Allerdings ist dieser Umstand auch die Ursache dafür, dass das Austauschschema so allgemein definiert ist, dass die Daten, die in denselben Feldern gespeichert werden, über eine unterschiedliche Bedeutung verfügen. Besonders deutlich tritt dieser Sachverhalt im LTER-Netzwerk (siehe Abschnitt 7.4.3) hervor. Im LTER-Netzwerk sind nur Metadaten vergleichbar. Die wissenschaftlichen Primärdaten werden in einer beliebigen Struktur nach Wahl des Datenanbieters gespeichert (im Allgemeinen Excel- oder CSV-Format mit eigenem Schema) und können nicht verglichen werden.

7.3.2 Die Grundstruktur von IEF

In diesem Abschnitt wird die Grundstruktur einer Evaluation mit dem generischen 'Infrastracture Evaluation Framework' (IEF) vorgestellt. Dabei ist zu beachten, dass aufgrund des individuellen Charakters von Infrastrukturen nicht alle Kriterien bei der Evaluation einer Infrastruktur angewendet werden, sondern diese vor dem Anwendungshintergrund ausgewählt werden. Die Kriterien aus Abschnitt 7.3.1 stellen vielmehr eine Sammlung an Kriterien dar, aus denen – wie aus einem Baukasten – in Schritt 2 von IEF eine Evaluationsstruktur gebildet wird. Dies ist Teil der domänenspezifischen Anpassung von IEF.

Die generische Evaluation von Infrastrukturen mit IEF verfügt damit folgende Struktur:

1. Analyse des Anwendungshintergrunds
2. Kriterienauswahl (domänenspezifische Anpassung)
3. Ausführung der Evaluation nach dem Evaluationsschema
4. Fazit

Der wichtigste Schritt ist aber zunächst die Analyse des **Anwendungsbereichs** einer Infrastruktur, da viele der Kriterien nur vor einem spezifischen Anwendungszweck evaluiert werden können und eine Evaluation dieser Kriterien auch nur in bestimmten Anwendungsszenarien notwendig ist. Innerhalb der Analyse des Anwendungshintergrunds sind die Identifikation des Teilnehmerkreises und ihrer Interessen ein zwingender Bestandteil. Auf Grundlage dieser können die relevanten **Kriterien** aus Abschnitt 7.3.1 strukturiert nach Ebenen identifiziert werden. Dabei kann es erforderlich sein, in der Evaluation nach verschiedenen Teilnehmergruppen zu unterscheiden. So können die Interessen einer Institution in einer Infrastruktur in anderer

Weise repräsentiert sein, als die einer Einzelperson. Das **angepasste Evaluationsschema** wird anschließend zur Bewertung einer speziellen Infrastruktur herangezogen und die Ergebnisse in einem **Fazit** zusammengefasst.

7.3.3 Evaluation von Infrastrukturen mit IEF-Biodiv

Im folgenden Abschnitt wird das allgemeine Framework IEF zur Evaluation in Biodiversitätsinformatik angepasst. Dazu wird zunächst Schritt 1 von IEF ausgeführt und der Anwendungshintergrund analysiert. Auf Basis dieses Anwendungshintergrund wird in Schritt 2 von IEF mit der domänenspezifischen Kriterienauswahl IEF-Biodiv erstellt. IEF-Biodiv wird für die Evaluation von Infrastrukturen in Abschnitt 7.4 verwendet. Die ist gleichzeitig Schritt 3 in der Anwendung von IEF. Das Fazit der Evaluation ist in Abschnitt 7.5 beschrieben und stellt damit den Abschluss der Analyse der Biodiversitätsinformatik mit IEF dar.

Analyse des Anwendungshintergrunds

Der Anwendungsbereich von IEF-Biodiv ist die Evaluation von Infrastrukturen zum Datenaustausch in der Biodiversitätsinformatik. Dabei lassen sich verschiedene Teilnehmergruppen unterscheiden:

- **Institutionelle Repositorien**
- **Wissenschaftler**
- **Koordinationsstelle** der Infrastruktur

Aufgrund der überragenden Bedeutung dieser Datenstandards in der Biodiversitätsinformatik haben alle diese Teilnehmergruppen ein Interesse an der Unterstützung von ABCD und DwC als Datenstandards.

Institutionelle Repositorien treten primär als Datenanbieter auf– möchten aber auch über materialisierte Integration die eigenen Datenbestände vermehren. Primäres Interesse der Repositorien ist der qualitativ hochwertige Datenaustausch bei gleichzeitiger Wahrung der Eigentümerschaft der Daten. Für die Qualitätssicherung des Datenaustauschs sind alle Kriterien der funktionalen Ebene – insbesondere aber das Kriterium der Datenverluste – von Interesse. Darüber hinaus haben die Repositorien ein Interesse an einem moderaten Ressourcenaufwand und wünschen prinzipiell eine offene Infrastruktur, damit ihnen die Teilnahme ermöglicht wird. Da Repositorien ein weites Spektrum an Aufgaben erfüllen müssen, ist die Teilnahme an der Infrastruktur eine von vielen Aufgaben eines Repositoriums. Dementsprechend

müssen Repositorien soweit eigenständig arbeiten können, dass diese in der Erfüllung dieser Aufgaben nicht eingeschränkt werden. Somit benötigen diese einen hohen Grad an Designautonomie im Bezug auf die interne Gestaltung ihres Datenspeichers. Da der freie Datenaustausch nicht immer erwünscht ist, benötigen Repositorien Zugriffsautonomie und juristische Autonomie. Schnittstellenautonomie wird im Allgemeinen nicht benötigt, weil Repositorien keine eigene Software zur Teilnahme an der Infrastruktur entwickeln. Deshalb ist für diese auch die organisatorische Unterstützung durch Schulungen und Informationsmaterial besonders wichtig.

Wissenschaftler sind Einzelpersonen oder Forschergruppen. Diese möchten keine eigenen Hardware oder Software betreiben und möchten so wenig Aufwand wie möglich zur Partizipation an einer Infrastruktur aufbringen. Das Interesse dieser Teilnehmergruppe liegt hingegen primär in der langfristigen Archivierung und der Garantie der Zitierfähigkeit ihrer Daten. Dementsprechend liegt ein Interesse an materialisierter Informationsintegration in die Datenbestände der Repositorien. Für die langfristige Archivierung wird insbesondere eine verlustfreie Datenübertragung benötigt. Des Weiteren benötigen diese Teilnehmer eine Partizipationsmöglichkeit an der Infrastruktur. Aus Gründen des personellen Aufwands wird eine indirekte Partizipation über ein Repositorium zu bevorzugt. Ein weiteres Interesse von Wissenschaftlern ist die Verfügbarkeit von Forschungsdaten aus anderen Forschergruppen und die Vergleichbarkeit dieser Daten.

In Infrastrukturen in der Biodiversitätsinformatik ist im Allgemeinen eine **zentrale Koordinationsstelle** und zum Ausgleich der Interessen integriert, welche auch über eigene Ziele verfügt. Dabei steht neben der Qualität des Datenaustauschs auch die Vergleichbarkeit der Daten im Fokus. Dies wird häufig über eine globale Sicht der Daten mit einem globalen Austauschschema realisiert. Da die zentrale Koordinationsstelle auch für die Auswahl und Programmierung gemeinsam genutzter Software verantwortlich ist, ist ein geringer Grad an Heterogenität der Teilnehmer wünschenswert. Ist die Heterogenität unter den potentiellen Teilnehmern zu groß, kann die zentrale Stelle durch Entscheidungen bestimmte Teilnehmer ausschließen z.B. durch Beschränkung auf Datenspeicher, die auf relationalen Datenbanken beruhen. Allerdings strebt die zentrale Koordinationsstelle ein möglichst reichhaltiges Datenangebot an, um die Attraktivität der Infrastruktur zu steigern.

Die Interessen der drei Teilnehmergruppen an einzelnen Kriterien ist nach Ebenen getrennt in den Tabellen 7.1 bis 7.3 dargestellt. Wenn eine Teilnehmergruppe keine Präferenz im Bezug auf ein Kriterium hat, ist dies durch einen '−' gekennzeichnet. Die Werte in den Tabellen geben die Wunschwerte nach Teilnehmergruppen wieder.

Operationale Kriterien	Zentrale Koordinationsstelle	Repositorien	Wissenschaftler
Op1: Infrastrukturtyp	offen	offen	offen
Op2: Integrationsart	materiell	virtuell	materiell
Op3: technische Realisierung	individuell	individuell	-
Op4: Zentralität	zentral	dezentral	-
Op5: technische Partizipation	direkt	direkti	ndirekt
Op6: Autonomie	gering	hoch	-
Op7: Heterogenität	gering	hoch	-

Tabelle 7.1: Operationale Kriterien

Organisatorische Kriterien	Zentrale Koordinationsstelle	Repositorien	Wissenschaftler
Org1: Zitierbarkeit	ja	ja	ja
Org2: organisatorische Partizipation	ja	ja	-
Org3: Aufwand	gering	gering	gering
Org4: Ressourcenangebot	-	hoch	hoch

Tabelle 7.2: Organisatorische Kriterien

Funktionale Kriterien	Zentrale Koordinationsstelle	Repositorien	Wissenschaftler
Funk1: unterstützte Datenstandards und Austauschschemata	ABCD, DwC	ABCD, DwC	ABCD, DwC
Funk2: Integrierte Datenarten	Alle	Alle	Alle
Funk3: Datenverluste	nein	nein	nein
Funk4: Identitätssicherung	ja	ja	ja
Funk5: Aktualitätssicherung	ja	ja	ja
Funk6: Vergleichbarkeit von Daten	ja	ja	ja
Funk7: Data Provenance	ja	ja	ja

Tabelle 7.3: Funktionale Kriterien

Evaluationsschema

In der Analyse des Anwendungshintergrunds hat gezeigt, welche Kriterien aus Anschnitt 7.3.1 für die verschiedenen Interessensgruppen von Bedeutung sind. Dabei haben diese in vielen Bereichen übereinstimmende Ziele. Wenn sich die Interessen aber unterscheiden oder das Angebot einer Infrastruktur für verschiedene Teilnehmergruppen unterschiedlich ist, müssen diese separat bewertet werden. Die separate Betrachtung eines Kriteriums nach Teilnehmergruppen ist im Evaluationsschema dadurch gekennzeichnet, dass dem Kriterium nach seinem Kürzel für Repositorium 'r', für Wissenschaftler 'w' und für die Zentrale Koordinationstelle ein 'z' hinzugefügt wird.

Damit ergibt sich für IEF-Biodiv folgendes Evaluationsschema:

- Analyse der operationalen Ebene:

  Op1 Infrastrukturtyp

  Op2 Integrationsart

  Op3 technische Realisierung

  Op4 Zentralität

  Op5r technische Partizipation für Repositorien

  Op5w technische Partizipation für Wissenschaftler

  Op6 Autonomie

  Op7 Heterogenität der Datenspeicher

- Analyse der organisatorischen Ebene:

  Org1 Zitierbarkeit

  Org2 organisatorische Partizipation für Repositorien

  Org3a Aufwand der Repositorien

  Org3b Aufwand der Wissenschaftler

  Org4r Ressourcenangebot für Repositorien

  Org4w Ressourcenangebot für Wissenschaftler

- Analyse der funktionalen Ebene:

  Funk1 unterstützte Datenstandards und Austauschschemata

  Funk2 integrierte Datenarten

  Funk3 Datenverluste

  Funk4 Identitätssicherung

  Funk5 Aktualität

  Funk6 Vergleichbarkeit

  Funk7 Data Provenance

7.4 Anwendung von IEF-Biodiv

Im folgenden Abschnitt werden wichtige Infrastrukturen aus dem wissenschaftlichen Bereich evaluiert. Die Auswahl der Infrastrukturen beschränkt sich dabei nicht ausschließlich auf die Domäne der Biodiversitätsinformatik, sondern betrachtet mit den Gendatenbanken auch eine wichtige Infrastruktur aus der Bioinformatik. Aufgrund der Verwandtschaft des Anwendungshintergrunds kann diese auch nach dem Evaluationsschema aus Abschnitt 7.3.3 evaluiert werden.

Vor der eigentlichen Evaluation wird die jeweilige Infrastruktur vorgestellt. Die Anordnung der evaluierten Infrastrukturen folgt dabei nach ihrer Bedeutung in der Biodiversitätsinformatik. Dabei werden die wichtigsten Infrastrukturen für die Biodiversitätsinformatik zuerst evaluiert und besonders ausführlich vorgestellt. Aufgrund seiner besonderen Bedeutung in der Biodiversitätsinformatik wird GBIF in zwei Varianten vorgestellt und evaluiert. Zum einen als eigenständige Infrastruktur. Zum anderen im Kontext des IBF-Projekts, da zumindest zum Teil auf der GBIF-Infrastruktur basiert. Dabei wird in der ersten Variante der Fokus auf die grundlegende Funktionsweise von GBIF gelegt wohingegen in der zweiten Variante die konkreten Datenflüsse detailliert betrachtet werden.

Infrastrukturen, in denen kein Datenaustausch stattfindet, werden nicht evaluiert. Dazu gehören Infrastrukturen, die mit einem zentralen Server arbeiten und die Daten ausschließlich über ein Webinterface empfangen wie z.B. LIAS oder Fishbase.

7.4.1 GBIF

Vorstellung: GBIF

GBIF wurde im Rahmen seiner Bedeutung in der Biodiversitätsinformatik in Abschnitt 2.3.1 eingeführt. GBIF ist eine internationale Organisation mit dem Ziel,

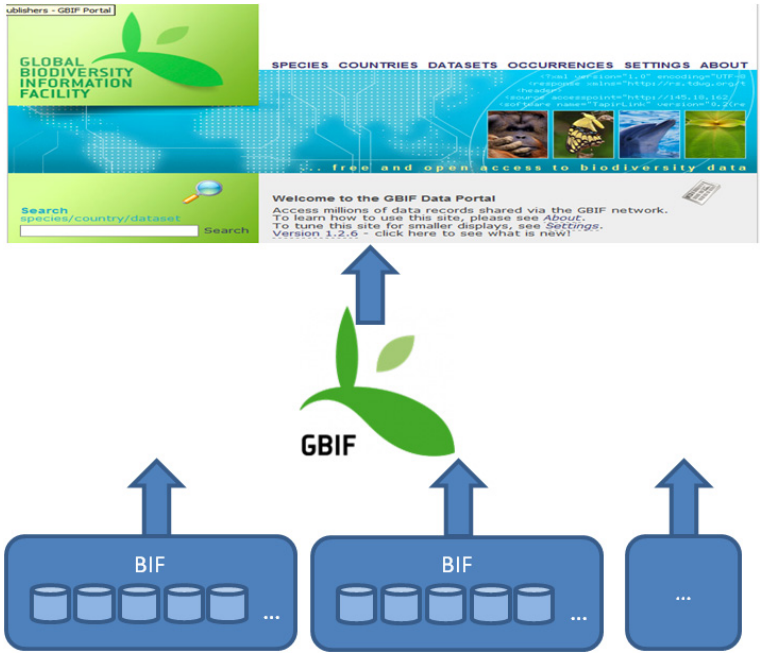

Abbildung 7.8: Datenfluss im GBIF-Datenportal

technische und wissenschaftliche Bestrebungen zur Entwicklung und Unterhaltung einer Einrichtung für den globalen Informationsaustausch in der Biodiversitätsforschung zu fördern [75]. Aus diesen Aktivitäten hat sich das GBIF-Netzwerk entwickelt, welches maßgeblich von GBIF als Organisation gelenkt wird. Dabei versteht sich GBIF als eine globale und dezentrale Initiative. Diese stellt zentrale Dienste, eine Infrastruktur und Kapazitäten für GBIF-Knoten, die auch als 'Participant Biodiversity Information Facility' (BIF) bezeichnet werden, zur Verfügung [75].

GBIF ermöglicht die freie Distribution der Daten des Netzwerks über das GBIF-Portal (siehe Abbildung 7.8). Dieses ist über ein Webinterface zugänglich. Ein beliebiger Nutzer kann so auf Daten aller angeschlossenen Anbieter zugreifen und sich Datensätze in verschiedenen Formaten (DwC, CSV, Excel, KML) herunterladen. Dabei werden die Daten der einzelnen Institute über die GBIF-Knoten dem GBIF-Netzwerk zur Verfügung gestellt. In Abbildung 7.8 ist dies über den Zusammenschluss der Datenspeicher über die Knoten (BIF) dargestellt. Diese moderieren den Datenexport an das GBIF-Netzwerk. Auf der Anderen Seite greift das GBIF-Portal, welches im oberen Bildbereich dargestellt ist, auf die Daten dieses Netzwerks zurück.

Die Datenanbieter sind hierbei über die BIF an das GBIF-Portal angeschlossen [75]. GBIF-Knoten sind im Allgemeinen auf nationaler Ebene organisiert. Allerdings können auch internationale Organisationen GBIF-Knoten werden, wenn diese über ausreichende Ressourcen verfügen. In Deutschland existiert GBIF-Deutschland als

Organisation, welche aus acht GBIF-Knoten besteht [84]. Diese acht Knoten verfügen jeweils über ein eigenes Themengebiet, wie z.B. 'Pilze und Flechten'.

Ein GBIF-Knoten stellt dabei neben einer technischen Infrastruktur auch Personal bereit, um die Ziele von GBIF zu unterstützen [75]. Damit ist es auch Aufgabe eines GBIF-Knotens, untergeordnete Datenanbieter über Schulungsmaßnahmen und mit der Bereitstellung von technischen Ressourcen zu unterstützen [82]. Dabei nehmen die GBIF-Knoten über Delegierte auch aktiv an Entscheidungen für das GBIF-Netzwerk teil.

Institutionelle Repositorien, die keine GBIF-Knoten sind, können nur über den organisatorischen Anschluss an einen GBIF-Knoten Teilnehmer im GBIF-Netzwerk werden. Für die technische Partizipation ist die eigenständige Installation der Software am Repositorium mit der eigenen Erstellung eines Mappings notwendig. Wenn Wissenschaftler an GBIF als Datenanbieter teilnehmen möchten, müssen sich diese über ein institutionelles Repositorium anschließen.

Die Komponenten eines Netzwerks – wie sich GBIF versteht – ist in Abbildung 7.9 dargestellt. Die einzelnen Komponenten dieses Netzwerks sind nach [74]:

- Datenhalter: Organisationen, die Biodiversitätsdaten produzieren, verwalten und halten
- Datennutzer: Personen und Organisationen, welche Daten in verschiedenen Bereichen nutzen
- Gemeinsame Infrastruktur: Eine Menge von Regeln, Standards, Richtlinien und Protokollen, welche alle Teilnehmer des Netzwerks einhalten müssen
- Führungsstruktur: Ein Mechanismus, welcher allen Teilnehmern der Netzwerks ermöglicht, an gemeinsamen Entscheidungen, die das Netzwerk betreffen, teilzuhaben
- zentrale Koordinationseinheit: Ein Team, welches den Datenaustausch zwischen allen Teilnehmern fördert und koordiniert
- Daten und Dienste: Primärdaten, aggregierte Daten und Tools

Auch wenn GBIF das Netzwerk aus Abbildung 7.9 als dezentral versteht, wird in [74, 75] und in Abbildung 7.9 deutlich, dass GBIF als zentrale Organisation auftritt, die den Datenaustausch im Netzwerk organisiert. Unter dezentral ist somit zu verstehen, dass einzelne Datenhalter auch direkt Daten austauschen können, ohne dass der Datenfluss über GBIF laufen muss. Dazu muss aber die von GBIF spezifizierte

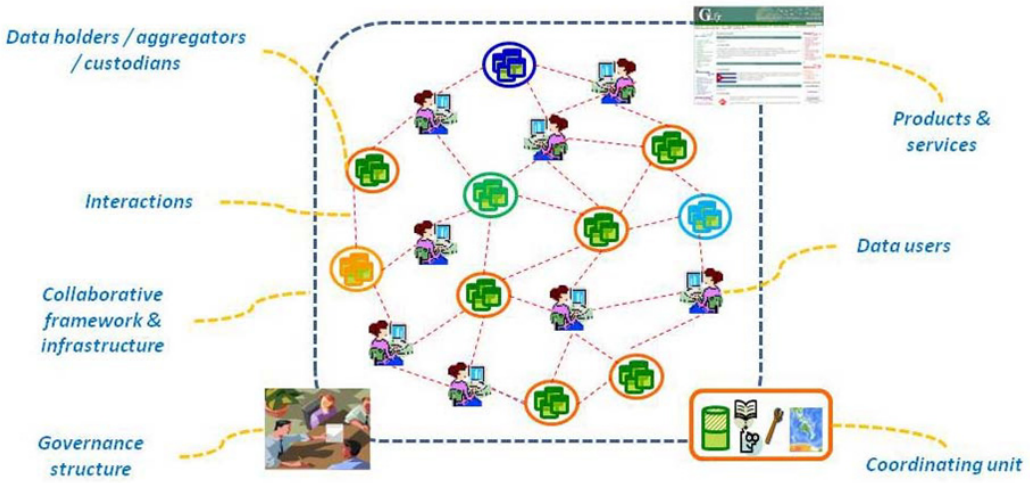

Abbildung 7.9: Komponenten des GBIF-Netzwerks nach [74]

Infrastruktur angewendet werden. Das GBIF-Netzwerk ist demnach eine dezentrale Infrastruktur mit zentraler Koordinationsstelle.

Die technische Architektur von GBIF und die Anbindung von Datenquellen ist in Abbildung 7.10 dargestellt. Die technische Anbindung von Datenquellen an GBIF wird über folgende vier Wege realisiert [78, 73]:

- **Integrated Publishing Toolkit** (IPT): Dies ist der von GBIF empfohlene Weg [78]. IPT ist eine auf Java basierende Middleware, die als gemeinsames Datenmodell DwC verwendet, aber auch EML unterstützt [262].

- **TDWG Access Protocol for Information Retrieval** (TAPIR): TAPIR ist ein Protokoll mit dem Status eines TDWG Standards [228]. Der Ansatz zur Entwicklung von TAPIR war die Kombination und die Erweiterung von BioCase und DiGIR [45]. TAPIR setzt dabei unter allen Teilnehmern eines Netzwerks ein gemeinsames Datenaustauschmodell voraus, das aber nicht weiter spezifiziert sein muss [46]. In der Praxis wird allerdings fast ausschließlich ABCD oder DwC als globales Datenschema im Kontext von TAPIR eingesetzt. Es gibt verschiedene Provider Software, die TAPIR unterstützt. Hierbei ist insbesondere TAPIRLink von Bedeutung.

- **Biological Collection Access Service for Europe** (BioCASE): BioCASE ist primär eine Organisation, die ein eigenes Netzwerk betreibt [17], welches mit GBIF kooperiert. Diese bietet den PyWrapper als Software zum Datenaustausch an [19]. BioCase ist mit dem PyWrapper prinzipiell von einem bestimm-

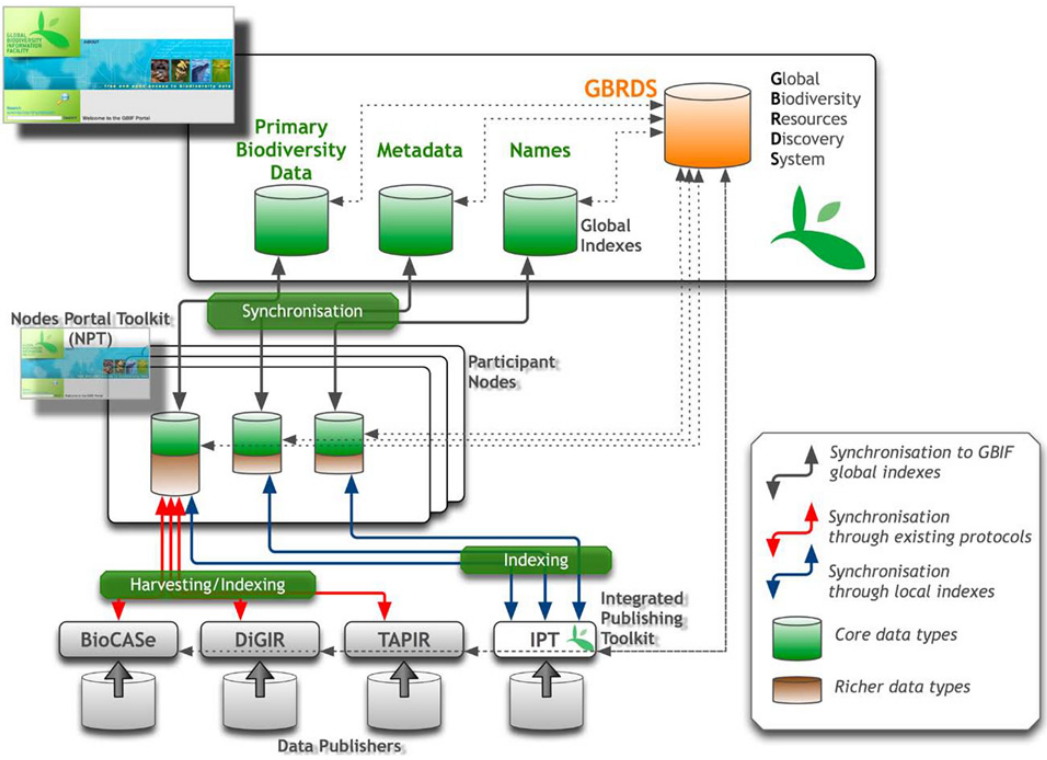

Abbildung 7.10: GBIF Dezentralisierungsstrategie nach [73]

ten Schema unabhängig [16], wird aber in der Praxis für den Datenaustausch mit ABCD verwendet.

- **Distributed Generic Information Retrieval** (DiGIR): Ein Protokoll auf PHP-Basis [247], welches mit DwC als globales Schema arbeitet. Es wird als veraltet betrachtet, ist aber durchaus noch praxisrelevant. DiGIR bildet auch den Ausgangspunkt der Entwicklung von TAPIRLink.

Die Entwicklung von IPT wurde in Angriff genommen, da die älteren Standards insbesondere im Bezug auf den Umgang mit großen Datenmengen sich als nachteilig erwiesen [262]. Darüber hinaus ist IPT Teil der GBIF Dezentralisierungsstrategie (Abbildung 7.10) [262]. Der Einsatz von DwC in IPT als globales Datenschema ist eine wesentlichen Einschränkung gegenüber den anderen Wegen. Dies führt aber auch zu einer einfacheren Anwendbarkeit. IPT erlaubt das Einspielen von Datensätzen im DwC-Format und den Anschluss von Datenbanken an Datenquellen [262]. Das Mapping zwischen einem lokalen Datenbankschema und DwC muss aber in der IPT Provider Software manuell hergestellt werden [262].

Weitere wesentliche Komponenten der GBIF-Architektur sind das 'Global Biodiversity Resources Discovery System' (GBRDS), das 'Harvesting and Indexing Toolkit' (HIT) und das 'Nodes Portal Toolkit' (NPT) [73]. Das GBRDS fungiert hierbei als zentrale Stelle für die Registrierung von Datenquellen und Services [6, 79] und hat damit eine Aufgabe analog zu UDDI bei Webservices. In diese werden Datensätze aus Datenquellen, die IPT verwenden, automatisch eingetragen.

HIT ermöglicht die materialisierte Informationsintegration von angeschlossenen Datenbanken an GBIF und die Indizierung der Daten aus diesen Quellen [76]. NPT ist ein Werkzeug, das für GBIF-Knoten entwickelt wurde, um Daten speziell an deren Nutzer zu verteilen und weitere Ressourcen einzubinden [77]. Diese Elemente der Infrastruktur sind aktuell Gegenstand einer stetigen Entwicklung und zum aktuellen Zeitpunkt noch nicht vollständig implementiert.

Evaluation: GBIF

operationale Ebene: Das GBIF-Netzwerk ist eine **offene, dezentrale Infrastruktur** mit einer **zentralen Koordinationsstelle**, welche **virtuelle und materialisierte Informationsintegration** ermöglicht. Die Datenquellen werden dabei auch bei der materialisierten Informationsintegration über Wrapper, welche TAPIR, BioCASE oder DiGIR unterstützen, an GBIF angebunden. Die **technische Realisierung** ist vom Integrationsweg abhängig. Als einziger Wrapper generiert IPT

einen Zugang zu GBIF über objektorientierte Middleware mit Hilfe von Java. DiGIR transportiert die Daten über das HTTP-Protokoll [247]. BioCase und TAPIR stellen Protokolle dar mit Hilfe derer Wrapper wie z.B. TAPIRLink und der PyWrapper Daten im XML-Format transportieren. Die **direkte technische Partizipation** ist für **institutionelle Repositorien** nur mit hohem Aufwand und als Anbieter einer relationalen Datenbank als Datenspeicher möglich. **Wissenschaftler** können nur indirekt über den Anschluss an ein Repositorium als Datenanbieter partizipieren – die **Partizipation als Datennutzer** ist hingegen frei.

Heterogenität kann in allen vorgestellten Formen der Heterogenität auftreten. Die Lösung des Heterogenitätsproblems von GBIF liegt in der Einschränkung der Autonomie der angeschlossenen Datenquellen. Die **Schnittstellenautonomie** ist auf vier Zugangswege eingeschränkt, zu deren Einbindung unterschiedliche Softwareprodukte (z.B. TAPIRLink, PyWrapper, TAPIRDotNet) angeboten werden. Bei der Verwendung von BioCase und TAPIR ist theoretisch **Designautonomie** vorhanden, da in diesen Protokollen kein bestimmtes konzeptuelles Schema verwendet werden muss. Allerdings muss zur Anbindung an GBIF das Datenformat auch bei diesen Protokollen zu GBIF kompatibel sein, so dass auch hier ABCD und DwC als Austauschformat praktisch vorgegeben sind. Bei der Verwendung von IPT und DiGIR ist keine Designautonomie vorhanden, da diese Protokolle nur zusammen mit DwC eingesetzt werden [247, 262]. Da die Daten über das GBIF-Netzwerk frei zugänglich sind, ist keine **Zugriffsautonomie** vorhanden. Innerhalb des Netzwerks konnten auch keine Strukturen zur Wahrung der juristischen Autonomie der einzelnen Datenquellen identifiziert werden. Den einzelnen Datenanbietern ist das Mapping auf die Datenmodelle selbst überlassen. Somit müssen diese das globale Schema nicht vollständig unterstützen und es können Daten zurückgehalten werden (z.B. durch Felder wie 'Datawithholding reason', mit Hilfe derer am SNSB die Weitergabe von Daten aktiv verhindert werden kann).

organisatorische Ebene: Das GBIF-Netzwerk verfügt in der Domäne der Biodiversitätsforschung über ein gutes Renommee. Die **Zitierfähigkeit** der Daten ist damit gewährleistet. Repositorien können **indirekt** über den Anschluss an einen GBIF-Knoten an der Organisation **partizipieren**. Für GBIF-Knoten ist die **direkte Partizipation** möglich. Der **Aufwand** für Repositorien ist hoch, da neben der selbständigen Installation von Software auch der eigene Datenbestand über selbst zu erstellende Mappings in die Infrastruktur integriert werden muss. Zusätzlich muss ein Repositorium, das als GBIF-Knoten fungiert, Schulungen für angeschlossene Institute anbieten, so dass der personelle Aufwand für GBIF-Knoten als sehr hoch an-

zusehen ist. Wissenschaftler können mit geringem Aufwand Daten nutzen, aber nicht direkt als Datenanbieter auftreten. Das Ressourcenangebot für GBIF-Knoten ist dafür umfassend. Es werden auch **Ressourcen** für Wissenschaftler und Repositorien, die keine GBIF-Knoten sind, zur Verfügung gestellt, für die Ressourcenversorgung dieser Parteien sind primär der GBIF-Knoten zuständig.

funktionale Ebene: Durch die Verwendung von ABCD und DwC als **Austauschschema** ist der Datenaustausch über das GBIF-Netzwerk mit hohen Datenverlusten verbunden (siehe Abschnitten 1.3, sowie die Evaluation von ABCD in Abschnitt 4.4.1 und DwC in Abschnitt 4.4.2), da diese Standards die Domäne der Biodiversitätsforschung nicht ausreichend erfassen können. Mechanismen zur Sicherung der Identität eines Datensatzes sind zwar im Ansatz vorhanden, werden aber nicht konsequent eingesetzt. Somit ist das Kriterium nicht erfüllt. Mechanismen zur Sicherung der Aktualität der Daten bei der materialisierten Integration über das Netzwerk wurden nicht getroffen – allerdings übernimmt die GBIF-Cache Datenbank regelmäßig Aktualisierungen aus angeschlossenen Datenquellen. Die Vergleichbarkeit der Daten ist im Rahmen der Möglichkeit von ABCD und DwC gegeben. Die Daten können dabei eindeutig den Datenquellen zugeordnet werden. Über die Garantie der Herkunft hinaus wird **Data Provenance** nicht unterstützt.

Fazit: GBIF

GBIF stellt für den Datenaustausch in der Biodiversitätsinformatik durch sein hohes Renommee einen Quasi-Standard dar und ist auf organisatorischer Ebene insgesamt vorbildlich. Die operationale Ebene ist für den Anwendungshintergrund geeignet und ermöglicht auf verschiedenen Wegen an GBIF zu partizipieren. Problematisch ist allerdings, dass nach der Architektur von GBIF Daten nur in bestimmten Formaten übertragen werden können, welche Daten aus der Biodiversitätsforschung nicht mit der erforderlichen Genauigkeit erfassen können (siehe Abschnitt 4.4). Auch der Transport von Multimediadaten wird aktuell nicht unterstützt [83]. Damit treten in der GBIF-Infrastruktur erhebliche Datenverluste auf. Auch die anderen Kriterien der funktionalen Ebene sind nur unzureichend erfüllt, so dass die Identität und die Aktualität der Daten über das Netzwerk nicht garantiert ist. Dies ist insbesondere vor dem Hintergrund der herausragenden Stellung von GBIF in der Biodiversitätsinformatik kritisch, so dass GBIF in der funktionalen Ebene dringend einer Weiterentwicklung bedarf (siehe Kapitel 8).

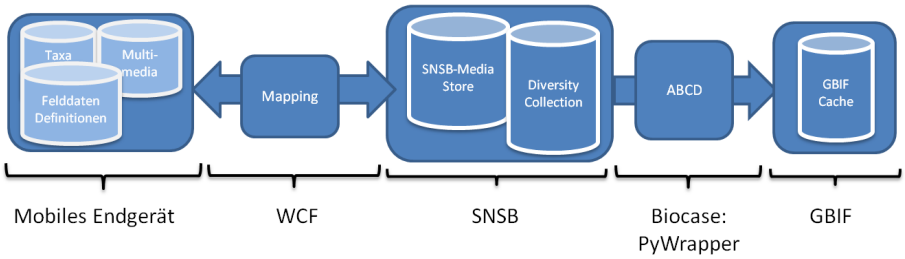

Abbildung 7.11: Informationsintegrationsschritte im IBF-Projekt

7.4.2 IBF-Infrastruktur

Vorstellung: IBF-Infrastruktur

Das IBF-Projekt wurde in Abschnitt 2.3.2 bereits vorgestellt. Ziel des Projekts ist die langfristige Speicherung von Primärdaten inklusive Multimediadaten und deren Integration in das GBIF-Netzwerk. Dazu werden primäre Forschungsdaten von Wissenschaftler mit mobilen Geräten mit 'DiversityMobile' im Feld erhoben und an das SNSB (siehe Abschnitt 2.3.1) übertragen. Von der dort werden die Daten in die GBIF-Infrastruktur exportiert, welche damit Teil der Infrastruktur des IBF-Projektes ist. Dadurch lässt sich die Infrastruktur des IBF-Projektes in verschiedene Komponenten unterteilen:

- Primäre Datenspeicherung im Mobilgerät
- Speicherung am institutionellen Repositorium des SNSB
- Export in die GBIF-Infrastruktur

Innerhalb des IBF-Projektes läuft die Informationsintegration in zwei Schritten ab. Zunächst werden die mit dem Mobilgerät erhobenen Daten in das Repositorium des SNSB integriert. Von dort werden die Daten an GBIF weitergeleitet. Ein Überblick über diese Architektur findet sich in Abbildung 7.11.

Darüber hinaus wird mit 'DiversityCollection' eine Software vom SNSB zur Verfügung gestellt, mit der Daten direkt auf dem Repositorium bearbeitet werden können. Dabei ermöglicht 'DiversityCollection' die Speicherung in einer lokalen Datenbank mit anschließender Synchronisation in das SNSB-Repositorium.

Im ersten Integrationsschritt werden die Daten von mobilen Geräten über einen WCF-Service an das SNSB übertragen. Da auf dem Mobilgerät eine leicht variierte Version des Datenmodells (DM siehe [211]) von 'DiversityCollection' (DC siehe [261]) verwendet wird, müssen die Daten aus 'DiversityMobile' dem Datenmodell von

'DiversityCollection' angepasst werden. Sowohl 'DiversityMobile' als auch 'DiversityCollection' sind in C# programmiert. Das Repositorium arbeitet auf Basis eine MSSQLServers und die Webservices der Infrastruktur basieren auf der 'Windows Commmunication Foundation' (WCF). Folglich wird objektorientierte Middleware aus dem .NET-Bereich eingesetzt. Das Mapping zwischen den Modellen erfolgt über einen speziell dafür entwickelten WCF-Webservice. Die für RPC´s oftmals bemängelte langsame Datenübertragung wirkt sich hierbei noch nicht aus, da vom Mobilgerät vergleichsweise kleine Datenmengen transferiert werden.

Für die Datenübertragung nach GBIF wird die Variante des Datenimports über 'BioCase' mit dem 'PyWrapper' verwendet. Das Mapping von DC nach ABCD wird in der Konfiguration des 'PyWrappers' spezifiziert [19]. Über den 'PyWrapper' wird eine Oberfläche zur Verfügung gestellt, welche es ermöglicht ein Mapping zwischen der Datenbank des Repositoriums und dem ABCD-Schema herzustellen. Hierbei wird nur eine Teilmenge der über 1200 Konzepte von ABCD unterstützt (vgl. [18] und [229]), wobei allerdings einige als obligatorisch bzw. empfohlen gekennzeichnet sind [18]. Bei der Informationsintegration über den PyWrapper wird somit nicht die volle Mächtigkeit von ABCD ausgenutzt, was eines der Gründe für die in Abschnitt 2.2.3 beschriebenen Datenverluste ist. Die Datenverluste an dieser Stelle werden dabei nicht nur durch Mängel von ABCD (siehe 4.4.1) sondern auch durch Mängel im Mapping von DCD nach ABCD in der Konfiguration des PyWrappers hervorgerufen.

Darüber hinaus ist die IBF-Infrastruktur ab dem Datenexport aus dem Repositorium Teil des GBIF-Netzwerkes (siehe 7.4.1) und damit den Beschränkungen dieses Netzwerks unterworfen.

Ein zentraler Punkt in der Infrastruktur ist das Repositorium des SNSB. Dieses ist auch organisatorischer Ebene auch eine zentrale Koordinationsstelle und als GBIF-Knoten für Pilze und Flechten [84] auch auf organisatorischer Ebene die Schnittstelle zu GBIF.

Evaluation: IBF

Die Infrastruktur von IBF ist zweigeteilt, in eine Infrastruktur bis zur Informationsintegration am Repositorium des SNSB und in die genutzte Infrastruktur von GBIF mit dem 'PyWrapper' über 'BioCase'. Da die GBIF-Infrastruktur bereits in Abschnitt 7.4.1 evaluiert wurde, bezieht sich die folgende Evaluation auf die Infrastruktur bis zur Integration am SNSB – sofern dies nicht explizit kenntlich gemacht ist.

Operationale Ebene: Die IBF-Infrastruktur ist eine **offene, zentrale Infrastruktur** mit dem SNSB Repositorium als zentrale Stelle. Die Informationsintegration erfolgt **materialisiert** auf Basis einer **technischen Realisation** mit einer objektorientierten Middleware in .Net. Die **technische Partizipation** ist für Repositorien nicht möglich. Wissenschaftler benötigen für die technische Partizipation eine WindowsPhone-Mobilgerät oder einen auf Windows basierenden PC oder Tablet-PC, auf denen 'DiversityCollection' oder 'DiversityMobile' installiert ist. Als Datenspeicher kommen ausschließlich MSSQL CE-Datenbanken und MSSQL-Datenbanken zum Einsatz. Die **Heterogenität** ist dementsprechend gering und **Autonomie** praktisch nicht vorhanden.

Organisatorische Ebene: Das SNSB als eigenständige Organisation kann **Zitierfähigkeit** von Daten indirekt über den Anschluss an GBIF gewährleisten, da das SNSB als GBIF-Knoten für Pilze und Flechten [84] in Deutschland fungiert. Darüber hinaus verfügt das SNSB als staatliche Sammlung über einen tadellosen Ruf. Da der gesamte Datenbestand aber nicht allgemein zugänglich ist, ist die Zitierfähigkeit nur zusammen mit GBIF gewährleistet. Das SNSB verwaltet seine Infrastruktur auch auf organisatorischer Ebene zentral und ermöglicht **keine organisatorische Partizipation**. Der **Aufwand** für Wissenschaftler ist mit der Installation für die Software gering – wird aber im mobilen Bereich durch die Beschränkung auf bestimmte Plattformen und Technologien eingeschränkt, so dass dieser insgesamt als moderat zu bezeichnen ist. Als **Ressourcenangebot** stehen regelmäßige Schulungen für Wissenschaftler und ein umfassendes Wiki (siehe [52]) zur Verfügung.

Funktionale Ebene: In der IBF-Infrastruktur entstehen potentielle **Datenverluste** an drei Stellen. Zum einen kann das vollständige Datenmodell in [261] nicht alle gewünschten Sachverhalte abbilden – zum anderen können auch Sachverhalte nicht in dem mobilen Datenmodell [211] vollständig abgebildet werden. Für die im IBF-Projekt betrachteten wissenschaftlichen Projekte war eine verlustfreie Datenübertragung von DiversityMobile [211] nach DiversityCollection[261] möglich. Auch Multimediadaten werden vollständig übertragen. Problematisch ist allerdings das Mapping über den 'PyWrapper' nach ABCD, welches mit erheblichen Datenverlusten verbunden ist. Die **Identitätssicherung** der Datensätze erfolgt über den Primärschlüssel des Repositoriums, welcher nach Synchronisation in den mobilen Clients mitgeführt wird und über GUID´s im Tabletbereich. Aufgrund des **zentralen Aufbaus der Infrastruktur** besteht **kein Aktualitätsproblem** im Sinne von Abbildung 7.7, so dass diesbezüglich keine Vorkehrungen getroffen werden mussten. Datensätze, die an das Repositorium einmal übertragen wurden, können allerdings

nicht mehr vom Mobilgerät editiert werden [51]. Aufgrund der Kompatibilität von DC [261] und DM [211] ist die **Vergleichbarkeit** der Daten bis zum Repositorium am SNSB gegeben. Anschließend gelten die Beschränkungen der GBIF-Infrastruktur (siehe 7.4.1). **Data Provenance** wird über Versionierung und Speicherung der Urheberschaft in der Datenbank am SNSB umgesetzt [261]. Da [211] und [261] miteinander kompatibel sind, werden keine Transformationen benötigt und diese somit auch nicht verfolgt.

Fazit: IBF

Die IBF-Infrastruktur bietet Wissenschaftlern eine gute Möglichkeit zur langfristigen Speicherung ihrer Daten an einem Repositorium mit dem potentiellen Export nach GBIF. Dies ist notwendig, um die Zitierfähigkeit der Daten zu garantieren. Allerdings ist der **Export** nach GBIF durch die Konvertierung der Daten in das ABCD-Schema mit **erheblichen Datenverluste** verbunden. Dies beinhaltet auch den Verlust der Multimediadaten. Intern verwendet das IBF-Projekt die Datenmodell DM und DC. Diese sind zueinander kompatibel und ermöglichen den verlustfreien Transport der Primär- und Multimediadaten an das SNSB. Wissenschaftler halten darüber hinaus auf organisatorischer Ebene eine gute Unterstützung durch Schulungsangebote und der Software- und Projektdokumentation in Wikis.

7.4.3 LTER-Netzwerk

Vorstellung: LTER-Netzwerk

Das 'Long Term Ecological Research' (LTER)-Netzwerk geht auf eine Initiative der 'US National Science Foundation' zurück [105]. Seit dem Start von LTER 1980 [105] ist das Netzwerk von sechs auf 26 Standorten in den USA angewachsen und hat die Expansion zu einem internationalen LTER-Netzwerk[2] [162] als Ziel. So gibt es auch eine Vielzahl an europäischen Standorten, welche über LTER-Europe organisiert sind [147]. Allerdings sind diese Datenbestände noch nicht in das US-LTER-Netzwerk integriert. Das zentrale Ziel von LTER ist das Verständnis von Mustern und Prozessen von ökologischen Systemen in unterschiedlichen räumlichen Dimensionen über lange Zeiträume [105]. Die Betrachtung von Langzeit-Daten ist dabei die Basis, um Veränderungen in Ökosystemen zu identifizieren [105], die sich nur langsam vollziehen. So werden über LTER Daten gesammelt, deren Evaluation Jahrzehnte oder Jahrhunderte dauern kann [105]. Dazu werden auch historische Daten in Veröffentlichungen

[2] http://www.ilternet.edu/

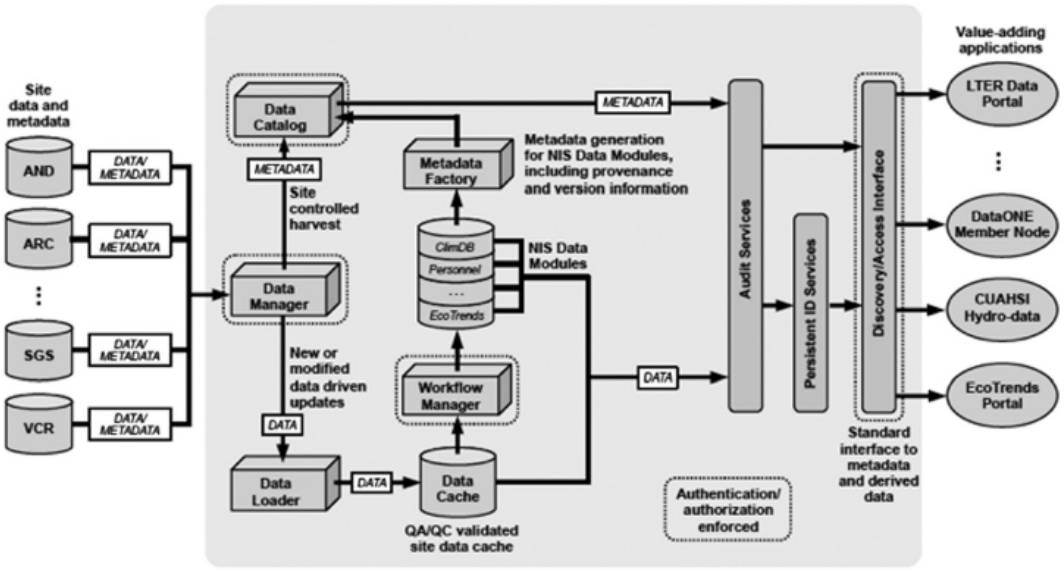

Abbildung 7.12: Planung der LTER-Infrastruktur nach [162]

wie in [136] ausgewertet [105]. Diese Veröffentlichungen können dann über die Homepage des LTER-Netzwerks [3] gefunden werden. Die jeweiligen LTER-Standorte arbeiten dabei primär unabhängig voneinander in fünf verschiedenen Bereichen der ökologischen Langzeitbetrachtung, wobei auch standortübergreifende Projekte Teil des Netzwerks sind [105].

Die Infrastruktur von LTER basiert auf dem 'LTER Network Information System' (NIS) (siehe Abbildung 7.12). Diese Infrastruktur ist aktuell noch nicht vollständig implementiert und soll 2014 fertiggestellt werden [162]. Kern dieser Infrastruktur ist das 'Provenance Aware Synthesis Tracking Architecture' (PASTA)-Framework, welches in Abbildung 7.12 grau hinterlegt dargestellt ist und auf [216] zurückgeht. PASTA ermöglicht die materialisierte Informationsintegration von Rohdaten über den 'Data Manager' und von sogenannten Metadaten in den 'Data Catalog'. Unter Metadaten[4] werden in diesem Kontext Daten über Projekte und personenbezogenen Daten verstanden, welche in EML (siehe Abschnitt 4.4.7) geführt werden [162] und die Basis des Datenaustausch in LTER bilden. PASTA basiert auf Service-Orientierter Architektur (SOA) und wurde in Java implementiert [216]. Für die Informationsintegration ist der Metacat-Harvester verantwortlich [216], welcher an den Standorten des LTER-Netzwerks installiert sein muss. Die Installation von Metacat setzt die Installation von Apache Tomcat, Java und Postgres SQL voraus [200].

[3] www.lternet.edu

[4] Nicht zu Verwechseln mit der Metamodellierung aus Abschnitt 5.2

Die Distribution der Daten im Netzwerk erfolgt über das LTER-Datenportal [5]. Dabei sind zu einzelnen Projekten neben den Daten in EML die Rohdaten des Projektes verfügbar, die nicht standardisiert sind. Diese sind im Allgemeinen als Textdateien z.B. in einem csv-Format oder in Excel verfügbar, welche ohne den projektspezifischen Kontext in EML nicht verstanden werden können. Das LTER-Netzwerk ist dabei als Infrastruktur zentral organisiert. An den Standorten werden primär relationale Datenbanken [162] als Datenspeicher verwendet. Die Metadaten der Standorte werden über PASTA in eine XML-Datenbank zentral integriert, wobei EML als globales Schema dient [216].

Daten, die über PASTA in das LTER-Netzwerk integriert werden, werden im Allgemeinen erst nach zwei Jahren öffentlich zur Verfügung gestellt [216]. Je nach Integrationsgrad werden fünf Qualitätsstufen unterschieden [162]. Bei der Integration in das Netzwerk sind kuratorische Schritte involviert, um die Datenstruktur verändern können und Transformationen an Daten auszuführen(wie z.B. Umrechnungen in Standardeinheiten) [162]. Um die Originaldaten zu erhalten und Veränderungen nachvollziehen zu können, unterstützt NIS Data Provenance. Die Arbeiten an NIS mit PASTA sind aktuell noch nicht abgeschlossen [162]. Aktuell ist bereits die Suche im LTER-Netzwerk über das LTER-Datenportal möglich, das Zugang zu projektspezifischen Daten in EML und Rohdaten in nicht-standardisierter Form bietet. Die Informationsintegration erfolgt aktuell nur für die Metadaten auf EML-Ebene. Die Primärdaten werden zwar zur Verfügung gestellt, aber nicht integriert.

LTER verfügt über eine Führungsstruktur, welche aus dem Executive Board, der LTER-Network-Office und weiteren Strukturen besteht, an welchen Abgeordnete von LTER-Standorten mitwirken [162].

Evaluation: LTER-Netzwerk

Operationale Ebene: LTER ist eine **offene, zentrale Infrastruktur**, welche mit **materialisierter Informationsintegration** ermöglicht. Die **technische Realisierung** erfolgt auf Basis einer SOA mit Java. Die **technische Partizipation** ist ausschließlich Repositorien vorbehalten, die bis auf die obligatorische Verwendung von PASTA und der Strukturierung der Metadaten in EML in der **Autonomie** nicht eingeschränkt sind. Die **Heterogenität** der Datenspeicher ist sehr hoch. So basieren lediglich 15 der 26-LTER Standorte auf relationalen Datenbanken – die übrigen Standorte arbeiten mit einem Dateisystem oder gemischten Ansätzen [145].

Organisatorische Ebene: Die Einspeisung der Daten in das LTER-Netzwerk

[5] https://metacat.lternet.edu/das/lter/index.jsp

ermöglicht das **Zitierfähigkeit** der Daten, wobei diese im Allgemeinen auf anderen Wegen zuerst veröffentlicht werden. Die **organisatorische Partizipation** an LTER ist Repositorien vorbehalten, welche zu LTER-Standorten werden können. Repositorien müssen Metacat installieren und müssen damit einen Webserver unterhalten. Darüber hinaus müssen Datensätze über HHTP und HTTPS zugänglich sein [145]. Der **Aufwand** der Partizipation an LTER ist damit hoch. Wissenschaftler können indirekt als Datenanbieter über den Anschluss an einen LTER-Standort partizipieren. Der Zugang zu Daten über das LTER-Netzwerk ist jedermann möglich, wobei die Daten erst nach zwei Jahren nach Erhebung veröffentlicht werden [216]. Die Installation von Software ist für Wissenschaftler für den Download von Daten nicht erforderlich – der Upload kann nur über den Anschluss an einen LTER-Standort erfolgen. Der Aufwand für Wissenschaftler ist damit gering. Das **Ressourcenangebot** des LTER-Netzwerks besteht in gelegentlichen Trainings für Wissenschaftler und durch eine umfassende Dokumentation für Wissenschaftler und Repositorien. Außer der Dokumentation konnten aber keine speziellen Ressourcen für das LTER-Netzwerk für Repositorien gefunden werden. Das Ressourcenangebot ist damit für Repositorien gering, das für Wissenschaftler im mittleren Bereich.

Funktionale Ebene: Der **Datenaustausch** im LTER-Netzwerk ist aktuell auf den Datenaustausch von EML-Daten und Rohdaten, die nicht standardisiert sind, beschränkt. Dementsprechend treten im LTER-Netzwerk **keine Datenverluste** auf. Die Verteilung von Multimediadaten wird nicht unterstützt. Allerdings sind die Primärdaten in EML oft unzureichend beschrieben [216], so dass die Nutzung von Primärdaten durch andere Wissenschaftler im Allgemeinen nicht möglich ist. Dies liegt daran, dass im LTER-Netzwerk die globale Sicht der Daten aktuell auf EML beschränkt ist, welches keine Primärdaten erfassen kann. Primärdaten werden damit unstrukturiert über das Netzwerk verteilt. Eine **Vergleichbarkeit** der Daten kann über die Metadaten hinaus somit nicht erfolgen. Die Sicherung der **Identität** und **Aktualität** ist ebenfalls auf EML beschränkt. Die Identitäts- und Aktualitätssicherung von Primärdaten ist nicht möglich. Die Unterstützung von **Data Provenance** ist in LTER auch für Primärdaten geplant [216]. Der aktuelle Stand der Umsetzung ist nicht bekannt.

Fazit: LTER-Netzwerk

Das LTER-Netzwerk tauscht Daten im EML-Format aus. Dazu müssen aktuell die Daten von den Standorten im EML-Format zur Verfügung gestellt werden. Eine eigentliche Integration von Primärdaten in ein globales Schema findet nicht statt. Pri-

märdaten werden somit nur unstrukturiert zur Verfügung gestellt. Dementsprechend kann das LTER-Portal aktuell nur als Suchmaschine für Primärdaten und Veröffentlichungen betrachtet werden, welche eine Suche auf Basis einer EML-Formatierung ausführt. Da EML nur Metadaten erfassen kann, wird der eigentliche Transfer von Primärdaten nicht unterstützt. Der Transfer von Multimediadaten ist in der aktuell nicht implementiert und scheint auch nicht geplant zu sein. Damit ist der Datenaustausch über das LTER-Netzwerk in der aktuellen Implementierung stark eingeschränkt.

7.4.4 DataONE

Vorstellung: DataONE

Das 'Data Observation Network for Earth' (DataONE) ist ein verteiltes Netzwerk, welches eine Infrastruktur zu dem Zweck unterhält, den Zugriff auf Beobachtungsdaten über das Leben auf der Erde und die belebte Umwelt zu ermöglichen [44, 163]. Damit umfasst DataONE auch den Datenaustausch in der Biodiversitätsforschung, wobei der Ansatz von DataONE umfassender ist. Dieser Aufgabe kommt DataONE über folgende Wege nach [163]:

- Einbeziehung aller relevanten Wissenschafts-, Daten- und politischen Communities
- Ermöglichen von dauerhafter Datenspeicherung
- Verbreitung von Tools zur Aufbereitung, Visualisierung und Analyse von Daten

DataONE ist als ein verteiltes Netzwerk aus Knoten mit unterschiedlicher Bedeutung aufgebaut. Drei Knoten sind aktuell als 'Coordinating Nodes' ausgezeichnet, die für unterschiedliche Communities verantwortlich sind und neben der Datenhaltung auch koordinative Aufgaben übernehmen [163]. Diesen sind die 'Member Nodes' untergeordnet, die für die eigentliche Datenhaltung verantwortlich sind und aus Repositorien aus verschiedenen Domänen bestehen [163]. Einzelpersonen können über den Anschluss an einen Member Node teilnehmen und erhalten über das 'Investigator Toolkit' für DataONE geschriebene Software oder angepasste Standardprogramme [163]. Für die Koordination verfügt DataONE über eine umfangreiche Führungsstruktur, an der Member und Coordinating Nodes organisatorisch partizipieren können, aber unter dem Einfluss von Geldgebern wie der 'National Science Foundation' (NSF) stehen [44].

Der Stand der Implementierung ist aktuell prototypisch [163]. Es können aber bereits Daten über das in der Homepage (siehe [44]) integrierte Datenportal abgerufen werden. Daten werden wie im LTER-Netzwerk Daten primär auf der Ebene der Metadaten integriert, wobei neben EML auch eine Vielzahl anderer Metadaten-Standards wie Dublin-Core unterstützt wird. Dies ist vom Anschluss an den 'Coordinating Node' abhängig [163]. Für wissenschaftliche Primärdaten ist kein gemeinsames Datenmodell für DataONE publiziert. Test der Funktionalität des DataONE-Datenportals[6] ergaben, dass Primärdaten analog zum LTER-Netzwerk in unstandardisierter Rohform erhältlich sind. Die technische Grundlage für den Datenaustausch zwischen den Knoten bieten sowohl RESTful Services und als auch Remote Procedure Calls [43].

Ein wesentliches Merkmal von DataONE ist die Rolle von Identifikatoren, über die die Identität eines Datensatzes sichergestellt wird. Innerhalb von DataONE werden diese nach bestimmten Prinzipien gebildet, zu denen Eindeutigkeit, Unveränderlichkeit, Verborgenheit und Auflösbarkeit zählen [163]. Die Verantwortlichkeit für die Vergabe von Identifikatoren liegt dabei bei den Member Nodes [163]. Darüber hinaus ist es Ziel von DataONE Repositorien und Wissenschaftler durch ein umfassendes Trainingsangebot zu unterstützen [163].

Evaluation: DataONE

Operationale Ebene: DataONE ist eine **offene, dezentrale Infrastruktur mit zentraler Koordination**, welches **virtuelle Informationsintegration** ermöglicht. Allerdings existieren aktuell drei 'Coordinating Nodes' mit besonderen Aufgaben. Der Kreis der 'Coordinating Nodes' ist nicht erweiterbar, so dass DataONE im Bezug auf die 'Coordinating Nodes' nicht offen ist. Die **technische Realisierung** über RESTFul Services und RPCs. Die **technische Partizipation** erfolgt für Repositorien als Member Nodes direkt wohingegen die technische Partizipation von Wissenschaftlern nur indirekt über den Anschluss an einen 'Member Node' möglich ist. Die **Heterogenität** der Datenspeicher ist hoch und die **Autonomie** ist vor allem in der Schnittstellenautonomie eingeschränkt.

Organisatorische Ebene: DataOne ermöglicht die **Zitierfähigkeit** der Daten. Der **Aufwand** der Repositorien zur Partizipation ist hoch. Die **organisatorische Partizipation** von Wissenschaftlern ist nur indirekt über den Anschluss an einen 'Member Node' möglich. Sowohl Wissenschaftlern als auch Repositorien werden **umfassende Ressourcen** über ein Trainingsangebot [44] und das 'Investigator Toolkit' zur Verfügung gestellt.

[6] http://www.dataone.org/data

Funktionale Ebene: Aufgrund der Beschränkung auf virtuelle Informationsintegration besteht weder das **Aktualitäts- noch das Identitätsproblem**. **Datenverluste** können nur auf Ebene der Metadaten entstehen, da die Primärdaten als Originaldaten über das Netzwerk verfügbar sind. Die Primärdaten sind dabei allerdings nicht in standardisierter Form verfügbar, so dass keine **Vergleichbarkeit** der Daten vorliegt. Dabei unterstützt DataONE eine Vielzahl von Metadatenstandards, so dass bei Verwendung dieser in den Member Nodes keine Datenverluste auftreten. Für **Data Provenance** wurde in [167] ein Konzept auf Basis von Workflows vorgestellt, welches aber aktuell noch nicht implementiert ist. Der Austausch von Multimediadaten ist aktuell nicht verfügbar, könnte aber Teil der zukünftigen Entwicklung von DataONE sein.

Fazit: DataONE

DataONE ermöglicht analog zu LTER lediglich die Integration von Metadaten und die Distribution von Primärdaten in unstandardisierter Form, so dass der Nutzen des Datenaustauschs von Primärdaten stark eingeschränkt ist. Allerdings ist auf funktionaler Ebene die Rolle von Identifikatoren hervorzuheben, welche die Identitätssicherung im Netzwerk ermöglicht. Ein weiterer fortschrittlicher Ansatz in DataONE ist die Unterstützung von REST. Damit werden mit DataONE gute Ansätze für eine Weiterentwicklung von Infrastrukturen in der Biodiversitätsinformatik gegeben.

7.4.5 Gendatenbanken

Vorstellung: Gendatenbanken

Gendatenbanken für Nukleotidsequenzen sind Elemente der Infrastruktur der Bioinformatik und nicht der Biodiversitätsinformatik. Sie sollen hier trotzdem kurz besprochen werden, da sie in der Bioinformatik von herausragender Bedeutung sind und Gendaten auch in der Biodiversitätsinformatik zunehmend wichtiger werden. Der Ansatz der Infrastruktur von Gendatenbanken ist auch deshalb interessant, da bei diesen ein völlig anderer Ansatz gewählt wurde als in GBIF (siehe Abschnitt 7.4.1). Da hier eine Infrastruktur betrachtet wird, die nicht direkt der Domäne der Biodiversitätsinformatik zugeordnet ist und Informationen über die Infrastruktur der Gendatenbanken nur eingeschränkt verfügbar sind, können die Gendatenbanken nicht in derselben Tiefe wie die anderen Infrastrukturen dieses Kapitels besprochen werden.

Der Austausch von Nucleotidsequencen ist seit 1987 über die 'International Nucleotide Sequence Database Collaboration' (INSDC) organisiert [33]. Die INSDC koope-

riert mit folgenden großen Gendatenbanken [33]:

- DNA Databank of Japan (DDBJ)
- European Molecular Biology Laboratory's European Bioinformatics Institute (EMBL-EBI)
- National Center for Biotechnology Information (NCBI)

Um eine neue Sequenz zu publizieren und in einem Paper zitieren zu können ist es erforderlich, diese zunächst bei einer dieser drei Datenbanken hochzuladen [33]. Dadurch erhält der Forscher eine ISNDC-Nummer, durch welche sein Datensatz erst zitierfähig wird [33]. Die Wissenschaftler verlieren durch diese Datenübertragung aber nicht die Rechte an ihren Daten sondern bleiben deren Eigentümer und der Zugriff auf die Datenbestände ist frei [33]. Die Datenbestände dieser drei Datenbanken werden über regelmäßigen Datenaustausch und ein gemeinsames Datenformat synchron gehalten [33].

Für eine globale Sicht auf die Daten wird von der INSDC die 'Feature Table Definition' (siehe [110]) geführt, welche zweimal jährlich aktualisiert wird [33]. Darüber hinaus werden auf [111] Dokumente zur Verfügung gestellt, welche ein kontrolliertes Vokabular definieren. Neben den eigentlichen Gendaten werden Daten zu den Experimenten über das 'Sequence Read Archive' (SRA) verwaltet [137]).

Evaluation: Gendatenbanken

Operationale Ebene: Die Infrastruktur der ISNDC-Gendatenbanken ist eine **geschlossene, dezentrale** Infrastruktur, auf Basis von **materialisierte Informationsintegration**. Die Die **technische Realisierung** ist nicht bekannt. Die **technische Partizipation** am Datenaustausch ist auf die drei großen Gendatenbanken beschränkt, welche von der ISNDC koordiniert werden. Die technische Partizipation für Repositorien und Wissenschaftler ist somit nur indirekt möglich. Die großen Gendatenbanken stellen aber ihre Daten einer Vielzahl von Repositorien zur Verfügung. Die technische Basis des Datenaustauschs konnte nicht ermittelt werden. Die **Heterogenität** und **Autonomie** der Datenspeicher kann aber als gering betrachtet werden.

Organisatorische Ebene: Die **Zitierfähigkeit** der Daten ist nicht nur gewährleistet, sondern der Upload der Daten bei einer Gendatenbank ist für die Domäne der Bioinformatik sogar für die Zitierfähigkeit obligatorisch. Die drei großen Gendatenbanken **partizipieren organisatorisch** über das 'INSDC Advisory Board'. Wissenschaftler haben keine Möglichkeit zur organisatorischen Partizipation. Die drei

großen Gendatenbanken tauschen enorme Datenmengen aus und müssen eine umfassende Infrastruktur unterhalten. Deshalb kann der **Aufwand** dieser Datenbanken als sehr hoch betrachtet werden. Da keine direkte Partizipationsmöglichkeit für Wissenschaftler besteht, verfügen diesen über keinen Partizipationsaufwand. Ein öffentliches **Ressourcenangebot** für die drei großen Gendatenbanken durch die INSDC konnte nicht ermittelt werden. Auch für Wissenschaftler scheint es nur ein vergleichsweise geringes Ressourcenangebot zu geben.

Funktionale Ebene: Die gemeinsame Sicht auf die Daten erfolgt über die 'Feature Table Definition'. Weitere Merkmale der funktionalen Ebene konnten nicht ermittelt werden.

Fazit: Gendatenbanken

Die Datenbestände werden im Gegensatz zu GBIF in zentralen Datenbanken gespeichert. Die Integration der Daten erfolgt über materialisierter Integration. Die Homogenität ist bei der in ISNDC organisierten Form des Datenaustauschs sehr hoch – Autonomie quasi nicht vorhanden. Die ISNDC-Architektur ist damit insbesondere an die Anforderungen der Bioinformatik gut angepasst, da in diesem Bereich sehr homogene Daten erhoben werden, was einen hohen Grad an Standardisierung ermöglicht. Dies ist in der Domäne der Biodiversitätsinformatik nicht der Fall, da hier die Daten an sich sehr heterogen strukturiert sind und die Repositorien sehr unterschiedliche Aufgaben wahrnehmen.

7.5 Zusammenfassung

Die Evaluation der Infrastrukturen ist in den Tabellen 7.4 bis 7.6 zusammengefasst. GBIF wird einmal als eigenständige Infrastruktur und einmal im Verbund mit dem IBF-Projekt evaluiert. Bei der Evaluation der funktionalen Kriterien wurde nach verschiedenen Datenarten unterschieden. Dementsprechend sind die Kriterien Funk2 und Funk6 für die verschiedenen Datenarten separat aufgeführt.

Das GBIF-Netzwerk ist die wichtigste Infrastruktur zum Datenaustausch in der Biodiversitätsinformatik und bietet im Gegensatz zu LTER und DataONE die Integration von Primärdaten an. Bei der Integration der Primärdaten treten enorme Datenverluste aufgrund der Mängel von ABCD und DwC als Austauschformat auf. GBIF ist in der Informationsintegration auf diese Formate beschränkt. Wie in Kapitel 4 gezeigt werden konnte, sind diese aber unvollständig und unflexibel. Darüber hinaus unterstützt GBIF Data Provenance nicht und die Identität von Daten ist

nicht ausreichend gesichert.

Die Infrastrukturen von LTER und DataONE haben den schwerwiegenden Nachteil, dass in diesen Primärdaten nicht integriert werden. Wichtige Standards der Biodiversitätsinformatik wie ABCD und DwC werden nicht unterstützt. Informationsintegration findet nur in geringem Ausmaß auf Ebene der Metadaten statt. Mit der Vorstellung der Gendatenbanken wurde gezeigt, dass es in der Praxis auch andere Ansätze gibt, um ein Netzwerk zum Datenaustausch zu organisieren. Auch wenn der in ISNDC verfolgte, zentrale Ansatz in der Biodiversitätsinformatik nicht umgesetzt werden kann, muss auch diese Infrastruktur berücksichtigt werden, da in der Biodiversitätsforschung die Bedeutung der Gensequenzierung zunimmt.

Als Ausgangspunkt für eine eigene Infrastruktur kommt damit nur die GBIF-Infrastruktur in Frage. Diese hat auch in der Biodiversitätsinformatik das notwendige Renommee für einen domänenspezifische Infrastruktur und ist auf organisatorischer Ebene gut aufgestellt. Allerdings weist GBIF in der funktionalen Ebene erhebliche Mängel auf. Im Bereich der Austauschschemata werden lediglich DwC und GBIF unterstützt. Die Evaluation dieser Datenstandards in Kapitel 4 ergab, dass diese die Anforderung der Biodiversitätsinformatik nicht erfüllen können. Dies ist die Ursache für die hohen Datenverluste in der GBIF-Infrastruktur. Darüber hinaus ist über das GBIF-Netzwerk aktuell die Integration von Multimediadaten nicht möglich und die Sicherung der Identität eines Datensatzes im Netzwerk ist nicht gewährleistet. Die Vergleichbarkeit von Primärdaten ist als gering einzustufen und Data Provenance wird nicht ausreichend unterstützt. Dementsprechend bedarf es einer Weiterentwicklung von GBIF auf der funktionalen Ebene. Diese wird mit BDEI in Kapitel 8 als neue Entwicklung vorgestellt.

Operationale Kriterien	GBIF	IBF ohne GBIF	IBF mit GBIF	LTER	DataONE	INSDC
Op1: Infrastrukturtyp	offen	offen	offen	offen	offen	geschlossen
Op2: Integrationsart	materiell, virtuell	materiell	materiell	materiell	virtuell	materiell
Op3: technische Realisierung	WS, JAVA, Python	.NET	.NET Python	Java	REST, RPC	N/A
Op4: Zentralität	dezentral mit zentraler Koordination	zentral	zentral	zentral	dezentral mit zentraler Koordination	dezentral
Op5r: technische Partizipation für Repositorien	direkt	keine	keine	direkt	direkti	ndirekt
Op5w: technische Partizipation für Wissenschaftler	indirekt	direkt	direkti	ndirekti	ndirekti	ndirekt
Op6: Autonomie	hoch	keine	keine	hoch	hoch	gering
Op7: Heterogenität	hoch	gering	gering	hoch	hoch	keine

Tabelle 7.4: Evaluationsergebnis: Operationale Kriterien

Organisatorische Kriterien	GBIF	IBF ohne GBIF	IBF mit GBIF	LTER	DataONE	INSDC
Org1: Zitierbarkeit	ja	nein	ja	ja	ja	ja
Org2r: organisatorische Partizipation für Repositorien	ja	nein	nein	ja	ja	nein
Org3r: Aufwand für Repositorien	hoch	-	-	hoch	hoch	-
Org3w: Aufwand für Wissenschaftler	gering	mittel	mittel	gering	gering	mittel
Org4r: Ressourcenangebot für Repositorien	hoch	-	-	gering	hoch	-
Org4w: Ressourcenangebot für Wissenschaftler	hoch	hoch	hoch	mittel	hoch	gering

Tabelle 7.5: Evaluationsergebnis: Organisatorische Kriterien

Funktionale Kriterien	GBIF	IBF ohne GBIF	IBF mit GBIF	LTER	DataONE	INSDC
Funk1: unterstützte Datenstandards und Austauschschemata	ABCD, DwC	DC, DM	DC, DM, ABCD	EML	EML, DublinCore, ...	ISNDC-Table, SRA
Funk2a: Integration von Primärdaten	ja	ja	ja	nein	nein	ja
Funk2b: Integration von Metadaten	ja	ja	ja	ja	ja	ja
Funk2c: Integration von Multimediadaten	nein	ja	nein	nein	nein	N/A
Funk3: Datenverluste	hoch	keine	hoch	keine	keine	N/A
Funk4: Identitätssicherung	optional	ja	optional	nein	ja	ja
Funk5: Aktualitätssicherung	nein	nein	nein	nein	ja	ja
Funk6a: Vergleichbarkeit von Primärdaten	gering	hoch	gering	keine	keine	hoch
Funk6b: Vergleichbarkeit von Metadaten	mittel	hoch	mittel	hoch	hoch	hoch
Funk7: Data Provenance	nein	gering	nein	ja	ja	N/A

Tabelle 7.6: Evaluationsergebnis: Funktionale Kriterien

Kapitel 8

Entwicklung einer Infrastruktur zum Datenaustausch in der Biodiversitätsinformatik

Im folgenden Kapitel wird mit der 'Biodiversity Data Exchange Infrastructure' (BDEI) eine Infrastruktur vorgestellt, mit welcher die Qualität des Datenaustauschs in der Biodiversitätsinformatik verbessert wird. Dazu wird als Ausgangspunkt auf die Infrastruktur des GBIF-Netzwerks zurückgegriffen. In Abschnitt 8.1 wird gezeigt, welche Mängel der GBIF-Infrastruktur beseitigt werden müssen, um den Anforderungen der Biodiversitätsinformatik gerecht zu werden und welche anderen Ansätze Einfluss auf die Entwicklung von BDEI hatten. In Abschnitt 8.2 werden anschließend die Grundelemente mit ihren Aufgaben in BDEI vorgestellt. In Abschnitt 8.3 werden die Designprinzipien von BDEI beschrieben und welche Ziele damit verfolgt werden. Die Umsetzung dieser Prinzipien in Prozesse einer Infrastruktur erfolgt in Abschnitt 8.4. Mögliche Erweiterungen und weitere Verbesserungen werden in Abschnitt 8.5 diskutiert.

8.1 Ausgangspunkt

In der Biodiversitätsforschung ist es erforderlich den Datenaustausch zwischen Wissenschaftlern, Repositorien und einer zentralen, koordinierenden Einheit (siehe Abbildung 7.9) zu ermöglichen. Da diese Infrastruktur erweiterbar sein muss, ist es wichtig diese offen zu gestalten. Wissenschaftler benötigen den Datenaustausch an sich, möchten aber dafür keine eigene Infrastruktur betreiben müssen. Somit sollten Wissenschaftler über den Anschluss an Repositorien indirekt an der Infrastruktur

teilnehmen. Da die Interessen in der Biodiversitätsforschung vielfältig sind, müssen verschiedene Arten des Datenaustausch verfügbar sein:

- Datenaustausch der Repositorien mit einem zentralem Repositorium
- Datenaustausch zwischen den Repositorien
- Export von Daten der kompletten Infrastruktur

Wissenschaftler sind jeweils einem lokalen Repositorium zugeordnet und speisen ihre Daten bei diesem ein. Diese Einspeisung erfolgt zwingend über materialisierte Informationsintegration in den Datenbestand des lokalen Repositoriums. Der Zugang der Wissenschaftler zu Daten wird über das zugeordnete Repositorium oder über ein beliebiges anderes Repositorium des Netzwerks sowohl in virtueller als auch in materialisierter Form ermöglicht.

Die lokalen Repositorien und das zentrale Repositorium stellen ihre Daten dem Netzwerk sowohl über materialisierte als auch virtuelle Informationsintegration zur Verfügung. Die Repositorien haben dabei die Möglichkeit ihren gesamten Datenbestand oder aber auch nur Teil zu publizieren, so dass ein Höchstmaß an Autonomie gewährleistet ist. Damit ergeben sich die in Abbildung 8.1 dargestellten potentiellen Datenflüsse. Die Organisation der Infrastruktur soll dabei analog zum GBIF-Netzwerk auf drei Stufen erfolgen. In Abbildung 8.1 ist dabei rechts das zentrale Repositorium dargestellt. Diesem entspricht in der GBIF-Infrastruktur die GBIF-Cachedatenbank, welche aus den GBIF-Knoten gespeist wird. Diese sind in Abbildung 8.1 abstrakt als 'Repository' bezeichnet. An diese exportieren Wissenschaftler ihren Datenbestand. Somit fließen die Daten von den Wissenschaftler über die Repositorien an das zentrale Repositorium. Dieses publiziert die Daten für die Allgemeinheit (GBIF-Portal in der GBIF-Infrastruktur). Zusätzlich besteht aber auch ein Bedarf am Datenaustausch zwischen den Repositorien. Dieser Bedarf entsteht durch Daten, die nur einem eingeschränkten Nutzerkreis zugänglich gemacht werden sollen oder aber so spezialisiert sind, dass diese nicht über das globale Austauschschema portiert werden können. Darüber hinaus können Repositorien ihren eigenen Datenbestand über ein proprietäre Portal publizieren. Die Vorteile diese Vorgehensweise liegen wiederum in der Kontrolle des Nutzerkreises und in einer erhöhten Autonomie bezüglich des Schemas der publizierten Daten.

In diesem Kapitel wird mit der 'Biodiversity Data Exchange Infrastructure' (BDEI) eine Infrastruktur eingeführt, welches diese Anforderungen an den Datenfluss erfüllt. Ausgangspunkt hierfür ist die Infrastruktur des GBIF-Netzwerks, da diese die wich-

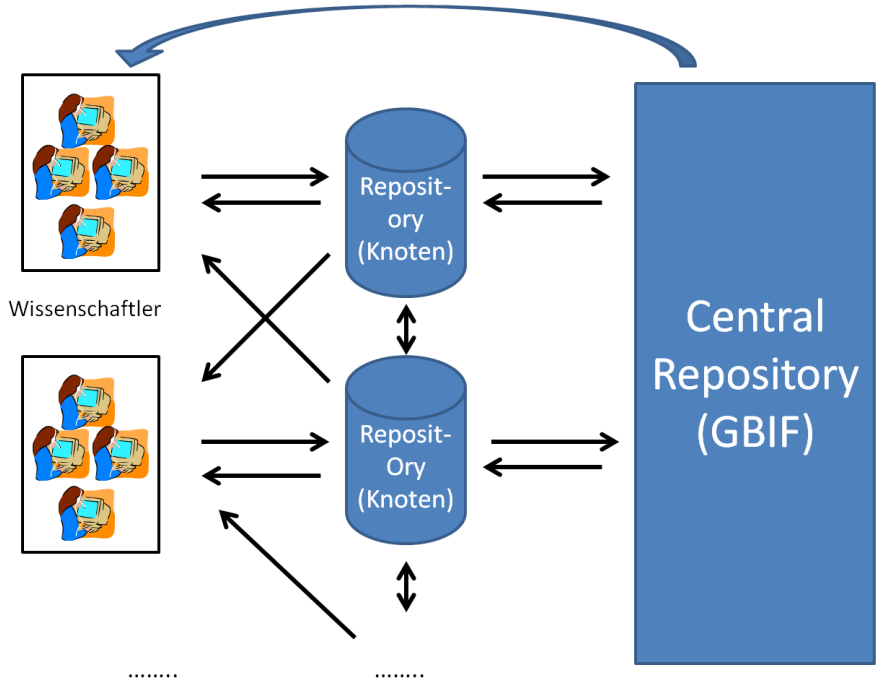

Abbildung 8.1: Anforderungen an den Datenfluss

tigste Infrastruktur in der Biodiversitätsinformatik ist. Allerdings verfügt die GBIF-Infrastruktur insbesondere auf der funktionalen Ebene (siehe Abschnitt 7.5) über signifikante Schwächen. Ziel der Architektur von BDEI ist es, folgende Schwächen der GBIF-Infrastruktur zu beseitigen:

- Starres Datenmodell
- Keine Identitätssicherung für Datensätze
- Kein direkter Datenaustausch zwischen Repositorien
- Mangelnde Unterstützung von Data Provenance
- Mangelnde Vergleichbarkeit von Daten
- Mangelnde Unterstützung von kuratorischen Prozessen

In BDEI werden dabei Lösungen für diese Probleme konzeptuell vorgestellt. Die technische Implementierung dieser Konzepte kann auf operationaler Ebene durch verschiedene Technologien aus Kapitel 6 umgesetzt werden. GBIF selbst setzt mit IPT einen erfolgversprechenden Ansatz auf Basis von objektorientierter Middleware in JEE (siehe Abschnitt 7.4.1) um. Alternativ ist auch eine vollständige Neuimplementierung des GBIF-Netzwerks auf Basis von RESTful-Services denkbar, welche

insbesondere beim Datenaustausch von großen Datenmengen eine bessere Performance bietet. Da eine konkrete Implementierung aber nur im Rahmen einer Projektumsetzung auf internationaler Ebene erfolgen kann, welche mehrere Jahre zur Umsetzung benötigt, werden im Folgenden nur die Konzepte zur Lösung der Probleme der GBIF-Infrastruktur vorgestellt.

Ein wichtiges Argument für GBIF als Ausgangspunkt sind die Qualitäten von GBIF organisatorischer Ebene mit einer umfassenden, internationalen Organisationsstruktur, welche in Biodiversitätsinformatik als Autorität akzeptiert ist. Außerdem stellt GBIF aktuell eine Infrastruktur zur Verfügung, in der die Anforderungen an BDEI zumindest ansatzweise erfüllt sind.

Die Architektur von BDEI ist außerdem von DaltON (siehe Abschnitt 6.3.3) beeinflusst. DaltON verfolgt beim Datenaustausch konsequent einen Prozessgedanken, welcher sich in der Unterteilung der datenorientierten Perspektive in eine Datenlogistik- und eine Datenintegrationskomponente beim Datenaustausch zeigt. Diese grundlegende Unterteilung wird für BDEI übernommen. Problematisch für die Umsetzung ist allerdings die Verwendung von Ontologien für den Prozess der Informationsintegration in der Biodiversitätsforschung, da dieser eine moderierende Ontologie voraussetzt. Diese kann aber nicht für die gesamte Domäne erstellt werden, so dass für den Prozess der Informationsintegration ein Datenmodell benötigt wird, welches sich im besonderen Maße durch Flexibilität auszeichnet. Darüber hinaus ist in der Biodiversitätsinformatik die Datenspeicherung in relationalen Datenbanken die übliche Form und die Expertise im Bereich der Ontologien bei Repositorien eher gering einzuschätzen, so dass die Erstellung von lokalen Ontologien mit einem hohem Personal- und Trainingsaufwand verbunden wäre.

Als Austauschformat soll das in Kapitel 5 vorgestellte PODSL-Biodiv verwendet werden, da dieses den Anforderungen an die Flexibilität eines Standards zum Datenaustausch gerecht wird. Die Vorteile von PODSL-Biodiv liegen insbesondere in der einfachen Erweiterbarkeit und der eindeutigen Identifizierbarkeit von Datensätzen und Konzepten des Metamodells. Dadurch kann PODSL-Biodiv so angepasst werden, dass der Datenaustausch zwischen zwei Repositorien dezentral erfolgen kann, ohne dass die Erweiterung von PODSL-Biodiv in der gesamten Infrastruktur verfügbar sein muss. Durch die eindeutige Referenzierung von Konzepten erhält PODSL-Biodiv bei der Datenübertragung die Semantik der Daten und ermöglicht so die Vergleichbarkeit der integrierten Daten.

Da die Software zum Datenaustausch für die Infrastruktur einheitlich sein muss, muss diese nach den Vorgaben der Zentrale erstellt und an die Repositorien verteilt

werden. Die Repositorien verfügen dabei über unterschiedliche Datenmodelle, für die jeweils ein Mapping nach PODSL-Biodiv erfolgen muss. Damit muss von der Zentrale eine Wrappersoftware zur Verfügung gestellt werden, welche die Daten nach außen hin in PODSE repräsentiert. Darüber hinaus muss ein Repositorium dazu in der Lage sein, der Infrastruktur Erweiterungen von PODSE mitzuteilen und die Auflösbarkeit von Referenzen auf den eigenen Datenbestand zu gewährleisten. Die Programmierung bzw. Auswahl dieser Software fällt damit in die Verantwortlichkeit einer zentralen Organisationseinheit zur Koordination.

Die Sicherung der Identität von Datensätzen ist eine zentrale Aufgabe einer Infrastruktur. Hierfür sind im Rahmen der TDWG mit LSID´s und UUID´s Standards definiert [233]. Ein alternativer Ansatz wird mit EZID in [163] publiziert. EZID soll die langfristige Archivierung von Datensätzen in Netzwerken wie LTER und DataONE garantieren und fordert von Identifiern elementare Eigenschaften wie Eindeutigkeit, Auflösbarkeit und Unveränderlichkeit [163]. In BDEI wird die Sicherung der Identität von Datensätzen über die Identifier in PODSL garantiert (siehe Abschnitt 5.3.4). Diese dienen einerseits der Identifikation von Modellelementen der M_1-Ebene und andererseits der Identifikation von konkreten Datensätzen auf der M_0-Ebene des PODSL Metamodells.

Da mit PODSL-Biodiv in BDEI ein flexibles Datenmodell verwendet wird, können bereits auf Ebene der Repositorien eigene Erweiterungen spezifiziert werden. Die Herkunft eines Datensatzes ist ein wesentliches Merkmal, um die Bedeutung des Datensatzes vollständig erfassen zu können.

Damit ist die Unterstützung von Data Provenance eng mit der Verwendung von Identifiern in BDEI verknüpft. Durch kuratorische Tätigkeiten und Informationsintegration können die Originaldaten modifiziert werden. Folglich ist es erforderlich, diese Veränderungen zu dokumentieren. Diese modifizierten Daten unterscheiden sich in einigen Punkten von den Originaldaten und können nicht mehr mit dem ursprünglichen Identifier aufgelöst werden. Folglich muss Data Provenance in BDEI neben der Datenherkunft auch die Verknüpfung zu den Originaldatensätzen garantieren.

8.2 Komponenten von BDEI

Als Ausgangspunkt für die Organisationsstruktur von BDEI dient die Architektur der GBIF-Infrastruktur, wie diese in Abbildung 7.9 dargestellt ist. In dieser Darstellung wird ein Netzwerk aus Wissenschaftlern, lokalen Repositorien, einer zentralen Koordinationseinheit, Datenportalen, einer Führungsstruktur und zentralen Services gebildet. Diese Komponenten sind auch in BDEI vorhanden und werden den Or-

ganisationseinheiten von BDEI eindeutig zugewiesen. BDEI besteht somit aus den folgenden, grundlegenden Komponenten:

- Zentrale
- Knoten
- Endnutzer (Wissenschaftler)

Diese grundlegende Unterscheidung wurde bereits für das GBIF-Netzwerk formuliert und ist auch in ähnlicher Form in Infrastrukturen wie DataONE oder LTER zu finden. In der konkreten Aufgabenverteilung wird aber in BDEI ein anderer Ansatz gewählt. Dieser wird in den folgenden Abschnitten vorgestellt.

8.2.1 Zentrale

Der Zentrale kommt die Koordination und Führung der Infrastruktur als Hauptaufgabe zu. Dadurch werden in der Zentrale die grundlegenden Entscheidungen für die Infrastruktur, wie die Wahl des globalen Datenmodells, getroffen. Im Fall von BDEI ist dies PODSL-Biodiv als domänenspezifische Sprache für die Datenmodellierung in der Biodiversitätsforschung. Erweiterungen von PODSL-Biodiv können dabei von der Zentrale erstellt und von allen Teilnehmern in BDEI übernommen werden. Die Hauptaufgaben der Zentrale sind:

- Betrieb eines zentralen Repositoriums
- Softwareentwicklung für die gesamte Infrastruktur
- Betrieb des zentralen Datenportals
- Betrieb von Services für die gesamte Infrastruktur
- Anbindung von Knoten

Das zentrale Repositorium ist dabei die zentrale Stelle für die materialisierte Informationsintegration. Knoten, die an das zentrale Repositorium exportieren, stellen dabei ihre Daten dem gesamten Netzwerk zur Verfügung. Wenn der Knoten dabei eine eigene Erweiterung von PODSL verwendet, können durch Rückgriff auf Basiskonzepte in PODSL-Biodiv die Daten mit minimalen Informationsverlust übertragen werden.

Die Software-Entwicklung ist eine Aufgabe der Zentrale. Anforderungen an die Entwicklung ist eine generische Implementierung, welche die Integration von Erweiterungen von PODSL-Biodiv ermöglicht. Dabei werden Softwareprodukte für die

Informationsintegration und die Veröffentlichung von Daten, sowie die Webservices zur Auflösungen von Referenzen im Netzwerk benötigt.

Das Datenportal der Zentrale gewährt Zugriff auf Daten des zentralen Repositoriums sowie auf virtuell angeschlossene Datenspeicher. In BDEI werden Daten über ein Webinterface publiziert, da dies dem Standard der Domäne entspricht. Das Datenportal der Zentrale steht dabei einem offenen Nutzerkreis zur Verfügung und ermöglicht neben den Download von Daten in PODSL den Download von konvertierten Daten in ABCD oder DwC.

Der Betrieb der Services für die gesamte Infrastruktur hat primär die Gewährleistung der Auflösbarkeit von Knoten und Datenbeständen der Repositorien von Knoten zum Zweck. Die Zentrale dient damit als Adressverzeichnis für angebundene Knoten im Netzwerk. Benötigt z.B. ein Knoten A die Adresse des Repositoriums von Knoten B, ist es Aufgabe der Zentrale diese dem Knoten A über einen Service mitzuteilen.

Eine weitere Aufgabe der Zentrale ist die Erweiterung des Netzwerks durch Anbindung neuer Knoten. Neben der Integration des Knotens in die organisatorische Struktur muss die zentrale die Referenzierbarkeit des Datenbestands und der Services eines Knotens über das Netzwerk gewährleisten.

8.2.2 Knoten

Knoten sind die mittlere Ebene in der Organisation von BDEI und koordinieren hierzu die Anbindung der Wissenschaftler. Dabei ist es wichtig, dass Wissenschaftler einen thematischen Bezug zu ihrem Knoten haben. Die Knoten werden somit analog zu GBIF regional nach Staaten und zudem thematisch nach Gruppen(Pflanzen, Pilzen, Ökologie,...) unterteilt.

Dabei verfügen Knoten über folgende Aufgaben:

- Anbindung von Wissenschaftlern
- Betrieb eines lokalen Repositorium
- Mapping von PODSL und Erweiterungen auf das eigene Datenmodell
- Spezifikation von PODSL-Erweiterungen
- Betrieb von Services, welche die Auflösbarkeit der Daten im lokalen Bestand und die Verwendung von lokalen Erweiterungen von PODSE garantieren
- Installation von gemeinsam genutzter Software

- Betrieb eines lokalen Datenportals (optional)
- Distribution gemeinsam genutzter Software an Wissenschaftler

Knoten besitzen ein lokales Repositorium zur Aufnahme der Daten von Wissenschaftlern. Sie sind in ihrer Entscheidung autonom, diesen Datenbestand dem Netzwerk in virtueller oder materialisierter Form zur Verfügung zu stellen und haben die Kontrolle über die selbst verwalteten Daten. Die Verpflichtung ein lokales Repositorium zu betreiben ist dabei aber nicht an ein bestimmtes Datenschema gebunden. Für die interne Struktur des lokalen Repositoriums ist jeder Knoten selbst verantwortlich und bindet den eigenen Datenbestand über eine Schnittstelle an die Infrastruktur an.

Da den Knoten das Schema der Datenspeicherung selbst überlassen ist, ist es Aufgabe der Repositorien ein Mapping des eigenen Datenbestands nach PODSL-Biodiv und möglichen Erweiterungen zu erstellen. Dabei werden die Repositorien durch die Software der Zentrale unterstützt.

Knoten haben das Recht, PODSL-Biodiv über lokale Erweiterungen an ihre eigenen Bedürfnisse anzupassen. Dies wird über das Prinzip der Vererbung in PODSL (siehe Abschnitt 5.3.6) ermöglicht, so dass die Daten stets unter geringst möglichen Informationsverlust über den Rückgriff auf eine Basisklasse zur Verfügung gestellt werden können. Damit die Daten vollständig ausgetauscht werden können, muss ein Knoten über den Referenzservice Erweiterungen von PODSL in der Infrastruktur publizieren. Analog dazu muss ein Knoten ein Mapping auf das eigene Datenschema vornehmen, sofern dieser eine Erweiterung eines anderen Knotens nutzen möchte.

Um entsprechende Erweiterungen und Daten zu publizieren, betreiben die Knoten Webservices. Datensätze sind im Netzwerk eindeutig identifiziert und geben darüber hinaus Auskunft über ihre Herkunft. Somit muss ein Knoten die Auflösbarkeit der Identifier des eigenen Datenbestands gewährleisten.

Knoten müssen keine eigene Software schreiben, sondern die Software der Zentrale an ihre Bedürfnisse anpassen und installieren. Dazu wird von der Zentrale Software für Services zum Zugriff auf den eigenen Datenbestand und zur Publikation von Erweiterungen von PODSL angeboten. Dabei liefert die Zentrale auch Software zum Betrieb eines Datenportals für Zugriff auf den Datenbestand eines lokalen Repositoriums. Der Betrieb eines Datenportals ist dabei für einen Knoten optional, da die Distribution der Daten bereits über das zentrale Datenportal gewährleistet ist. Der Betrieb eines lokalen Portals ermöglicht einem Knoten allerdings die Distribution von Daten, die nicht über das gesamte Netzwerk verteilt werden sollen an einen spezifischen Nutzerkreis, sowie die zielgenaue Verwendung von spezifischen Erweiterung von PODSL-Biodiv.

8.2.3 Wissenschaftler

Wissenschaftler treten sowohl als Datenproduzenten, wie als Datenkonsumenten in BDEI auf. Zur Teilnahme sind diese dazu verpflichtet, sich einem Knoten in BDEI anzuschließen. Die Wahl des Knotens ist dabei nach regionalen und thematischen Gesichtspunkten zu wählen. Dementsprechend sollte ein deutscher Fungologe auch dem BDEI-Knoten für Pilze in Deutschland zugeordnet sein.

Wissenschaftler müssen dazu keinen eigenen Datenspeicher betreiben, sondern verwenden zur Speicherung ihrer Daten das lokale Repositorium ihres Knotens. Dabei sollen die wissenschaftlichen Primärdaten vollständig an das Repositorium übertragen werden können. Dies wird durch die domänenspezifische Zuordnung von Wissenschaftlern an einen Knoten realisiert. So ist es Aufgabe der BDEI-Knotens für Pilze in Deutschland PODSL-Biodiv so zu erweitern, dass es den Anforderungen von Fungologen entspricht. Dadurch werden in PODSL-Biodiv die spezifischen Eigenschaften von Pilzen integriert.

Für die Datenübertragung an das lokalen Repositorium des zugeordneten Knotens wird dem Wissenschaftler Software zur Verfügung gestellt. Diese Software wird in der Zentrale programmiert und über die Knoten verteilt. Die Anpassung der Software an die Domäne erfolgt dabei auf der Ebene der Knoten. Der Wissenschaftler muss die Software lediglich installieren.

Die Aufgaben eines Wissenschaftlers lassen sich in folgender Weise zusammenfassen:

- Zuordnung zu einem Knoten
- Installation und Nutzung von Software der Zentrale
- Übertragung der eigenen Daten an das zugeordnete Repositorium

8.2.4 Überblick

Damit sind der Zentrale, Knoten und Wissenschaftlern eindeutig Aufgaben zugeordnet. Einen zusammenfassender Überblick über diese Aufgaben wird in Abbildung 8.2 gegeben. Die Aufgaben der Zentrale liegen in der Datenpublikation, der Datenhaltung, im Unterhalt von Web-Services und in der Erstellung eigener Softwareprodukte. Die Softwareprodukte werden von der Zentrale an die Knoten verteilt und von diesen installiert und angepasst. Die Distribution der angepassten Software erfolgt über die Knoten an die Wissenschaftler. Eine entscheidende Rolle innerhalb der Infrastruktur von BDEI spielen die Services. Services müssen zum Einen Referenzen auflösen und

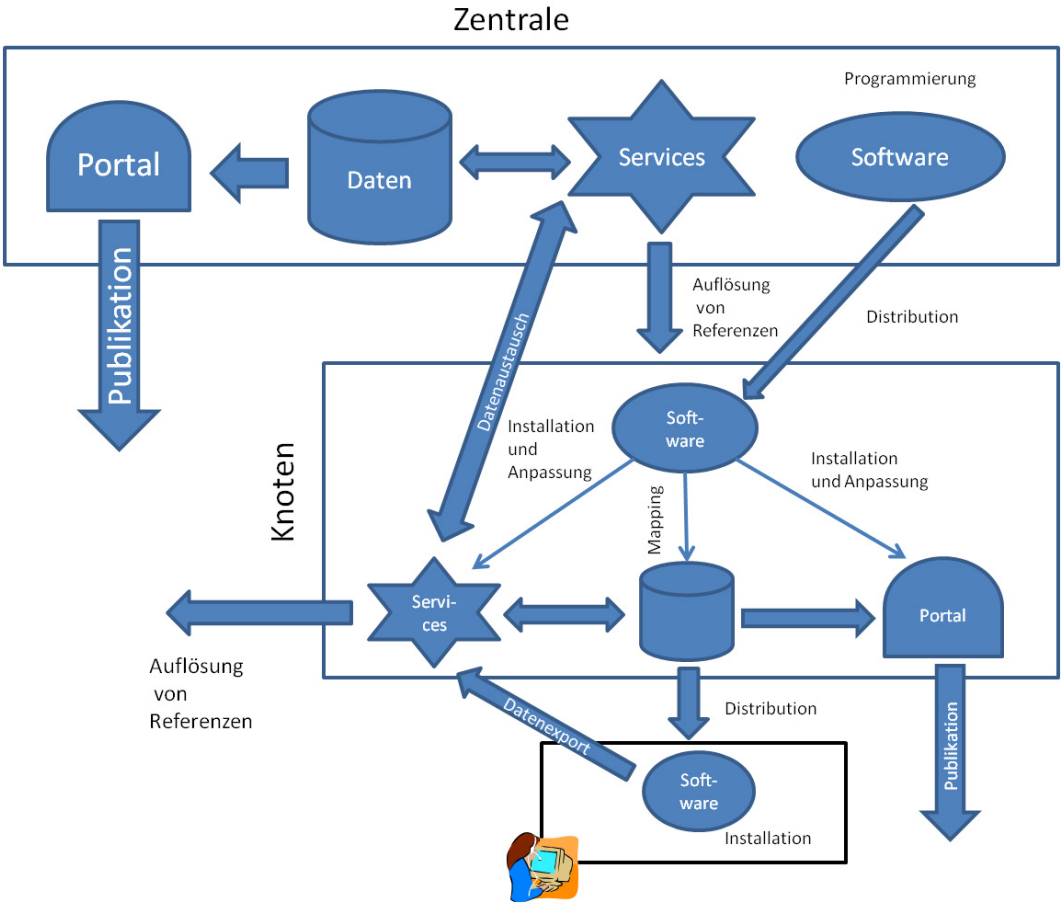

Abbildung 8.2: Aufgaben der Komponenten von BDEI

stellen damit die Referenzierbarkeit von Datensätzen in BDEI sicher. Zum anderen ist der gesamte Datenaustausch in BDEI über die Services organisiert. Die Ursache hierfür liegt in der Designautonomie der lokalen Repositorien der Knoten. In BDEI kann jeder Knoten ein eigenes Datenschema verwenden. Der Datenaustausch erfolgt über Schnittstellen in dem flexiblen Datenaustauschformat PODSL-Biodiv. Zusätzlich können Knoten eigene Datenportale betreiben. Da die Knoten in BDEI keine eigenen Softwareprodukte erstellen, wird diese von der Zentrale zur Verfügung gestellt.

Die zentrale Aufgabe der Knoten ist das erstellen von Mappings des eigenen Datenmodells auf PODSL-Biodiv. Dieser Vorgang sollte nach Möglichkeit durch Software, die von der Zentrale zur Verfügung gestellt wird, unterstützt werden. Dazu bieten sich Software für Schemamatching und Schemamapping an, die in der Literatur umfassend beschrieben werden. Der interessierte Leser sei hierzu auf [139] und Kapitel 6 verwiesen. Im Bereich des Schemamappings und Schemamatchings stehen

seit geraumer Zeit mit Rondo [161], Coma [53], Coma++ [53] und Cupid [148] generische Lösungen zur Verfügung, welche als Teil einer Softwarelösung einen Knoten bei dieser Aufgabe unterstützen können. Mit Quickmig [55] wurde sogar eine (semi-) automatische Lösung für diese Problemstellung publiziert. Darüber hinaus wurden in diesem Bereich in den letzten Jahren enorme Fortschritte erzielt [160] und es stehen mit ++Spicy [154] und OpenII [215] Programme zur Verfügung, die Opensource sind. Dementsprechend sind gute Ansätze vorhanden die Aufgabe des Schemamappings zu vereinfachen.

Ein weitere Aufgabe der Knoten liegt in der Auflösbarkeit von Referenzen auf den eigenen Datenbestand und der Spezifikation und Publikation von Erweiterungen von PODSL-Biodiv. Der Auftrag eines Knotens bei der Erweiterung von PODSL-Biodiv ist es, das erweiterte Schema über die Services zur Verfügung zu stellen.

Dem einzelnen Wissenschaftler obliegt die Installation von Software und die Übertragung der selbst erfassten Daten an das lokale Repositorium. Für die Publikation stellen die Zentrale und Knoten Portale zur Verfügung. Während die Publikation von Daten über die Zentrale nach PODSL-Biodiv mit akzeptierten Erweiterungen erfolgt, können Knoten Daten in spezifischen Erweiterungsformaten von PODSL-Biodiv veröffentlichen.

8.3 Aufgaben von BDEI

Um die verschiedenen Komponenten von BDEI zu einer Infrastruktur zu vereinen, muss es grundlegende Konventionen über das Design und Ziele der Infrastruktur geben. Um dies zu ermöglichen, werden im folgenden Abschnitt Designprinzipien in Form von Aufgaben vorgestellt, welche eine Infrastruktur wie BDEI erfüllen muss. Diese bilden die Grundlage für den Datenaustausch und der Informationsintegration in der gesamten Infrastruktur. Die Prinzipien sind hierbei im Folgenden:

- granulare Datenspeicherung und Vergleichbarkeit
- Identitätssicherung
- Auflösbarkeit von Referenzen
- Aktualität
- Verlustfreiheit
- Data Provenance

- Erweiterbarkeit
- Multimediaunterstützung

Ein wesentliches Element ist hierbei die **Datenspeicherung** in granularer Form. Infrastrukturen wie LTER oder DataONE speichern lediglich Metadaten und geben den Zugriff auf Primärdaten nur als Sammlungen (z.B. CSV-File oder Excel-Tabelle) in nicht standardisierter Form. Damit ist keine einheitliche Sicht auf die Primärdaten in diesen Netzwerken verfügbar. In BDEI soll deshalb eine tatsächliche Integration der Daten mit dem Ziel der Vergleichbarkeit von Datensätzen ausgeführt werden. Die Speicherung der Daten muss damit auf Ebene der Datenerhebung in granularer Form erfolgen. Dies bedeutet, dass jedes dokumentierte Objekt, wie z.B. eine einzelne Beobachtung einer Pflanze, auch ein separater Datensatz ist. Durch die Verwendung von PODSL-Biodiv in BDEI wird eine prozessorientierte Sicht auf die Daten gewählt, in welcher jede Ausführung eines Prozesses durch ein Dokument eindeutig beschrieben wird. Dabei sind alle Elemente dieses Dokuments über Identifier eindeutig gekennzeichnet und auflösbar. PODSL-Biodiv erlaubt somit über diesen Mechanismus die granulare Speicherung von Daten.

Die **Sicherung der Identität** eines Datensatzes ist eine zentrale Aufgabe einer Infrastruktur und eng verknüpft mit der Auflösbarkeit der Referenzen auf Datensätze, Data Provenance und der Granularität der Datenspeicherung. Wesentlich hierfür ist eine eindeutige Identifizierung für jeden Datensatz im gesamten Netzwerk, da sonst die Gefahr besteht Duplikate zu erzeugen und die Aktualität der Datensätze nicht gewährleistet ist (siehe Abschnitt 7.3.1, insbesondere Abbildung 7.6). Darüber hinaus muss für die virtuelle Informationsintegration und die Auflösbarkeit auch der Speicherort des Datensatzes aus dem Identifier erkenntlich sein. Wenn ein BDEI-Webservice zur Auflösung eines Identifiers aufgefordert wird, so muss dieser auflösen können, in welchem Repositorium der originale Datensatz gespeichert ist. Anschließend kann der BDEI-Webservice eine Anfrage zur Auflösung des entsprechenden Datensatzes an das Repositorium richten.

Damit gibt es für das Auflösen von Referenzen in BDEI zwei zentrale Aufgaben:

1. Auflösung der Referenz des Repositoriums
2. Auflösung der Referenz des eigentlichen Datensatzes in seinem Repositorium

Die erste Aufgabe ist dabei dem Web-Service der Zentrale zugewiesen. Dieser muss die Adressen aller Repositorien kennen und Anfragen diesbezüglich für die gesamte Infrastruktur beantworten. Die Web-Services der Knoten haben hingegen die

Aufgabe, Referenzen auf Datensätze im eigenen Repositorium für Anfragen aus der Infrastruktur aufzulösen. Grundlage für die Integrität der Daten ist hierbei die eindeutige Referenzierbarkeit der Daten im gesamten Netzwerk. Jeder Datensatz ist im gesamten Netzwerk über einen Identifier eindeutig gekennzeichnet. Der Originaldatensatz ist dabei nicht modifizierbar und kann dauerhaft im Netzwerk aufgelöst werden. Für die Veränderungen an Originaldaten werden neue Datensätze mit neuen Identifiern angelegt. Genauso werden Datensätze, die durch materialisierte Integration in der Infrastruktur entstehen, über neue Identifier gekennzeichnet. Der Bezug zu den Originaldaten wird über Data Provenance gesichert. Für die Garantie der Aktualität der Daten ist ein Prozess zur Sicherung der Aktualität vorgeschrieben.

Unter **Aktualität und Verlustfreiheit** ist die Lösung des Aktualitätsproblems aus Abbildung 7.7 und die Verhinderung von Datenverlusten im Sinn der Abbildungen 7.4, und 7.5 zu verstehen. Diese Eigenschaften hängen eng mit der Identität von Datensätzen zusammen und werden über die eindeutige Zuordnung der Datensätze zu lokalen Repositorien und der Auflösbarkeit der Datensätze in den lokalen Repositorien gelöst. Dementsprechend ist in BDEI garantiert, dass stets die Originalquelle eines Datensatzes identifiziert und von dort aus der Originaldatensatz wiederhergestellt werden kann.

Dies ist auch eine wichtige Voraussetzung für **Data Provenance**. Um diese vollständig zu erreichen, müssen Änderungen an Originaldaten als neuer Datensatz abgespeichert werden. Wird mit einem veränderten Datensatz weitergearbeitet, referenziert dieser seinen Originaldatensatz aus dem wiederum der ursprüngliche Originaldatensatz ermittelt werden kann. In PODSL werden diese Anforderungen an Data Provenance über das Konzept der 'ProvenanceTable' erfasst (siehe Abschnitt 5.3.7). Die Nennung der Identifier zusammen mit der Auflösbarkeit der Referenzen garantiert den Zugang zu den jeweiligen Versionen des Datensatzes.

Eine zusätzliche Eigenschaft von BDEI ist die **Flexibilität** des Datenmodells durch die Verwendung von PODSL-Biodiv und durch Erweiterungen dieses Datenmodells. Wesentlich hierbei ist die Möglichkeit der lokalen Repositorien spezifische Erweiterungen von PODSL-Biodiv in BDEI zu publizieren. Dies wird über die Web-Services der Knoten ermöglicht, welche lokale Erweiterungen von PODSL-Biodiv durch neue Konzepte ermöglichen. Die lokale PODSL-Variante ist allen Repositorien über BDEI zugänglich und muss von diesen an die eigene Datenstruktur angepasst werden.

Da die **Informationsintegration** auf granularer Ebene in eine gemeinsame (Meta-)Struktur erfolgt, kann auch die Vergleichbarkeit der einzelnen Datensätze hergestellt

werden. Dazu kann es erforderlich sein, kuratorische Schritte an den Originaldaten vorzunehmen. Durch die Unterstützung von Data Provenance, sind alle kuratorischen Schritte nachvollziehbar. Es können somit Daten aus verschiedenen Quellen zusammengeführt und gemeinsam ausgewertet werden. Diese Analyse generiert sekundäre Daten, welche mit den Quelldaten verknüpft sind, so dass die Basis der Analyse in BDEI nachvollziehbar bleibt.

Eine wichtige Aufgabe von BDEI ist der **Unterstützung von Multimediadaten**. Hierzu soll der Transport von Audio, Video und Bilddateien in gängigen Formaten über das Netzwerk unterstützt werden. Dazu müssen die Repositorien neben einer Struktur zur Speicherung von Primärdaten einen Fileserver unterhalten. Multimediadaten werden in diesen abgelegt und mit einem Identifier für das gesamte Netzwerk eindeutig gekennzeichnet. Analog zu der Kennzeichnung von Datensätzen enthält dieser Identifier die Information darüber in welchem Repositorium der Originaldatensatz zu finden ist. Das Repositorium muss über seinen Web-Service diesen Identifier auflösen können und die Multimediadaten auf Anfrage übertragen.

Mit diesen Prinzipien werden in BDEI die wesentlichen Anforderungen an eine Infrastruktur für die Biodiversitätsinformatik aus Abschnitt 7.3.1 erfüllt. Wie in Abschnitt 7.4.1 dargelegt wurde, muss GBIF als die derzeit wichtigste Infrastruktur in der Biodiversitätsinformatik verbessert werden. BDEI begreift sich entsprechend als ein Vorschlag, die GBIF-Infrastruktur so zu verbessern, dass alle Anforderungen, die für die Biodiversitätsinformatik relevant sind, erfüllt werden.

8.4 Design von BDEI als Infrastruktur für die Biodiversitätsinformatik

Im folgenden Abschnitt wird das Design von BDEI vorgestellt. Dazu werden Prozesse und Prinzipien vorgestellt, welche die Basis der Realisierung von BDEI bilden. Diese stehen in einem engen Zusammenhang zum Datenmodell PODSL-Biodiv. Grundlage für dieses Design ist eine prozessorientierte Sichtweise auf Grundlage des DaltON-Frameworks. Dieses unterteilt die datenorientierte Perspektive aus POPM in die Datenlogistik und die semantische Integration von Daten (siehe Abschnitt 6.3.3). Im Rahmen der Vorstellung Prinzipien von BDEI wird der Unterteilung in diese beiden Komponenten gefolgt und es werden Prozesse formuliert, welche diese Komponenten beim Austausch von Daten in BDEI unterstützen.

8.4.1 Identität

Die Identität eines Datensatzes ist Grundlage für die Auflösbarkeit seines Identifiers im Netzwerk. Dabei ist die Sicherung der Identität eines Datensatzes in einer Infrastruktur für die Referenzierbarkeit von Datensätzen (siehe Abschnitt 5.3.4) unabdingbar. Diese wird über die Verwendung von PODSL-Biodiv als Datenstandard garantiert. In PODSL-Biodiv ist der Identifier ein eigenständiges Element der M_2-Ebene. Durch die Metastruktur von PODSL sind alle Konzepten der M_1-Ebene und Datensätze der M_0-Ebene über einen PODSL-Identifier gekennzeichnet. Überdies ist nicht nur der Datensatz an sich sondern auch das Repositorium ermittelbar, in welchem dieser Datensatz gespeichert ist. Somit sind in BDEI alle Datensätze und Konzepte eindeutig identifizierbar.

8.4.2 Speicherung eines neuen Datensatzes in BDEI

Die Datenspeicherung in BDEI erfolgt über die Repositorien der Knoten und der Zentrale. Datensätze werden in BDEI im Allgemeinen von den Wissenschaftlern erzeugt und anschließend in BDEI zur Speicherung hinterlegt. Damit wird in dem Repositorium, an welches der Wissenschaftler angeschlossen ist, ein neuer Datensatz angelegt. Da dieses Repositorium das Erste ist, in welchem dieser Datensatz in BDEI verfügbar ist, wird dieses als das primäre Repositorium des Datensatzes bezeichnet.

Der Datensatz wird bei der Speicherung mit einem PODSL-Identifier gekennzeichnet und ist ab diesem Zeitpunkt unter diesem Identifier in BDEI bekannt. Gleichzeitig wird eine 'ProvenanceTable' für diesen Datensatz im zentralen Repositorium angelegt und der Datensatz mit dieser 'ProvenanceTable' verknüpft. Dieses bleibt dauerhaft mit dem Originaldatensatz verknüpft und dokumentiert alle materialisierten Datenübertragungen und Veränderungen an diesem Datensatz.

Neben der Speicherung der Daten an sich ist an dieser Stelle zu berücksichtigen, ob ein Datensatz über BDEI publiziert werden soll oder ausschließlich den Nutzern des Repositoriums vorbehalten bleiben soll. Soll der Datensatz öffentlich zugänglich sein, ist der Identifier des Datensatzes über den Resolutionserver verfügbar und wird von diesem bei Anfragen zurückgeben. Andernfalls gibt die Anfrage zurück, dass der Datensatz zwar bekannt, aber einem bestimmten Nutzerkreis vorbehalten ist.

Damit ist die Speicherung eines neuen Datensatzes ein kompositer Prozess (Abbildung 8.3). Im ersten Schritt gibt der Wissenschaftler den neuen Datensatz in einem Softwareprodukt von BDEI ein. Im IBF-Projekt sind dies DiversityCollection und DiversityMobile [52]. Anschließend übernimmt die Software die Aufgabe, den Datensatz im primären Repositorium zu speichern. Da dieses seinen Datenbestand über

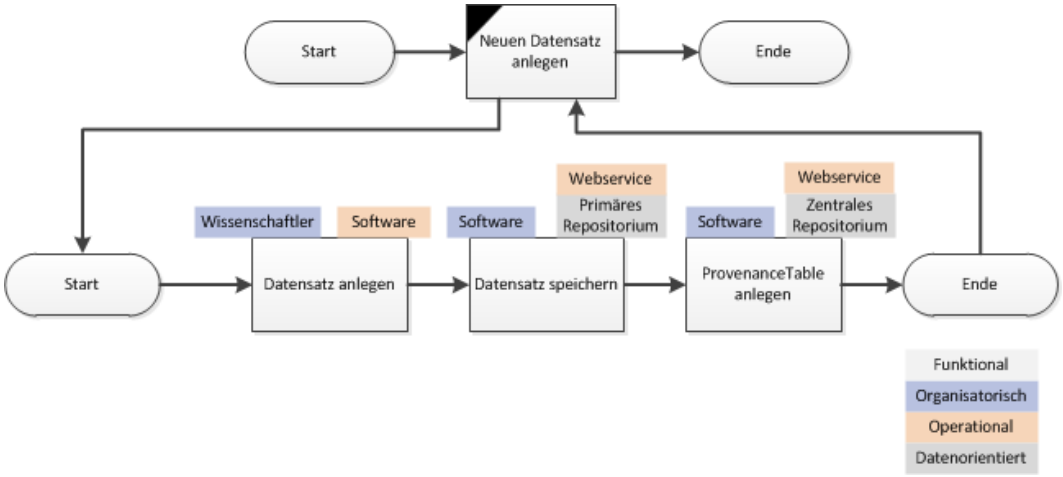

Abbildung 8.3: Prozess 'Neuen Datensatz anlegen' in einem Repositorium in BDEI

Web-Services kapselt, kommuniziert die Software mit dem Web-Service des Repositoriums. Im nächsten Schritt legt die Software eine 'ProvenanceTable' zu diesem Datensatz im zentralen Repositorium an.

Das Anlegen eines neuen Datensatzes ist der Data Logistics-Komponente von DaltON zugeordnet, da Daten einem Datenbestand hinzugefügt werden. Die semantische Integration ist diesem Prozess vorgelagert, weil die Software in der BDEI-Infrastruktur auf den Datenstandard PODSL-Biodiv ausgelegt ist. Somit sind die Daten beim Anlegen bereits semantisch kompatibel. Falls das Forschungsvorhaben eines Wissenschaftler nicht in PODSL-Biodiv abgebildet werden kann, muss dementsprechend das Modell entweder lokal oder global erweitert werden.

8.4.3 Data Provenance

Data Provenance bildet die Grundlage des Datenaustauschs in BDEI. Erst über die Dokumentation der Herkunft eines Datensatzes und alle Veränderungen ist es überhaupt möglich, die Duplikation von Daten bei der materialisierten Integration zu vermeiden und alle Versionen eines Datensatzes bis zum Originaldatensatz zurückverfolgen zu können. Grundlage für Data Provenance in BDEI ist die Unterstützung von Data Provenance in PODSL-Biodiv mit dem Konzept der 'ProvenanceTable'. Dieses verknüpft einen als Original ausgezeichneten Datensatz mit allen Versionen des Datensatzes und enthält Verweise auf alle Prozesse, in denen dieser oder aus diesem erzeugte Datensätze verändert wurden. Das Konzept der 'Provenance Table' ist in der Metastruktur von PODSL ein Element von M_1-Core und muss von allen Repositorien in BDEI unterstützt werden.

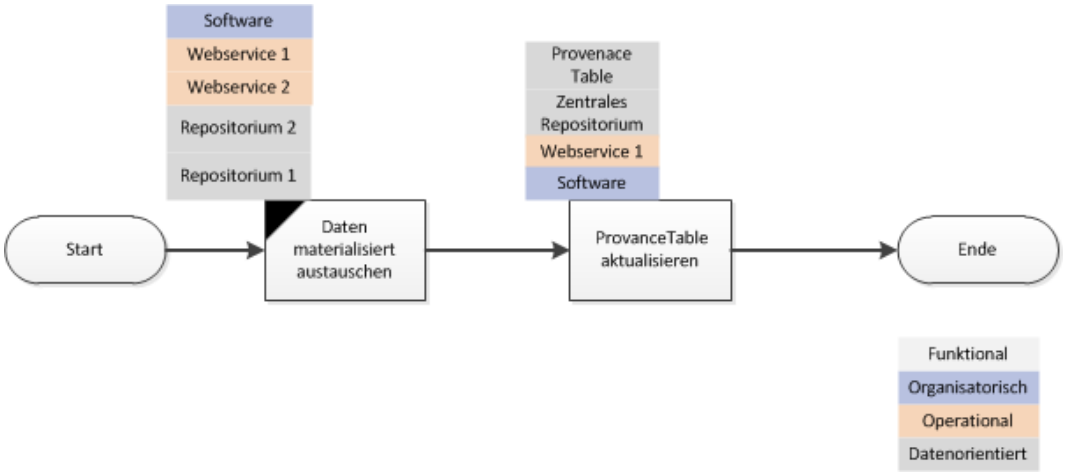

Abbildung 8.4: Prozess 'Data Provenance erhalten' bei materialisierter Informationsintegration

Die Informationen zur Data Provenance eines Datensatzes müssen für alle Teilnehmer von BDEI jederzeit zugänglich sein. Darüber hinaus ist es erforderlich, die 'ProvenanceTable' eines Datensatzes stets aktuell zu halten. Aufgrund dieser herausragenden Bedeutung der 'ProvenanceTable' werden diese im zentralen Repositorium gespeichert. Führt ein lokales Repositorium eine Veränderung an einem Datensatz aus, so muss dies in der 'ProvenanceTable' dieses Datensatzes im zentralen Repositorium dokumentiert werden, so dass alle Teilnehmer von BDEI über desselben Provenance Informationen für diesen Datensatz verfügen. Dies ist darüber realisiert, dass die 'ProvenanceTable' zentral gespeichert wird. Die lokalen Repositorien verfügen über keine Kopien der 'ProvenanceTable' sondern arbeiten mit der 'ProvenanceTable' des zentralen Repositoriums. Damit wird die 'ProvenanceTable' als einziges Element von PODSL ausschließlich von der Zentrale gespeichert. Die in der 'ProvenanceTable' referenzierten Datensätze können aber physisch über die gesamte Infrastruktur verteilt sein. Die Prozesse zur Unterstützung von Data Provenance sind damit den Data Logistics zugeordnet.

Dadurch müssen zur Sicherung von Data Provenance verschiedene Prozesse unterstützt werden, die zur Aufgabe haben die 'ProvenanceTable' eines Datensatzes aktuell zu halten. Diese treten als Subprozesse bei der materialisierten Informationsintegration und bei Veränderungen von Datensätzen auf (Abbildung 8.4). In diesem Prozess wird ein Datensatz aus einem Repositorium angefragt und dem Datenbestand eines anderen Repositoriums hinzugefügt. Auf diese Weise entsteht eine neue Version des Datensatzes mit einem neuen Identifier.

Die Veränderung eines Datensatzes ist in Abbildung 8.5 dargestellt. Die Verän-

derung eines Datensatzes ist ein Prozess und wird über ein 'ProcessExecutionDocument' in PODSL erfasst. Dabei bleibt die Originalversion des stets Datensatzes erhalten und es wird zusätzlich eine neue Version des Datensatzes im Repositorium gespeichert. Der Prozess der Veränderung und die neue Version des Datensatzes werden anschließend in der 'ProvenanceTable' des Originaldatensatzes verlinkt. Auf diese Weise bleibt die Verbindung zwischen einem Datensatz und allen abgeleiteten Datensätzen erhalten. Alle Veränderungen des Datensatzes sind dokumentiert.

Data Provenance betrifft unmittelbar Data Logistics, da alle Vorgänge der Data Logistics in BDEI über Data Provenance registriert werden. Die semantische Integration von Daten wird ebenfalls über Data Provenance aufgezeichnet, da alle Prozesse, welche Daten verändern, in der ProvenanceTable erfasst werden. Damit dokumentieren die Strukturen zur Unterstützung von Data Provenance den gesamten Datenfluss in BDEI.

8.4.4 Auflösen von Referenzen

Ein zentrale Ziel von BDEI ist die Auflösbarkeit von Referenzen. Diese wird zum Einen über die PODSL-Identifier und zum Anderen über die Resolutionserver der Knoten und der Zentrale garantiert. Das Auflösen von Referenzen findet dabei auf zwei Ebenen statt:

- Referenzen von Konzepten der Erweiterungen von PODSL-Biodiv
- Referenzen von Datensätzen

Für beide Arten von Referenzen werden PODSL-Identifier verwendet. Aus diesen lässt sich der Speicherort des Konzepts bzw. Datensatzes ausfindig machen. Der Prozess des Auflösens einer Referenz ist in Abbildung 8.6 dargestellt. Ausgangspunkt ist ein Web-Service in BDEI, welcher die Auflösung der Referenz eines PODSL-Identifiers benötigt. Aus diesem wird zunächst der Resolutionserver ermittelt, welcher über die Kennung des Instituts in den PODSL-Identifier codiert ist. Ist dieser nicht direkt bekannt, kann dieser über den Resolutionserver der Zentrale aufgelöst werden. Der Resolutionserver des Repositoriums hat anschließend die Aufgabe, den Datensatz im eigenen Datenbestand zu identifizieren. Als Ergebnis dieses Prozesses wird dadurch entweder das Konzept oder aber der Datensatz zurückgegeben.

Der Prozess des Auflösens einer Referenz ist ein Teilprozess in anderen Prozessen wie dem Datenaustausch (siehe Datenaustausch in Abschnitt 8.4.7) oder der semantischen Integration einer Modellerweiterung von PODSL-Biodiv (siehe Abschnitt

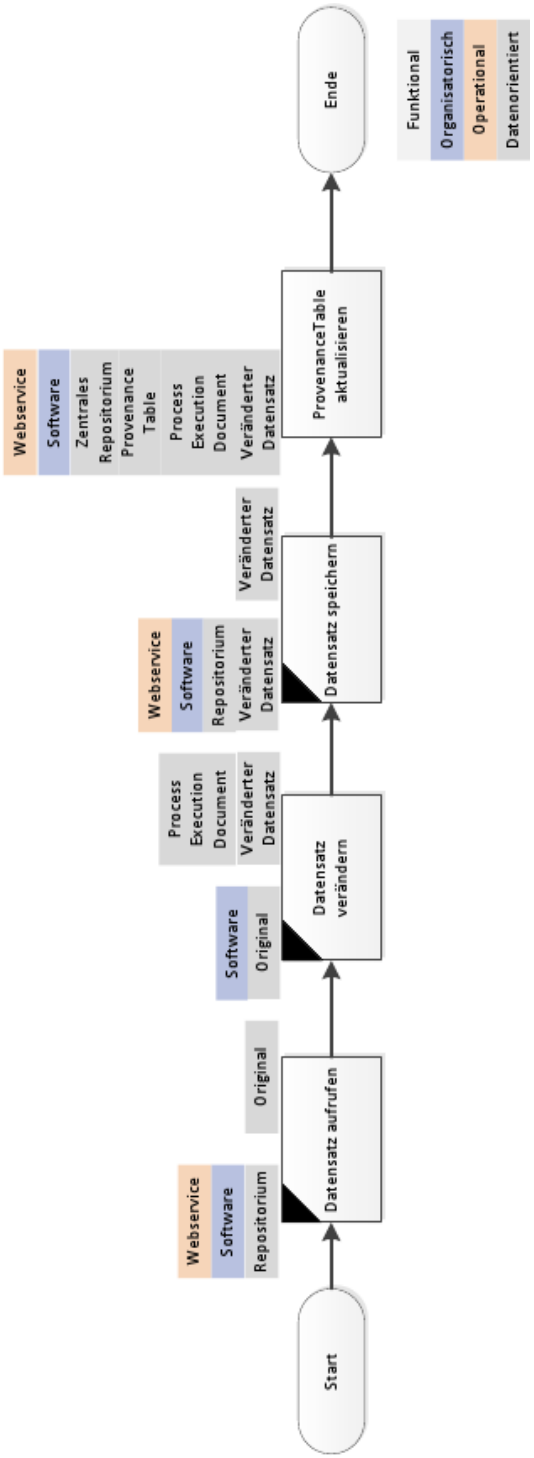

Abbildung 8.5: Prozess 'Datensatz verändern' in BDEI

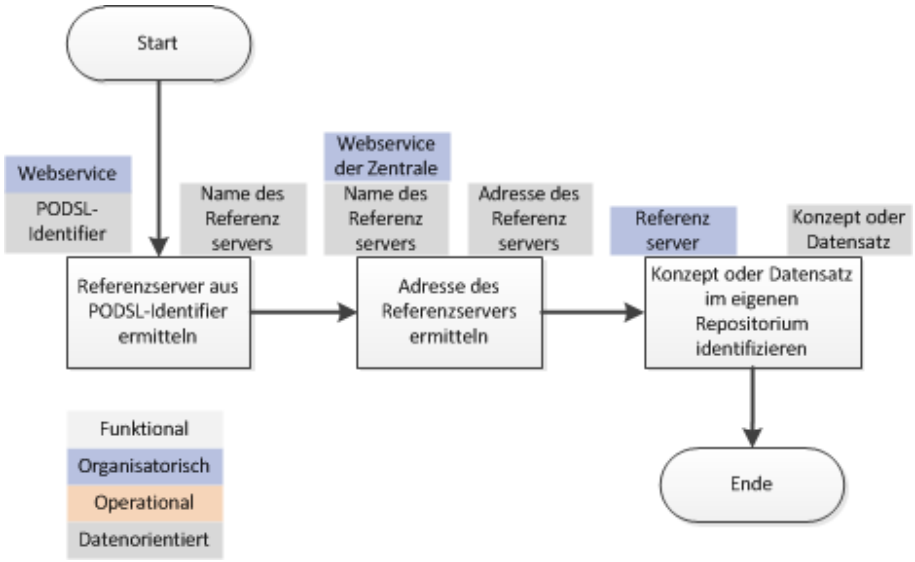

Abbildung 8.6: Prozess 'Referenz auflösen' in BDEI

8.4.5). Das Auflösen einer Referenz ist damit Grundlage für den Datenaustausch und im Bereich der Data Logistics Grundlage der semantischen Integration von Daten.

8.4.5 Berücksichtigung von Modellerweiterungen

In einer komplexen Domäne wie der Biodiversitätsforschung sind Modellerweiterungen zur Sicherstellung des langfristigen Betriebs einer Infrastruktur unumgänglich. Diese werden in BDEI über Erweiterungen von PODSL-Biodiv durch Vererbung realisiert (siehe Abschnitt 5.3.6). Dabei kann eine Modellerweiterung von PODSL-Biodiv auf zwei verschiedene Arten erfolgen:

- Globale Erweiterung des Gesamtmodells von PODSL-Biodiv durch die Zentrale
- Bildung einer lokalen Erweiterung in einem Knoten

Eine globale Erweiterung des Datenmodells von PODSL-Biodiv betrifft alle lokalen Repositorien. Die Knoten, welche diese Repositorien betreiben, müssen bei einer entsprechenden Erweiterung die Abbildung der neuen Konzepte von PODSL-Biodiv auf ihre interne Datenstruktur umsetzen. Auch wenn ein Knoten eine Erweiterung nicht sofort abbilden kann, kann dieser Knoten weiterhin an BDEI teilnehmen. In diesem Fall garantiert das Prinzip der Vererbung, dass durch den Rückgriff auf Basiskonzepte der Datenaustausch in BDEI weiterhin möglich ist.

PODSL-Biodiv deckt die grundlegende Anwendungsfälle der Biodiversitätsinformatik ab. Da die Domäne der Biodiversitätsinformatik sehr vielfältig ist, kann aber

nicht garantiert sein, dass jeder denkbare Anwendungsfall von PODSL-Biodiv erfasst wird. Diese Anwendungsfälle sind allerdings häufig Spezialfälle, die nur für einen oder wenige Knoten von Bedeutung sind. Dementsprechend wäre es zu aufwändig, diese direkt in PODSL-Biodiv als globale Erweiterung zu integrieren. Um diese zusätzlichen Sachverhalte abbilden zu können, kann ein Knoten in BDEI deshalb eine lokale Erweiterung spezifizieren.

Bei einer lokalen Erweiterung von PODSL-Biodiv wird die Erweiterung von einem Knoten vorgenommen und ist nicht für die gesamte Infrastruktur von BDEI gültig. Dazu wird auf Grundlage der Vererbung ein neues Konzept in einem lokalen Repositorium gebildet und dies von dem betreibenden Knoten auf die interne Datenstruktur abgebildet. Auf diese Weise wird es in BDEI möglich, lokal Spezialisierungen von PODSL-Biodiv zu erstellten. Damit Datensätze in spezialisierten Konzepten in BDEI ausgetauscht werden können, ist es notwendig, diese lokale Erweiterung in BDEI zu publizieren. Dies ist Aufgabe des Knotens, welcher die Erweiterung vorgenommen hat. Wird ein Datensatz eines spezialisierten Konzepts über BDEI ausgetauscht, gibt es zwei Möglichkeiten, wie der Betreiber eines Repositoriums mit einem solchen Datensatz umgehen kann:

- Erstellung eines Mappings des spezialisierten Konzepts auf das eigene Datenmodell
- Rückgriff auf ein Basiskonzept

Da das Prinzip der Vererbung die Ableitung neuer Konzepte aus Basiskonzepten zwingend vorschreibt, ist der Rückgriff auf ein Basiskonzept immer möglich. Allerdings können in diesem Fall die Felder des neuen Konzeptes beim Datenaustausch nicht berücksichtigt werden. Der Betreiber eines Repositoriums hat damit eine grundlegende Entscheidung zu treffen. Der erste Fall erzeugt durch die Erstellung des Mappings und der Erweiterung der eigenen Datenstruktur einen gewissen Aufwand. Im zweiten Fall kann das Repositorium allerdings nicht die zusätzlichen Attribute des spezialisierten Konzepts erfassen und es entstehen Datenverluste. Der Umgang mit lokalen Erweiterungen in BDEI ist nach dem Prozess aus Abbildung 8.7 strukturiert.

Ausgangspunkt für diesen Prozess ist ein Web-Service eines Repositoriums, welcher als Ergebnis des Auflösens einer Referenz ein neues Konzept erhält. Um einen Ausgangspunkt für die Verarbeitung von Daten zu haben, wird zunächst das Basiskonzept des neuen Konzeptes identifiziert. Dieses kann bereits Teil von PODSL-Biodiv oder aber eine unbekannte Erweiterung sein. Ist das Basiskonzept ebenfalls

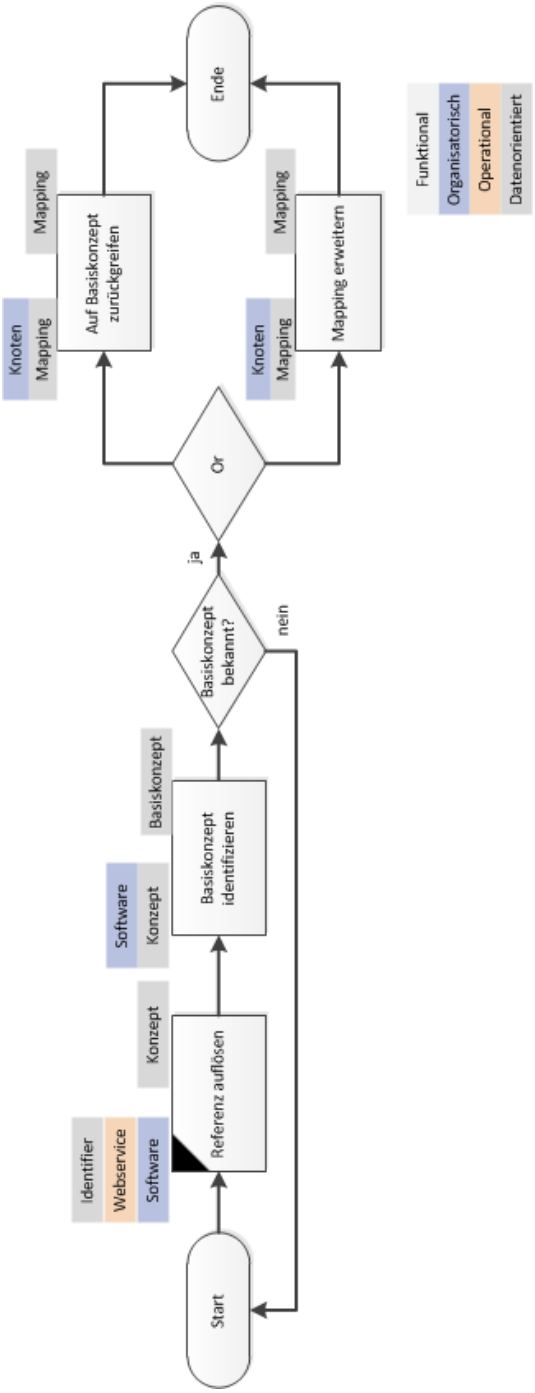

Abbildung 8.7: Prozess 'Lokale Erweiterungen berücksichtigen' in BDEI

Teil einer unbekannten lokale Erweiterung, muss zunächst dieses Basiskonzept aufgelöst werden. Folglich wird der Prozess erneut angestoßen. Andernfalls muss der Knoten des aufrufenden Web-Services festlegen, wie er mit dem neuen Konzept umgehen möchte. Dazu kann er zum Einen ein Mapping des neuen Konzeptes auf die eigene Datenstruktur erstellen oder aber auf das Basiskonzept zurückgreifen und dies für Daten des neuen Konzeptes in seinem Mapping auf den eigenen Datenspeicher vermerken. Die erste Variante hat dabei den Nachteil des höheren Aufwands, wohingegen bei der zweiten Variante Datenverluste bei der Übertragung entstehen können. Dementsprechend müssen hier im Einzelfall die Vor- und Nachteile beider Varianten abgewogen werden. Damit garantiert die Berücksichtigung von Modellerweiterungen die semantische Integration von Konzepten aus lokalen Erweiterungen von PODSL-Biodiv.

8.4.6 Sicherung der Aktualität

Innerhalb von BDEI wird sowohl virtuelle als auch materialisierte Informationsintegration unterstützt. Im Rahmen der materialisierten Informationsintegration kann das Aktualitätsproblem (siehe Abbildung 7.7) auftreten, welches dadurch entsteht, dass ein veralteter Datensatz mehrfach durch materialisierte Informationsintegration innerhalb einer Infrastruktur übertragen wird. Die Umsetzung von Data Provenance in BDEI bietet hierbei die Möglichkeit, das Aktualitätsproblem zu lösen. Dies geschieht auf Basis folgender Prinzipien:

- Referenz auf die Originaldaten in der 'ProvenanceTable'
- Speicherung von Veränderung von Daten in der 'ProvenanceTable'

Da die Originaldaten in BDEI erhalten bleiben, kann diese Referenz stets in der 'ProvenanceTable' aufgelöst werden. Darüber hinaus lassen sich auch Aktualisierungen eines Originaldatensatzes aus der ProvenanceTable ermitteln. Grundlage hierfür sind die über die 'ProvenanceTable' verknüpften 'ProcessExecutionDocument's, in welcher die Art und Weise der Veränderung eines Datensatzes als Prozessaufführung dokumentiert ist. PODSL-Biodiv ermöglicht es dabei, zwischen Prozessen zur Umwandlung von Datensätzen zur Anpassung an einen Datenspeicher und zu Prozessen zur Aktualisierung von Datensätzen zu unterscheiden. Zur Sicherung der Aktualität eines Datensatzes sind letztere Prozesse von Bedeutung.

Dementsprechend wird bei der materialisierten Integration von Daten in BDEI die Sicherung der Aktualität, wie in Abbildung 8.8 dargestellt ist, als Zwischenschritt eingeführt. Ziel dieses Prozesses ist es, die jeweils aktuelle Version eines Datensatzes

zu ermitteln. Das geschieht durch Auswertung der 'ProvenanceTable' eines Datensatzes. Ist in dieser der Prozess einer Aktualisierung eines Datensatzes verzeichnet, wird über den Output dieses Prozesses die aktuelle Version zurückgeben. Andernfalls ist die aktuelle Version des Datensatzes der Originaldatensatz. Damit ist der Output des Prozesses der Sicherung der Aktualität eine Referenz auf die aktuelle Version des Datensatzes. Diese kann dann im Rahmen eines materialisierten Datenaustausch verwendet werden.

Durch den Prozess zur Sicherung der Aktualität von Daten erfolgt ein wesentlicher Beitrag zu Data Logistics in BDEI, da hierdurch garantiert wird, dass stets die richtige Version im Rahmen eines materialisierten Datenaustauschs übertragen wird.

8.4.7 Datenaustausch

Beim Datenaustausch in BDEI müssen alle Datenflüsse aus Abbildung 8.1 berücksichtigt werden. Das heißt, dass in BDEI sowohl virtuelle als auch materialisierte Informationsintegration unterstützt werden muss. Dabei tritt PODSL-Biodiv als Austauschschema zwischen Datenspeichern auf, die eine völlig unterschiedliche interne Datenstruktur aufweisen können. Beim Laden von Daten aus einem Datenspeicher müssen die Daten zunächst nach PODSL-Biodiv übertragen und serialisiert werden. Für PODSL-Biodiv stehen Repräsentationen für verschiedene Technologien zur Verfügung (siehe Abschnitt 5.5.2). Dabei wird für den Datenaustausch die Repräsentation von PODSL-Biodiv in XML verwendet (siehe Abschnitt 5.5.3). Folglich werden in BDEI Daten im logischen Schema von PODSL-Biodiv über XML übertragen.

Der Datenaustausch in BDEI ist ein kompositer Prozess, in dem mehrere Prozesses dieses Kapitels als Subprozesse auftreten. Ausgangspunkt für diesen Prozess ist ein Repositorium und eine Referenz auf einen Datensatz. Ziel des Prozesses ist es, diesen Datensatz an das Repositorium zu übertragen. Im Fall der materialisierten Informationsintegration soll der Datensatz zusätzlich in den lokalen Datenbestand des Repositoriums integriert werden. Im Fall der virtuellen Datenübertragung wird der Datensatz zwar nicht dem lokalen Datenbestand hinzugefügt, aber über das Portal des Knotens des Repositoriums angefordert oder von Software am Knoten des anfragenden Repositoriums benötigt. Ob virtuelle oder materialisierte Informationsintegration ausgeführt wird, wird vom Knoten des anfragenden Repositoriums festgelegt. Dabei ist in beiden Fällen zu beachten, dass die aktuelle Version des Datensatzes geladen wird und dieser in einer lokalen Erweiterung von PODSL-Biodiv spezifiziert sein kann. Im Falle der materialisierten Informationsintegration entsteht zusätzlich eine neue Version des Datensatzes, welche in der 'ProvenanceTable' verzeichnet wer-

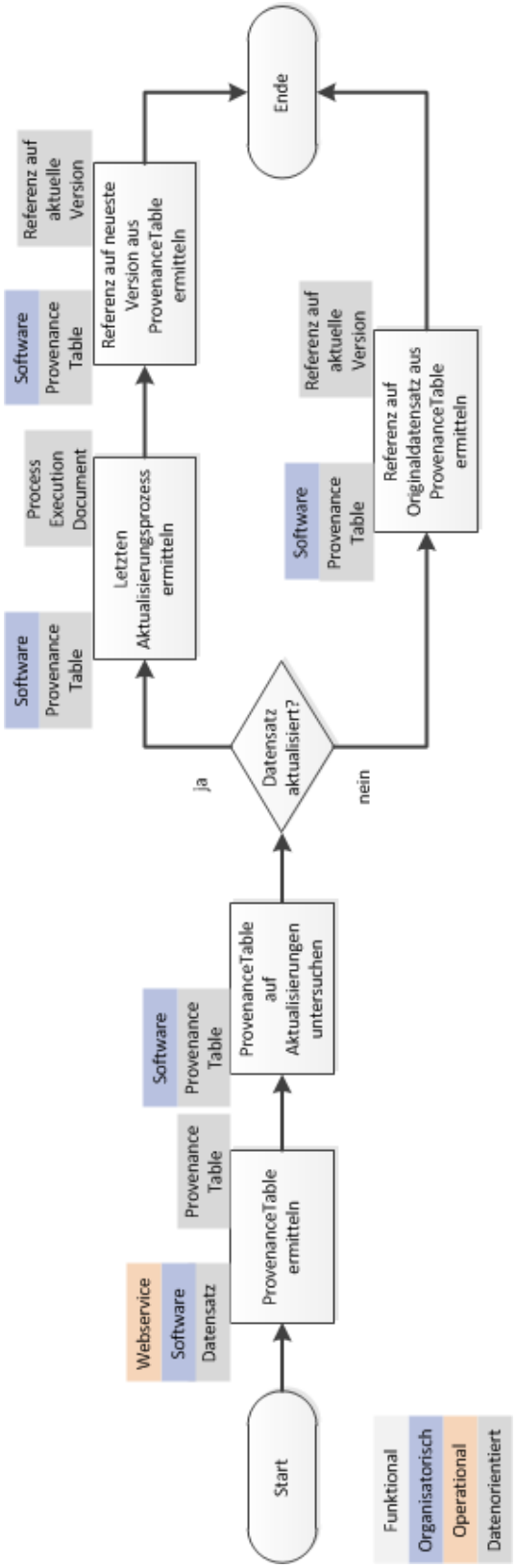

Abbildung 8.8: Prozess 'Aktualität sichern' in BDEI

den muss.

Der Prozess des Datenaustausch über BDEI ist für die virtuelle Informationsintegration in Abbildung 8.9 und für die materialisierte Informationsintegration in Abbildung 8.10 dargestellt. Grundlage für beide Prozesse ist zunächst die Auflösung des Identifiers des Datensatzes, der transferiert werden soll. Dies stellt sicher, dass der Datensatz in BDEI existiert. Darüber hinaus wird über diesen Prozess das PODSL-Konzept ermittelt, in welchem der Datensatz gespeichert ist. Danach muss unterschieden werden, ob dieses Konzept im Knoten, dessen Web-Service die Anfrage erstellt, bekannt ist. Ist das Konzept unbekannt, muss der Prozess der Berücksichtigung von lokalen Erweiterungen ausgeführt werden, damit die semantische Integration des Datensatzes gewährleistet ist. Anschließend wird der Prozess zur Sicherung der Aktualität ausgeführt. Durch diesen Prozess wird der Speicherort der aktuellen Version des Datensatzes ermittelt und anschließend im entsprechenden Repositorium angefragt. Da der Knoten dieses Repositoriums über Autonomie bezüglich seiner internen Speicherstruktur verfügt, kann der Datensatz in einem beliebigen Schema im internen Speicher des Repositoriums vorliegen. Folglich muss der Datensatz an dieser Stelle nach PODSL-Biodiv konvertiert werden. Dazu stellt betreibt der Knoten des Repositoriums, das den Datensatz speichert, einen Wrapper, in welchem der Knoten die Mappinginformationen für die Transformation von Daten von seiner internen Speicherstruktur nach PODSL-Biodiv hinterlegt hat. Mit Hilfe diese Wrapper werden die Daten aus dem lokalen Format in die XML-Repräsentation des Datensatzes von PODSL-Biodiv konvertiert. In dieser Form kann der Datensatz über Web-Services in BDEI übertragen werden.

Wird ein virtuelle Datenaustausch vorgenommen, kann diese XML-Repräsentation anschließend von Software oder einem Portal im anfragenden Knoten konsumiert werden. Damit ist im virtuellen Fall der Datenaustausch beendet. Im materialisierten Fall muss der Datensatz in das lokale Format konvertiert werden, da auch der Datenspeicher des anfragenden Knotens über Autonomie bezüglich seiner internen Struktur verfügt. Dazu steht analog zum Datenexport im anfragenden Knoten ein Wrapper mit Mappinginformationen zur Verfügung. Anschließend wird der Datensatz gespeichert. Dadurch entsteht eine neue Version des Datensatzes, die in der 'ProvenanceTable' verlinkt werden muss.

Im Rahmen des Datenaustausch in BDEI tritt sowohl semantische Integration als auch Data Logistics auf. Ausgangspunkt für die semantische Integration ist, dass jeder Datenspeicher in BDEI Autonomie über seine interne Datenstruktur benötigt. Das heißt, dass beim Export von Daten, diese zunächst über einen Wrapper

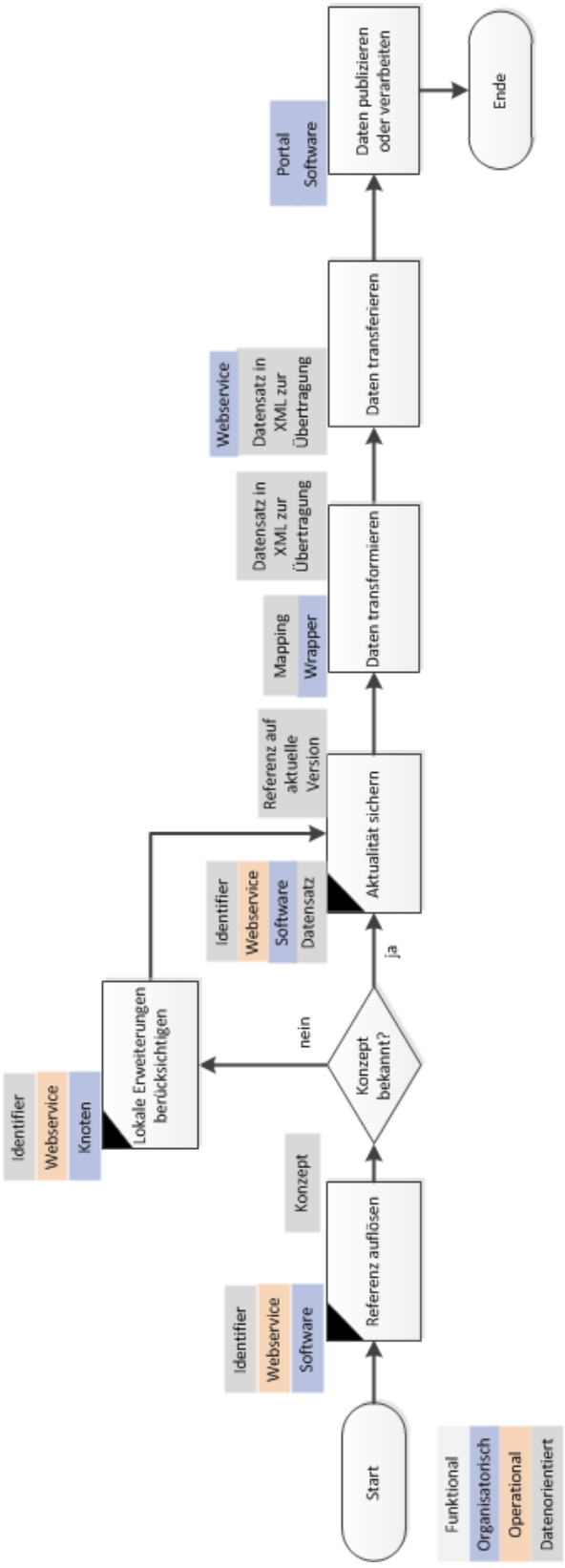

Abbildung 8.9: Prozess der virtuellen Datenübertragung in BDEI

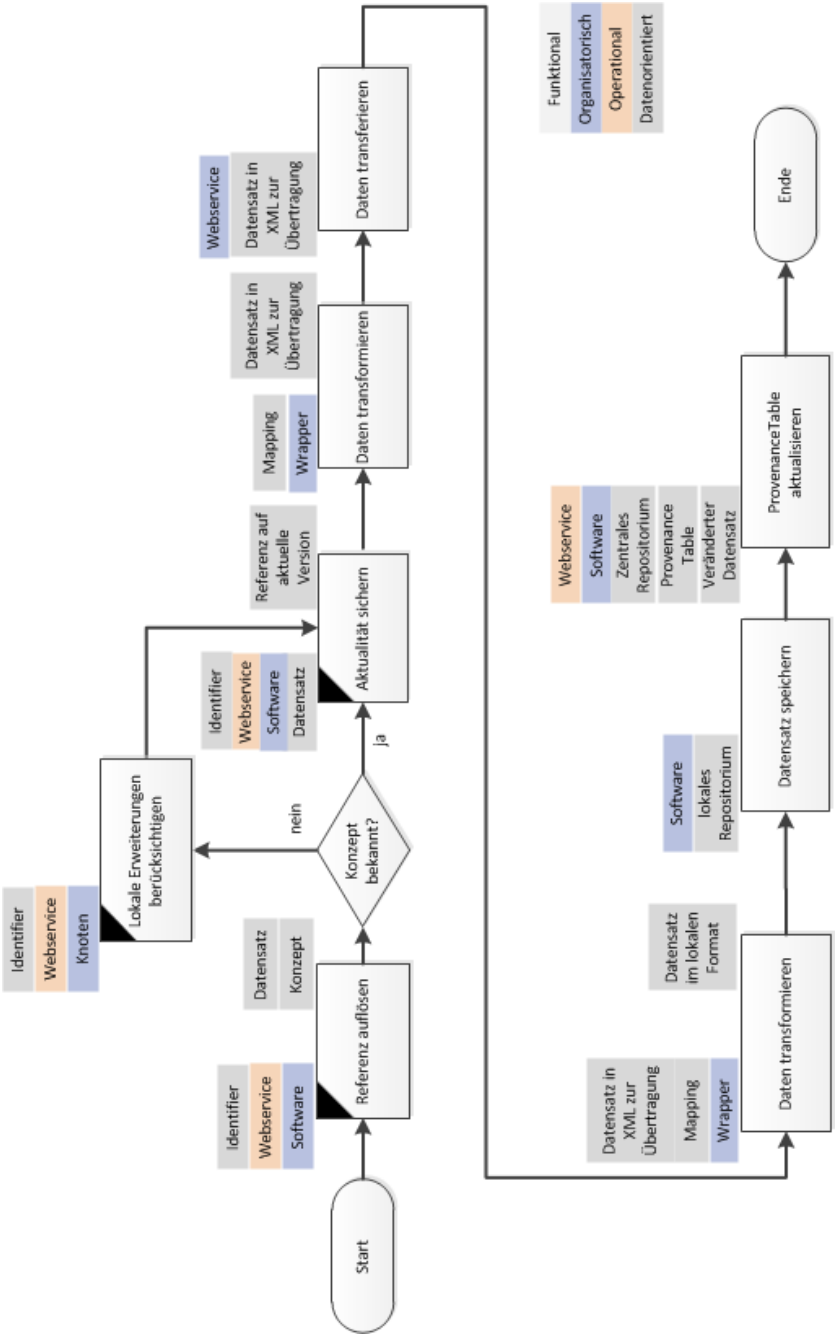

Abbildung 8.10: Prozess der materialisierten Datenübertragung in BDEI

in die Struktur von PODSL-Biodiv übertragen werden müssen. Es ist Aufgabe eines Knotens, für diesen Prozess ein Mapping zwischen der internen Datenstruktur seines Repositoriums und PODSL-Biodiv zu erstellen. Beim Import der Daten müssen diese dann umgekehrt über den Wrapper und das Mapping des Repositoriums von PODSL-Biodiv auf die eigene Struktur abgebildet werden. Zusätzlich können die Daten in einer Variante exportiert werden, welche eine lokale Erweiterung von PODSL-Biodiv ist. In diesem Fall wird die semantische Integration über den Prozess aus Abbildung 8.7 vorgenommen.

An dieser Stelle wird deutlich, wie wichtig Data Provenance in BDEI für die Minimierung von Datenverlusten ist. Datenverluste können in BDEI dadurch entstehen, dass ein Knoten eine Modellerweiterung von PODSL-Biodiv im eigenen Repositorium nicht vollständig unterstützt oder aber das Mapping von PODSL-Biodiv auf die eigene Datenstruktur unvollständig oder fehlerhaft ist. Dazu wird für BDEI bestimmt, dass materialisierte Informationsintegration ausschließlich ausgehend von der aktuellen Version eines Datensatzes erfolgen kann. Diese Information ist über die 'ProvenanceTable' des Datensatzes verfügbar (siehe Abbildung 8.8).

Data Logistics werden beim Datenaustausch in BDEI über die Web-Services der Repositorien und die Wrapper ausgeführt. Dabei übernehmen die Web-Services den Export und Import der Daten über die XML-Repräsentation von PODSL-Biodiv. Durch Wrapper wird spezielle Software zur Verfügung gestellt, welche von den Knoten angepasst wird. Aufgabe der Wrapper ist es, die Daten aus dem XML-Format in die lokale Struktur eines Repositoriums vorzunehmen und die Daten in diesem abzuspeichern.

8.4.8 Multimediadaten

Multimediadaten sind Bilder, Ton- oder Videoaufzeichnungen und verfügen damit über eine binäre Struktur. In PODSL-Biodiv werden Multimediadaten über Konzepte in einem Begleitdokument beschrieben, die speziell dazu in das Datenmodell aufgenommen wurden. Die Verwendung von Begleitdokumenten wurde bereits in DiversityCollection zur Verwaltung von Multimediadaten verwendet und konnte im Rahmen des IBF-Projektes auch zur Datenübertragung von Multimediadaten vom Mobilgerät an das SNSB eingesetzt werden. Die Multimediadaten in binärer Form an sich können nicht als Teil von PODSL-Biodiv dargestellt werden und werden in Form von Dateien von den Repositorien gespeichert. Die Datenübertragung von Multimediadaten kann sowohl in virtueller als auch in materialisierter Form erfolgen.

Bei einer virtuellen Datenübertragung wird zunächst das Begleitdokument an den

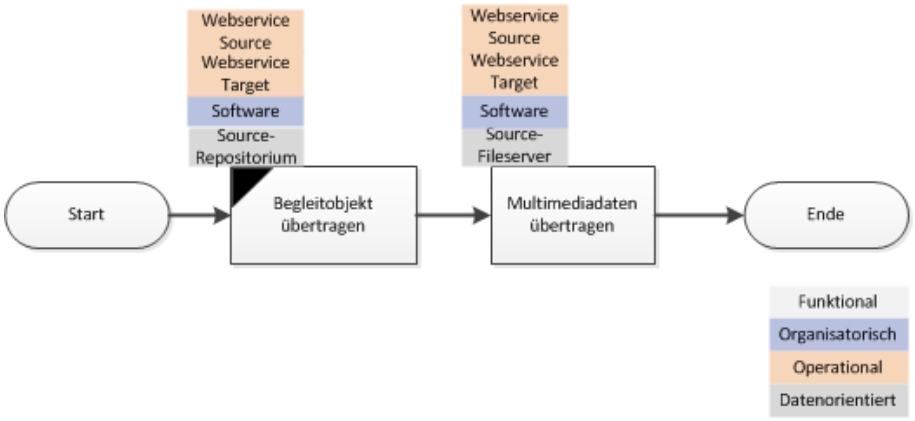

Abbildung 8.11: Prozess der virtuellen Übertragung von Multimediaobjekten in BDEI

aufrufenden Web-Service übermittelt. Dies ist eine normale virtuelle Datenübertragung, wie diese in Abbildung 8.9 beschrieben ist. Zusätzlich müssen allerdings noch die Multimediadaten in binärer Form transferiert werden. Dies ist in Abbildung 8.11 dargestellt. Aus dem Begleitdokument wird der Speicherort der eigentlichen Multimediadaten auf dem Fileserver der Quelle ermittelt. Diese werden anschließend an den aufrufenden Web-Service übertragen und können dort über ein Portal publiziert oder in Softwareprodukten verwendet werden.

Der Prozess für die materialisierte Integration von Multimediadaten in BDEI ist in Abbildung 8.12 modelliert. In diesem Prozess werden sowohl das Begleitdokument als auch das die Multimediadaten in binären Form gespeichert. Dadurch entsteht eine neue Version des Begleitdokuments. In diesem muss der Speicherort des Multimediaobjekts angepasst und das Begleitdokument als neue Version in die ProvenanceTable des Datensatzes aufgenommen werden. Dies stellt die Veränderung eines existierenden Datensatz – wie in Abschnitt 8.4.3 beschrieben – dar.

Der Austausch von Multimediadaten fällt in den Bereich der Data Logistics, da Bilder und Videoeufzeichnungen nicht semantisch integriert werden müssen. Die Anforderungen an die Data Logistics bestehen in der Übertragung der Multimediadaten in binärer Form zwischen dem Quell- und dem Ziel-Fileserver.

8.4.9 Datenpublikation

Die Publikation von Daten in BDEI erfolgt über Datenportale. Dazu betreibt die Zentrale ein Portal, welches die allgemein zugänglichen Daten der Infrastruktur zur Verfügung stellt. Darüber hinaus besteht für die Knoten die Möglichkeit lokale Portale zu betreiben, die es ermöglichen, den Zugang zu Daten auf ausgewählte Personen zu

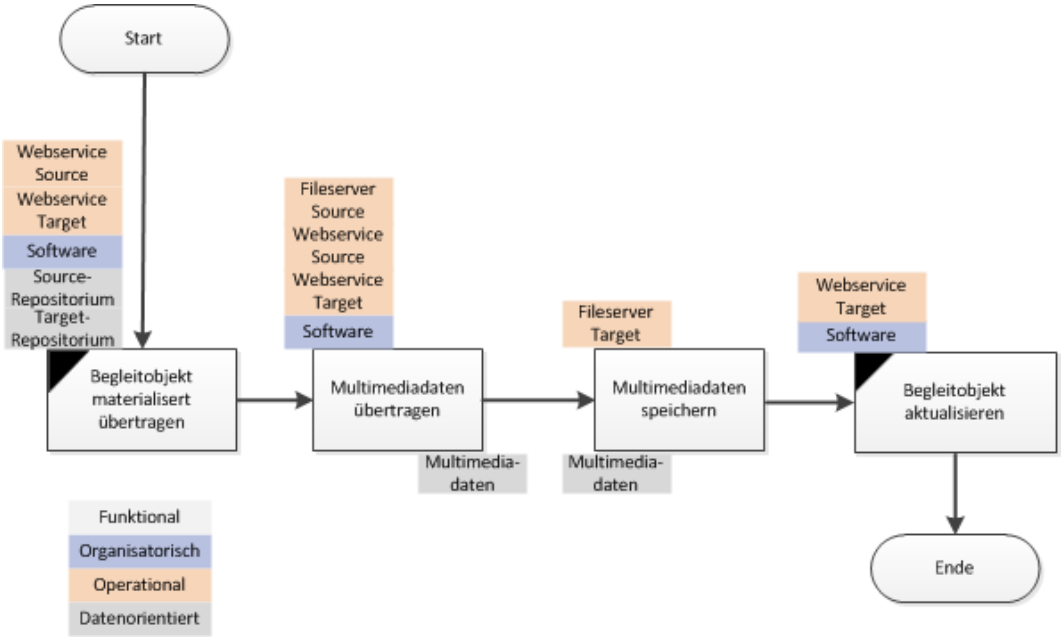

Abbildung 8.12: Prozess der materialisierten Übertragung von Multimediaobjekten in BDEI

beschränken und lokale Erweiterungen von PODSL-Biodiv zu berücksichtigen. Dabei können Datenportale einerseits den materialisierten Bestand eines lokalen Repositoriums oder andererseits Daten virtuell über BDEI anfordern und diese darstellen. Die Software zur Publikation von Daten wird in BDEI über die Zentrale angeboten. In diesem Zusammenhang ist für einen Knoten die Erstellung eines lokalen Portals optional, da die Publikation der Daten bereits über das zentrale Datenportal garantiert ist. Ein Vorteil in der Erstellung eines lokalen Datenportals liegt darin, dass ein Konten die Kontrolle über die publizierten Daten behält, wenn diese seine Daten dem Netzwerk nur in eingeschränkter Form zur Verfügung stellen möchte. Des Weiteren ist ein Knoten mit der Erstellung eines lokalen Datenportals dazu in der Lage, Daten einem ausgewählten Nutzerkreis zugänglich zu machen. Eine wichtige Funktion von Portalen ist es dabei, die Suche von Datensätzen in BDEI zu ermöglichen. Dazu werden über ein Datenportal Suchkriterien spezifiziert, nach denen Datensätze im Netzwerk gesucht werden können. Als Suchkriterium ist dabei jedes Attribut eines Konzeptes erlaubt, welches in PODSL-Biodiv spezifiziert ist. Darüber hinaus kann ein Datenportal die Suche im gesamten Netzwerk ausführen oder sich auf ausgewählte Repositorien beschränken. Für die Durchführung einer Suche werden Web-Services der Knoten aufgerufen. Diese geben dann die Identifier der Datensätze ihres lokalen Repositoriums zurück, die das Suchkriterium erfüllen.

8.5 Fazit

Mit BDEI wird eine Infrastruktur für die Biodiversitätsinformatik vorgestellt, welche sich an der Architektur von GBIF orientiert und Probleme der GBIF-Infrastruktur in der funktionalen Ebene löst. BDEI unterstützt dabei die Autonomie der lokalen Repositorien und bietet die Möglichkeit der projektspezifischen Erweiterung von PODSL-Biodiv. Ein zentrales Element ist die Vermeidung von Datenverlusten bei der materialisierten Informationsintegration. Diese wird durch Prozesse ermöglicht, welche die Erweiterbarkeit von PODSL-Biodiv durch Vererbung ausnützen. Des Weiteren wird in BDEI der Umgang mit Multimediadaten explizit berücksichtigt und die Publikation von Multimediadaten ermöglicht. Ein besonderer Vorteil beim Einsatz von BDEI ist die umfassende Unterstützung von Data Provenance. Über die zentrale Speicherung von Informationen zur Data Provenance wird gewährleistet, dass alle Teilnehmer von BDEI jederzeit über dieselben Informationen zu Herkunft und Veränderungen eines Datensatzes verfügen. Durch die Verwendung von PODSL-Biodiv als Datenaustauschformat ist die Identifikation von Datensätzen und die Vergleichbarkeit der Daten gegeben. Zusammen mit der Auflösung der PODSL-Identifier über die Web-Services der Repositorien und der Zentrale können Datensätzen in BDEI referenziert werden. Damit erfüllt BDEI die Anforderungen an die funktionale Ebene aus Abschnitt 7.3.1. Zusammen mit den operationalen und organisatorischen Komponenten von GBIF ist BDEI damit eine Infrastruktur, welche die Anforderungen der Biodiversitätsinformatik erfüllt.

Im Rahmen des IBF-Projektes konnten bereits erste Versuche zu den Prinzipien von BDEI ausgeführt werden. Hierbei ist zuerst der materialisierte Austausch von Multimediadaten zu nennen. Dieses ist für die Übertragung von Multimediadaten von DiversityMobile an das SNSB umgesetzt und konnte im Projektverlauf erfolgreich erprobt werden. Darüber hinaus werden Begleitdokumente auch zur internen Verwaltung von Multimediaobjekten am SNSB verwendet. Darüber hinaus konnte die Nutzung von Web-Services im Rahmen des IBF-Projektes zum Datenaustausch erfolgreich getestet werden. Innerhalb des IBF-Projektes trat das SNSB-IT-Center als zentrales Repositorium auf. Es konnte somit nur der Einsatz von Web-Services in einer zentralen Infrastruktur getestet werden. Web-Services können aber aauch Daten zwischen zwei gleich berechtigten Repositorien austauschen und sind somit auch zum Datenaustausch in dezentralen Infrastrukturen geeignet.

Die Grundlagen der Referenzierung und des Austausch von Spezifikationen über eine Infrastruktur wurden bereits bei der Entwicklung von OMME [205] getestet. Dabei stellt die Entwicklung von lokalen Erweiterungen von PODSL-Biodiv in BDEI

einer Erweiterung des Spezifikationsaustauschs mit OMME dar, da Erweiterungen in OMME nicht auf einen speziellen Nutzerkreis beschränkt werden können. Die eindeutige Referenzierbarkeit und Auflösung von globalen Referenzen in Infrastrukturen ist in Webtechnologien wie UDDI weit verbreitet und kann somit als etabliert betrachtet werden. Im Kontext der Datenspeicherung werden bereits Identifier in Ontologien und OMME eingesetzt. Darüber hinaus existiert wird mit den 'Life Science Identifiern' (LSID) ein ähnliches Prinzip in den Lebenswissenschaften angewendet. Damit ist die Sicherung der Identität und die Auflösbarkeit von Referenzen sowie die lokale Erweiterung von PODSL-Biodiv technisch realisierbar.

Die Erhaltung von 'Data Provenance' lässt sich auf die zentrale Verwaltung der 'ProvenanceTable' zurückführen. Dabei handelt es sich um spezifische Datensätze, die in BDEI ausschließlich vom zentralen Repositorium verwaltet werden. Auch wenn dieses Prinzip im Rahmen des IBF-Projektes nicht im Bezug auf Data Provenance getestet werden konnte, konnte im IBF-Projekt gezeigt werden, dass die zentrale Verwaltung von Datensätzen über Web-Services technisch realisierbar ist. Damit sind alle Voraussetzungen des Prozesses 'Aktualität sichern' erfüllt und dieser könnte in BDEI in dieser Form implementiert werden. Genauso sind auch alle Voraussetzungen für den virtuellen und den materialisierten Datenaustausch erfüllt. Sowohl virtueller als auch materialisierter Datenaustausch wird in der Praxis erfolgreich angewendet. Dementsprechend handelt es sich in beiden Fällen um ein etabliertes Prinzip. Da die Voraussetzungen dafür in BDEI technisch realisierbar sind, kann der virtuelle und der materialisierte Datenaustausch in BDEI umgesetzt werden. Damit sind alle Prinzipien von BDEI technisch realisierbar. Folglich kann BDEI mit diesen Prinzipien als Infrastruktur zum Datenaustausch in der Biodiversitätsinformatik tatsächlich eingesetzt werden.

Kapitel 9

Zusammenfassung und Ausblick

Die vorliegende Arbeit setzt den Grundstein für die Erstellung von Datenstandards und dem Datenaustausch über Infrastrukturen in der Biodiversitätsinformatik. Dazu wurden die Teilbereiche der Erstellung von Datenstandards und Infrastrukturen separat in generischer Form betrachtet. Dadurch sind die Erkenntnisse dieser Arbeit auch auf andere Domänen anwendbar. Darüber hinaus werden in dieser Arbeit Vorarbeiten des Lehrstuhl aus verschiedenen Bereichen eingesetzt und als Grundlage angewendet und erweitert. Für die Entwicklung von PODSL-Biodiv und BDEI wurde konsequent eine prozessorientierte Sichtweise eingenommen. Damit bildet POPM als Methode für die Modellierung von Prozessen die Basis dieser Arbeit. Eine weitere wichtige Grundlage ist die Metamodellierung nach OMME, über die die Flexibilität von PODSL-Biodiv garantiert wird. Im Bereich der Infrastrukturen konnte mit BDEI ein Konzept einer Infrastruktur für den flexiblen Datenaustausch entwickelt werden.

Es konnten in dieser Arbeit durch folgende eigene Entwicklungen ein Beitrag zu aktuellen Fragestellungen der Biodiversitätsinformatik erbracht werden:

- Evaluation von Datenstandards: Für die Evaluation von Datenstandards konnte ein Kriterienkatalog auf Basis vorhandener Arbeiten zu einem Evaluationsframework vereinigt werden. Kern dieses Frameworks ist die Evaluation der Vollständigkeit eines Datenstandards vor einem Anwendungshintergrund mit POSE. Dabei stellt POSE eine generische Eigenentwicklung auf der Basis von Prozessen dar, welche die Evaluation der Vollständigkeit ermöglicht. Mit Hilfe diese Evaluationssystems wurden etablierte Standards der Biodiversitätsinformatik untersucht. Es wurde gezeigt, dass die etablierten Standards die Anforderungen der Biodiversitätsinformatik insbesondere im Bezug auf Vollständigkeit und Flexibilität nicht erfüllen und somit ein neuer Standard entwickelt werden muss.

- Entwicklung von PODSL-Biodiv: PODSL-Biodiv setzt sich aus einem domänenspezifischen und einen generischen Teil zusammen. Beide Teile wurde mit Hilfe von OMME in eine Metastruktur eingebettet. PODSL ist dabei als ein logisches Datenschema auf M_1-Ebene entworfen, welches über Repräsentationen in verschiedenen Anwendungstechnologien wie Datenbanken, XML-Schemata oder Ontologien verfügt. Ein wesentlicher Aspekt von PODSL ist die Erweiterbarkeit über Vererbung. Dies minimiert Datenverluste, wenn PODSL-Biodiv von einem Datenspeicher nicht vollständig unterstützt wird.

- Evaluation von Infrastrukturen mit IEF und IEF-Biodiv: IEF stellt ein Baukastensystem von Kriterien zur Entwicklung von domänenspezifischen Evaluationsframeworks zur Verfügung. Auf dieser Basis wurde für die Biodiversitätsinformatik EIF-Biodiv erstellt. Bei der Evaluation von Infrastrukturen werden neben der Evaluation der technischen Infrastruktur auch soziale Kriterien berücksichtigt. Der Kern der Evaluation ist aber durch die Evaluation der funktionalen Ebene geprägt. In dieser wird analysiert, ob eine Infrastruktur für eine Domäne über ein vollständiges Austauschmodell verfügt und in dieser Datenverluste auftreten. Ein weiteres Merkmal ist die Unterstützung von Data Provenance. Für die Evaluation wurden wichtige Infrastrukturen aus den Lebenswissenschaften analysiert. Es konnte gezeigt werden, dass das GBIF-Netzwerk als Infrastruktur in der Biodiversitätsinformatik zwar prinzipiell gut geeignet ist, aber über eine Reihe von Mängeln auf der funktionalen Ebene verfügt.

- Entwicklung von BDEI: Mit BDEI wird auf Grundlage der GBIF-Infrastruktur ein eigenes Konzept entwickelt, um die Mängel von GBIF auf funktionaler Ebene zu beseitigen. Die Konzepte zur Verbesserungen wurden dabei in Form von Prozessen präsentiert, welche in den Kontext zu Data Logistics und der semantischen Integration DaltON gesetzt wurden. BDEI verwendet dabei PODSL-Biodiv als Austauschformat und profitiert von der Flexibilität dieser Datenstruktur.

Damit lassen sich die Beiträge dieser Arbeit verschiedenen Arten zuordnen. Die erste Art von Beiträgen ist die Analyse des Kernproblems und der verfügbaren Werkzeuge in einem Themenbereich. Im Bereich der Datenstandards ist dies das Problem der Evaluation von konzeptuellen Schemata mit den verschiedenen Frameworks zur Evaluation konzeptueller Schemata als Werkzeug. Im Bereich der Infrastrukturen ist dies die Analyse der Komponenten einer Infrastruktur mit den zu Grunde liegenden

Technologien als Werkzeug. Die zweite Art von Beiträgen basiert auf den Analysen und hat die Erstellung eines Konzeptes zur Evaluation der Ist-Situation zum Ziel. Deshalb wurde für Datenstandards POSE zur Evaluation der Vollständigkeit und für Infrastrukturen IEF-Biodiv zur Evaluation von Infrastrukturen in der Biodiversitätsinformatik entwickelt. Im nächsten Schritt wird diese Evaluation an etablierten Datenstandards und Infrastrukturen ausgeführt. Durch die Evaluation können die Schwächen der etablierten Standards und Infrastrukturen aufgedeckt werden. Dies stellt die dritte Art von Beiträgen dar. Die vierte Art von Beiträgen liegt letztlich in der Entwicklung einer Lösung für die identifizierten Schwächen. Diese führte für Datenstandards zur Entwicklung von PODSL-Biodiv und für Infrastrukturen zur Entwicklung von BDEI.

Ein Ansatzpunkt für weitere Arbeiten ist die Erweiterung und Zertifizierung von PODSL-Biodiv. Mit PODSL-Biodiv konnte für viele relevante Prozesse der Biodiversitätsinformatik eine Struktur zur Datenspeicherung erstellt werden. Ausgehend vom Schema erweitert PODSL-Biodiv die Mächtigkeit von DwC durch Berücksichtigung durch Einbeziehung von wichtigen Anwendungsfällen. Allerdings besteht in einer komplexen Anwendungsdomäne wie der Biodiversitätsforschung stets der Bedarf einer Anpassung des Datenmodells. Dementsprechend muss PODSL-Biodiv kontinuierlich weiterentwickelt werden. Dazu bietet sich die Gelegenheit in den Folgeprojekten zu IBF, welche sich aktuell in der Planung befinden. Durch den erfolgreichen Einsatz von PODSL-Biodiv in diesen Projekten kann damit die Grundlage einer Zertifizierung von PODSL-Biodiv durch die TDWG als allgemein gültiger Datenstandard für die Biodiversitätsinformatik gelegt werden.

Im Bereich der Infrastrukturen ist die konkrete Implementierung von BDEI in einem Projekt mit einer großen Anzahl von Beteiligten der nächste Schritt. Dabei konnten die Konzepte von BDEI im Rahmen des IBF-Projekts wie z.B. die Multimediaunterstützung bereits teilweise umgesetzt werden. Die Umsetzung des Data Provenance-Konzepts von BDEI in einer großen Infrastruktur ist als zukünftiges Projekt von besonderem Interesse, da das GBIF-Netzwerk Data Provenance nicht unterstützt. Data Provenance muss aber eine zentrale Komponente einer Infrastruktur für den Datenaustausch sein.

Über die Verwendung des flexiblen Datenstandards PODSL-Biodiv ist es möglich, zukünftige Anforderungen der Domäne der Biodiversitätsinformatik zu berücksichtigen und gleichzeitig die Vergleichbarkeit von Daten zu garantieren. Diese Merkmale sind in aktuellen Datenstandards der Biodiversitätsinformatik nicht zu finden. Dies ist aber die Grundlage für die Wiederverwendbarkeit der Daten in einem anderen

Kontext und die Auswertung der Daten mit Data Mining. Die prozessorientierte Strukturierung von PODSL-Biodiv und BDEI ermöglicht die Anwendung von Process Mining zur Auswertung der Daten. Zu diesem Themenbereich konnten mit [213] relevante Beiträge von Arbeitsgruppen des Lehrstuhls publiziert werden. Eine interessante Weiterentwicklung dieser Arbeit wäre somit die Verknüpfung der Erkenntnisse aus dem Bereich des Process Mining mit der Datenerfassung in PODSL-Biodiv, um die Prozesse in der Biodiversitätsinformatik auszuwerten.

Anhang A

Anhang: Anforderungsprofile für POSE

Aspekt	Attribut	Genauigkeit	Datentyp
Funktional	Identifikation	Taxonomische Bezeichnung URL	String String
Organisatorisch	Kartierer	Name URL	String String
Operational	GPS	Abweichung des Ausführungsorts	Float
Temporal	Ausführungszeitpunkt	Jahr, Monat, Tag, Stunden, Minuten, Sekunden	DateTime
Lokal	Ausführungsort	WGS84-Latitude WGS84-Longitude	Float Float
Datenorientiert	Identifier	URL	String

Tabelle A.1: Anforderungsprofil PED1 aus UC1

Aspekt	Attribut	Genauigkeit	Datentyp
Funktional	Identifikation	Taxonomische Bezeichnung URL	String String
Organisatorisch	Sammler	Name URL	String String
Operational	GPS	Abweichung des Ausführungsorts	Float
Temporal	Ausführungszeitpunkt	Jahr, Monat, Tag, Stunden, Minuten, Sekunden	DateTime
Lokal	Ausführungsort	WGS84-Latitude WGS84-Longitude	Float Float
Datenorientiert	Identifier	URL	String
	Referenz PED3	URL	String
	Referenz Beleg	URL	String

Tabelle A.2: Anforderungsprofil PED2 aus UC2

Aspekt	Attribut	Genauigkeit	Datentyp
Funktional	Identifikation	Taxonomische Bezeichnung	String
Organisatorisch	Sammler	Name URL	String String
Operational	---	---	---
Temporal	Ausführungszeitpunkt	Jahr, Monat, Tag	DateTime
Lokal	Ausführungsort	Latitude in degrees Longitude in degrees	String String
		Textuell	String
Datenorientiert	Identifier	Belegnummer	String
	Referenz PED2	URL	String
	Referenz PED3	URL	String

Tabelle A.3: Anforderungsprofil Beleg aus UC2

Aspekt	Attribut	Genauigkeit	Datentyp
Funktional	Identifikation	Taxonomische Bezeichnung URL	String String
Organisatorisch	Kurator	Name URL	String String
	Sammlung	Name URL	String String
Operational	GPS	Abweichung des Ausführungsorts	Float
Temporal	Ausführungszeitpunkt	Jahr, Monat, Tag, Stunden, Minuten, Sekunden	DateTime
Lokal	Ausführungsort	WGS84-Latitude WGS84-Longitude	Float Float
Datenorientiert	Identifier	URL	String
	Referenz PED2	URL	String
	Referenz Beleg	URL	String

Tabelle A.4: Anforderungsprofil PED3 aus UC2

Aspekt	Attribut	Genauigkeit	Datentyp
Funktional	Identifikation	Taxonomische Bezeichnung URL	String String
Organisatorisch	Sammler	Name URL	String String
Operational	GPS	Abweichung des Ausführungsorts	Float
Temporal	Ausführungszeitpunkt	Jahr, Monat, Tag, Stunden, Minuten, Sekunden	DateTime
Lokal	Ausführungsort	WGS84-Latitude WGS84-Longitude	Float Float
Datenorientiert	Identifier	URL	String
	Referenz PED5	URL	String
	Referenz PED6	URL	String

Tabelle A.5: Anforderungsprofil PED4 aus UC3

Aspekt	Attribut	Genauigkeit	Datentyp
Funktional	Identifikation	Taxonomische Bezeichnung URL	String String
Organisatorisch	Sammler	Name URL	String String
Operational	GPS	Abweichung des Ausführungsorts	Float
Temporal	Ausführungszeitpunkt	Jahr, Monat, Tag, Stunden, Minuten, Sekunden	DateTime
Lokal	Ausführungsort	WGS84-Latitude WGS84-Longitude	Float Float
Datenorientiert	Identifier	URL	String
	Referenz PED4	URL	String
	Referenz PED6	URL	String

Tabelle A.6: Anforderungsprofil PED5 aus UC3

Aspekt	Attribut	Genauigkeit	Datentyp
Funktional	Typ des MMO	Text	String
Organisatorisch	Urheber	Name URL	String String
Operational	Aufnahmegerät	Textuell	String
Temporal	Ausführungszeitpunkt	Jahr, Monat, Tag, Stunden, Minuten, Sekunden	DateTime
Lokal	Ausführungsort	WGS84-Latitude WGS84-Longitude	Float Float
Datenorientiert	Identifier	URL	String
	Referenz PED4	URL	String
	Referenz PED5	URL	String

Tabelle A.7: Anforderungsprofil PED6 aus UC3

Aspekt	Attribut	Genauigkeit	Datentyp
Funktional	Identifikation	Taxonomische Bezeichnung URL	String String
Organisatorisch	Kurator	Name	String
	Sammlung	URL	String
Operational	---	---	---
Temporal	Auspflegezeitpunkt	Jahr, Monat, Tag, Stunden, Minuten, Sekunden	DateTime
	Einpflegezeitpunkt	Jahr, Monat, Tag, Stunden, Minuten, Sekunden	DateTime
Lokal	Archivierungsort	Textuell	String
Datenorientiert	Identifier	URL	String
	Referenz DNAAnalyse	URL	String

Tabelle A.8: Anforderungsprofil PED7 aus UC4

Aspekt	Attribut	Genauigkeit	Datentyp
Funktional	DNA-Analyse	Externes Dokument	Extern
Organisatorisch	Institut	Name URL	String String
	Laborant	Name URL	String String
Operational	Analysemethode	Externes Dokument	Extern
Temporal	Ausführungszeitpunkt	Jahr, Monat, Tag, Stunden, Minuten, Sekunden	DateTime
Lokal	Ausführungsort	Textuell	String
Datenorientiert	Identifier	URL	String
	Referenz PED7	URL	String

Tabelle A.9: Anforderungsprofil DNAAnalyse aus UC4

Aspekt	Attribut	Genauigkeit	Datentyp
Funktional	Identifikation	Taxonomische Bezeichnung URL	String String
	Art der Beziehung	Bezeichnung URL	String String
	Höhe	Ergebnis der Messung Einheit	Double String
	Umfang	Ergebnis der Messung Einheit	Double String
Organisatorisch	Ökologe	Name URL	String String
Operational	Messmethode	Textuell	String
Temporal	Ausführungszeitpunkt	Jahr, Monat, Tag, Stunden, Minuten, Sekunden	DateTime
Lokal	Auführungsort	Textuell	String
Datenorientiert	Identifier	URL	String
	Referenz PED9	URL	String

Tabelle A.10: Anforderungsprofil PED8 aus UC5

Aspekt	Attribut	Genauigkeit	Datentyp
Funktional	Identifikation	Taxonomische Bezeichnung URL	String String
	Art der Beziehung	Bezeichnung URL	String String
	Höhe	Ergebnis der Messung Einheit	Double String
	Umfang	Ergebnis der Messung Einheit	Double String
Organisatorisch	Ökologe	Name URL	String String
Operational	Messmethode	Textuell	String
Temporal	Ausführungszeitpunkt	Jahr, Monat, Tag, Stunden, Minuten, Sekunden	DateTime
Lokal	Auführungsort	Textuell	String
Datenorientiert	Identifier	URL	String
	Referenz PED8	URL	String
	Referenz PED10	URL	String

Tabelle A.11: Anforderungsprofil PED9 aus UC5

Aspekt	Attribut	Genauigkeit	Datentyp
Funktional	Identifikation	Taxonomische Bezeichnung URL	String String
	Art der Beziehung	Bezeichnung URL	String String
	Höhe	Ergebnis der Messung Einheit	Double String
	Umfang	Ergebnis der Messung Einheit	Double String
Organisatorisch	Ökologe	Name URL	String String
Operational	Messmethode	Textuell	String
Temporal	Ausführungszeitpunkt	Jahr, Monat, Tag, Stunden, Minuten, Sekunden	DateTime
Lokal	Auführungsort	Textuell	String
Datenorientiert	Identifier	URL	String
	Referenz PED9	URL	String
	Referenz PED11	URL	String

Tabelle A.12: Anforderungsprofil PED10 aus UC5

Aspekt	Attribut	Genauigkeit	Datentyp
Funktional	Identifikation	Taxonomische Bezeichnung URL	String String
	Art der Beziehung	Bezeichnung URL	String String
Organisatorisch	Ökologe	Name URL	String String
Operational	---	---	---
Temporal	Ausführungszeitpunkt	Jahr, Monat, Tag, Stunden, Minuten, Sekunden	DateTime
Lokal	Ausführungsort	Textuell	String
Datenorientiert	Identifier	URL	String
	Referenz PED10	URL	String

Tabelle A.13: Anforderungsprofil PED11 aus UC5

Anhang B

Mapping zwischen PODSL-Biodiv und DwC

In diesem Anhang ist ein Mapping zwischen den Basisklassen von DarwinCore und den korrespondierenden Strukturen von PODSL-Biodiv angegeben. In PODSL werden zur Aufzeichnung von Prozessen 'ProcessExecutionDocuments' verwendet, welche mit Beziehung arbeiten. Dementsprechend ist die korrespondierende Struktur in PODSL-Biodiv meist über Beziehung verfügbar. In diesem Fall wird der Pfad bis zur korrespondierenden Struktur angegeben.

Mapping DwC <-> PODSL-Biodiv		
DwC	**PODSL-Biodiv**	
RecordLevelTerms	**DataSet**	
	Entity	**Attribute/Relation**
dcterms:type (interne Verwaltung von DwC)	Best Match: ObservationObject	From ObservationObject derived concepts
dcterms:modified	BaseEntity	ProvenanceRelation
dcterms:language	BiodivDataSet	language
dcterms:rights	BiodivDataSet	DataSetLicenseRelation
dcterms:rightsHolder	BaseDataSet	OwnershipRelation
dcterms:accessRights	BaseDataSet	AccessRightsRelation
dcterms:bibliographicCitation	BiodivDataSet	Identifier
dcterms:references	BaseDataSet	ReferenceRelation
institutionID	BiodivDataSet	DataSetCollectionRelation -> Collection -> CollectionInstituteRelation -> Institute -> Identifier
collectionID	BiodivDataSet	DataSetCollectionRelation -> Collection -> Identifier
datasetID	BiodivDataSet	Identifier
institutionCode	BiodivDataSet	DataSetCollectionRelation -> Collection -> CollectionInstituteRelation -> Institute -> Identifier
collectionCode	BiodivDataSet	DataSetCollectionRelation -> Collection
datasetName	BiodivDataSet	entityName
ownerInstitutionCode	BaseDataSet	OwnershipRelation
basisOfRecord	Best Match: ObservationObject (interne Verwaltung von DwC)	From ObservationObject derived concepts
informationWithheld	BiodivDataSet	informationWithHeld
dataGeneralizations	BaseEntity	ProvenanceRelation
dynamicProperties (can contain any Information in CSV-Format or Key-Value-Pairs)	BiodivDataSet	ExternalDocumentRelation

Tabelle B.1: Mapping zwischen den RecordLevelTerms von DwC und PODSL-Biodiv

Mapping DwC <-> PODSL-Biodiv		
DwC	**PODSL-Biodiv**	
Taxon	**Entity**	**Attribute or Relation**
taxonID	BaseTaxon	Identifier
scientificNameID	BaseTaxon	scientificNameID
acceptedNameUsageID	TaxonLink	AcceptedNameTaxonRelation -> BaseTaxon -> Identifier
parentNameUsageID	TaxonLink	ParentTaxonRelation -> BaseTaxon -> Identifier
originalNameUsageID	TaxonLink	OriginalNameTaxonRelation -> BaseTaxon -> Identifier
nameAccordingToID	TaxonPublication	WorkOfReferenceRelation -> WorkOfReference -> Identifier
namePublishedInID	TaxonPublication	WorkOfReferenceRelation -> WorkOfReference -> Identifier
taxonConceptID	Identifier	BaseTaxonIdentifier
scientificName	BaseTaxon	scientificName
acceptedNameUsage	TaxonLink	AcceptedNameTaxonRelation -> BaseTaxon -> scientificName
parentNameUsage	TaxonLink	ParentNameTaxonRelation -> BaseTaxon -> scientificName
originalNameUsage	TaxonLink	OriginalNameTaxonRelation -> BaseTaxon -> scientificName
nameAccordingTo	TaxonPublication	WorkOfReferenceRelation -> WorkOfReference
namePublishedIn	TaxonPublication	WorkOfReferenceRelation -> WorkOfReference
namePublishedInYear	TaxonPublication	WorkOfReferenceRelation -> WorkOfReference -> yearOfPublication
higherClassification	TaxonRanks	csv of all TaxonRank-Attributes
kingdom	TaxonRanks	kingdom
phylum	TaxonRanks	phylum
class	TaxonRanks	class
order	TaxonRanks	order
family	TaxonRanks	family
genus	TaxonRanks	genus
subgenus	TaxonRanks	subgenus
specificEpithet	TaxonRanks	specificEpithet
taxonRank	TaxonRanks	taxonRank
verbatimTaxonRank	TaxonRanks	verbatimTaxonRank
scientificNameAuthorship	TaxonRanks	scientificNameAuthorship
vernacularName	Taxon	vernacularName
nomenclaturalCode	BaseTaxon	nomenclaturalCode
taxonomicStatus	Taxon	taxonomicStatus
nomenclaturalStatus	TaxonLink	nomenclaturalStatus
taxonRemarks	BaseTaxon	taxonRemarks

Tabelle B.2: Mapping zwischen der Basisklasse Taxon von DwC und PODSL-Biodiv

Mapping DwC <-> PODSL-Biodiv		
DwC	**PODSL-Biodiv**	
Location	**Entity**	**Attribute**
locationID	LocationObject	Identifier
higherGeographyID	ParentGeography-Relation	ParentGeographyRelation -> LocalityDescription -> Identifier
higherGeography	ParentGeography-Relation	ParentGeographyRelation -> LocalityDescription -> description
continent	LocalityDescription	continent
waterBody	MarineLocality	waterbody
islandGroup	Islandlocality	islandGroup
island	IslandLocality	island
country	PoliticalLocality	country
countryCode	PoliticalLocality	countryCode
stateProvince	PoliticalLocality	stateProvince
county	PoliticalLocality	county
municipality	PoliticalLocality	municipality
locationRemarks	LocationObject	remarks

Tabelle B.3: Mapping zwischen den Konzepten Basisklasse Location von DwC und PODSL-Biodiv mit Entitätscharakter

Mapping DwC <-> PODSL-Biodiv		
DwC	**PODSL-Biodiv**	
Location	**ProcessExecutionDocuments derived from BaseMeasuring**	
	TypeOfElement	**PathToElement**
	All	
locationAccordingTo	operational Aspect	WorkOFReferenceRelation -> WorkOfReference
	Describing	
locality	functional Aspect	ObjectDescriptionRelation -> Description -> description
verbatimLocality	functional Aspect	ObjectDescriptionRelation -> (specialised) ExtendedDescription -> verbatimDescription
	SurfaceMeasuring	
verbatimElevation	functional Aspect	LocalitySurfaceDescriptionRelation -> (specialised, modality: typeOfSurface) VerbatimSurfaceDescription -> description
minimumElevationIn-Meters	functional Aspect	LocalitySurfaceDescriptionRelation -> (specialised, modality: typeOfSurface) UnitSurfaceDescription -> minValue
maximumElevationIn-Meters	functional Aspect	LocalitySurfaceDescriptionRelation -> (specialised, modality: typeOfSurface) UnitSurfaceDescription -> maxValue
verbatimDepth	functional Aspect	LocalitySurfaceDescriptionRelation -> (specialised, modality: typeOfSurface) VerbatimSurfaceDescription -> description
minimumDepthInMeters	functional Aspect	LocalitySurfaceDescriptionRelation -> (specialised, modality: typeOfSurface) UnitSurfaceDescription -> minValue
maximumDepthInMeters	functional Aspect	LocalitySurfaceDescriptionRelation -> (specialised, modality: typeOfSurface) UnitSurfaceDescription -> maxValue
minimumDistanceAbove-SurfaceInMeters	functional Aspect	LocalitySurfaceDescriptionRelation -> (specialised, modality: typeOfSurface) UnitSurfaceDescription -> minValue
maximumDistanceAbove-SurfaceInMeters	functional Aspect	LocalitySurfaceDescriptionRelation -> (specialised, modality: typeOfSurface) UnitSurfaceDescription -> maxValue

Tabelle B.4: Mapping zwischen den Konzepten Basisklasse Location von DwC und PODSL-Biodiv mit Prozesscharakter (Seite 1)

Mapping DwC <-> PODSL-Biodiv		
DwC	**PODSL-Biodiv**	
Location	**ProcessExecutionDocuments derived from BaseMeasuring**	
	TypeOfElement	**PathToElement**
	GeoReferencing	
georeferencedBy	organizational Aspect	ResponsibleScientistRelation -> Scientist
georeferencedDate	temporal Aspect	ExecutionTimeRelation -> ExecutionTime -> DateTime -> Date
georeferenceProtocol	operational Aspect	MethodRelation -> BaseMethod
georeferenceSources	operational Aspect	WorkOFReferenceRelation -> WorkOfReference
georeferenceVeri-ficationStatus	functional Aspect	CoordinateDeterminationRelation -> (modality) VerificationOfDetermination
georeferenceRemarks	Attribute	remarks (inherited)
verbatimCoordinates	functional Aspect	CoordinateDeterminationRelation -> (specialised) VerbatimCoordinates
verbatimLatitude	functional Aspect	CoordinateDeterminationRelation -> (specialised) VerbatimCoordinates -> verbatimLatitude
verbatimLongitude	functional Aspect	CoordinateDeterminationRelation -> (specialised) VerbatimCoordinates -> verbatimLongtitude
verbatimCoordinate-System	functional Aspect	CoordinateDeterminationRelation -> (specialised) VerbatimCoordinates -> coordinateSystem
verbatimSRS	functional Aspect	CoordinateDeterminationRelation -> (specialised) VerbatimCoordinates -> spatialReferenceSystem
decimalLatitude	functional Aspect	CoordinateDeterminationRelation -> (specialised) DecimalCoordinates -> latitude
decimalLongitude	functional Aspect	CoordinateDeterminationRelation -> (specialised) DecimalCoordinates -> longitude
geodeticDatum	functional Aspect	CoordinateDeterminationRelation -> (specialised) DecimalCoordinates -> spatialReferenceSystem
coordinateUncertaintyIn Meters	functional Aspect	CoordinateDeterminationRelation -> (specialised) DecimalCoordinatesWithPrecision -> uncertainty

Tabelle B.5: Mapping zwischen den Konzepten Basisklasse Location von DwC und PODSL-Biodiv mit Prozesscharakter (Seite 2)

Mapping DwC <-> PODSL-Biodiv		
DwC	**PODSL-Biodiv**	
Location	**ProcessExecutionDocuments derived from BaseMeasuring**	
	TypeOfElement	**PathToElement**
	GeoReferencing	
coordinatePrecision	functional Aspect	CoordinateDeterminationRelation -> (specialised) DecimalCoordinatesWithPrecision -> coordinatePrecision
pointRadiusSpatialFit	functional Aspect	CoordinateDeterminationRelation -> (specialised) DecimalCoordinatesWithPrecision -> pointRadiusSpatialFit
footprintWKT	functional Aspect	CoordinateDeterminationRelation -> (specialised) WKTArea -> wKT
footprintSRS	functional Aspect	CoordinateDeterminationRelation -> (specialised) WKTArea -> spatialReferenceSystem
footprintSpatialFit	functional Aspect	CoordinateDeterminationRelation -> (specialised) WKTArea -> pointRadiusSpatialFit

Tabelle B.6: Mapping zwischen den Konzepten Basisklasse Location von DwC und PODSL-Biodiv mit Prozesscharakter (Seite 3)

Mapping DwC<->PODSL-Biodiv		
DwC	**PODSL-Biodiv**	
Identification	**ProcessExecutionDocument: Identification**	
	TypeOfElement	**PathToElement**
identificationID	Identifier	IdentificationIdentifier
identifiedBy	organizational Aspect	ResponsibleScientistRelation
dateIdentified	temporal Aspect	ExecutionTimeRelation
identificationReferences	operational Aspect	WorkOfReferenceRelation
identificationVerificationStatus	functional Aspect	TaxonDeterminationRelation -> Modality -> VerificationOfDetermination
identificationRemarks	Attribute	remarks (inherited)
identificationQualifier	functional Aspect	TaxonDeterminationRelation -> Modality -> SecurityOfDetermination
typeStatus	Object	BiologicalObjectRelation -> BiologicalObject -> (specialisation) ExtendedBiologicalObject -> typeStatus

Tabelle B.7: Mapping zwischen der Basisklasse Identification von DwC und PODSL-Biodiv

Mapping DwC<->PODSL-Biodiv		
DwC	**PODSL-Biodiv**	
Occurrence	**ProcessExecutionDocuments derived from BaseMonitoring and BaseMeasuring**	
	TypeOfElement	**PathToElement**
	All	
occurrenceID	Identifier	Identifier of appropriate ProcessExecutionDocument
catalogNumber	Identifier	Identifier of appropriate ProcessExecutionDocument
occurrenceRemarks	Attribute	remarks (inherited)
recordedBy	organizational Aspect	ResponsibleScientist
individualID	Object	ObservationObject -> Identifier
otherCatalogNumbers	dataoriented Aspect inherited from BaseProcess-ExecutionDocument	ProvenanceRelation
associatedReferences	operational Aspect	WorkOfReferenceRelation
	MultimediaRecording	
associatedMedia	dataoriented Aspect	MultimediaDocumentRelation -> MultimediaDocument
	SpecimenGathering	
recordNumber	dataoriented Aspect	SpecimenRecordRelation -> SpecimenRecord -> fieldNumber
	NumericMeasuring	
individualCount	functional Aspect	NumericMeasurementValueRelation -> NumericMeasurement -> NumericMeasurementValue

Tabelle B.8: Mapping zwischen der Basisklasse Occurence von DwC und PODSL-Biodiv (Seite 1)

Mapping DwC<->PODSL-Biodiv		
DwC	**PODSL-Biodiv**	
Occurrence	**ProcessExecutionDocuments derived from BaseMonitoring and BaseMeasuring**	
	TypeOfElement	**PathToElement**
	Tagging	
sex	functional Aspect	TagAssignmentRelation -> Tag -> tagValue -> (specialisation) sex
lifeStage	functional Aspect	TagAssignmentRelation -> Tag -> tagValue -> (specialisation) lifeStage
reproductiveCondition	functional Aspect	TagAssignmentRelation -> Tag -> tagValue -> (specialisation) reproductiveCondition
behavior	functional Aspect	TagAssignmentRelation -> Tag -> tagValue -> (specialisation) behavoiur
occurrenceStatus	functional Aspect	TagAssignmentRelation -> Tag -> tagValue -> (specialisation) occurrenceStatus
establishmentMeans	functional Aspect	TagAssignmentRelation -> Tag -> tagValue -> (specialisation) establishmentMeans
	SpecimenPreparating	
preparations	operational Aspect	MethodRelation -> BaseMethod
	SpecimenArchiving, AddSpecimen, RemoveSpecimen, DNAAnalysing	
disposition	dataoriented Aspect	SpecimenRecordRelation -> SpecimenRecord -> disposition
	DNAAnalysing	
associatedSequences	dataoriented Aspect	DNAAnalysisRelation -> DNAAnalysis
	Identification	
previousIdentifications	dataoriented Aspect	ProvenanceRelation (inherited)
	CoexistenceObservation	
associatedOccurrences	functional Aspect	FormOfCoexistenceRelation -> BiologicalObject
associatedTaxa	functional Aspect	FormOfCoexistenceRelation -> target=BiologicalObject -> Identification -> TaxonDetermination

Tabelle B.9: Mapping zwischen der Basisklasse Occurence von DwC und PODSL-Biodiv (Seite 2)

Mapping DwC<->PODSL-Biodiv		
DwC	**PODSL-Biodiv**	
Event	**ProcessExecutionDocument: SiteInspection**	
	TypeOfElement	**PathToElement**
eventID	Identifier	SiteInspectionIdentifier
samplingProtocol	operational Aspect	SamplingMethodRelation -> SamplingMethod ->
samplingEffort	operational Aspect	SamplingMethodRelation -> SamplingMethod -> samplingEffort
eventDate	temporal Aspect	ExecutionTimeRelation -> DateTime -> Date
eventTime	temporal Aspect	ExecutionTimeRelation -> DateTime -> Time
startDayOfYear	temporal Aspect	TimeSpanRelation -> source=DateTime -> Date (transformation necessary)
endDayOfYear	temporal Aspect	TimeSpanRelation -> source=DateTime -> Date (transformation necessary)
year	temporal Aspect	ExecutionTimeRelation -> DateTime -> Date -> year
month	temporal Aspect	ExecutionTimeRelation -> DateTime -> Date -> month
day	temporal Aspect	ExecutionTimeRelation -> DateTime -> Date -> day
verbatimEventDate	temporal Aspect	ExecutionTimeRelation -> DateTime -> Date (transformation necessary)
habitat	local Aspect	ExecutionHabitatRelation -> target=Description -> description
fieldNumber	dataoriented Aspect	WrittenDocumentationRelation -> WrittenDocumention -> fieldNumber
fieldNotes	dataoriented Aspect	WrittenDocumentationRelation -> WrittenDocumention -> remarks (inherited)
eventRemarks	Attribute	remarks (inherited)

Tabelle B.10: Mapping zwischen der Basisklasse Event von DwC und PODSL-Biodiv

Mapping DwC<->PODSL-Biodiv		
DwC	**PODSL-Biodiv**	
MeasurementOrFact	**ProcessExecutionDocument: BaseMeasuring and derived Concepts**	
	TypeOfElement	**PathToElement**
measurementID	Identifier	Identifier of appropriate ProcessExecutionDocument
measurementType	functional Aspect	MesurementValueRelation -> BaseMeasurement ->
measurementValue	functional Aspect	MesurementValueRelation -> BaseMeasurement
measurementAccuracy	functional Aspect	MesurementValueRelation -> BaseMeasurement -> (specialisation) NumericMeasurement -> measurementAccuracy
measurementUnit	functional Aspect	MesurementValueRelation -> BaseMeasurement -> (specialisation) -> NumericMeasurement -> measurementUnit
measurementDeterminedDate	temporal Aspect	ExecutionTimeRelation -> DateTime -> Date
measurementDeterminedBy	organizational Aspect	ResponsibleScientisRelation -> Scientist
measurementMethod	operational Aspect	MeasurementMethodRelation -> MeasurementMethod
measurementRemarks	functional Aspect	MesurementValueRelation -> BaseMeasurement -> remarks

Tabelle B.11: Mapping zwischen der Basisklasse MeasurementOrFact von DwC und PODSL-Biodiv

Mapping DwC<->PODSL-Biodiv		
DwC	**PODSL-Biodiv**	
ResourceRelationship	**BaseRelation**	
	TypeOfElement	**PathToElement**
ResourceRelationShipID	`Identifier`	`BaseRelationIdentifier`
resourceID	`source`	`BaseEntity -> Identifier`
relatedResourceID	`target`	`BaseEntity -> Identifier`
relationshipOfResource	`derived Relations`	`Modality`
relationship AccordingTo	`usage of Relation in Process`	`organizational Aspect -> ResponsibleRelation ->`
relationshipEstablishedDate	`usage of Relation in Process`	`dataOriented Aspect -> ExecutionTimeRelation -> DateTime -> Date`
relationshipRemarks	`usage of Relation in Process`	`BaseProcessExecutionDocument -> remarks (inherited)`

Tabelle B.12: Mapping zwischen der Basisklasse ResourceRelationship von DwC und PODSL-Biodiv

Anhang C

Metamodell von PODSL

In diesem Anhang findet sich die konkrete Implementierung von PODSL auf verschiedenen Metaebenen.

C.1 M_2-Ebene von PODSL

```
1  model PODSLM2 {
2      uri "model:/www.ai4.uni-bayreuth.de/PODSLM2";
3      /*
           ***************************************************************************
4
5      PODSL Model Version 2.0 - February 2013
6
7      created by Tobias Schneider (tobias.schneider @ uni-bayreuth.de)
8
9    ***************************************************************************
         */
10
11     level M2 {
12       package PODSLM2{
13
14       //Schema
15       concept Schema {
16           1..1 string  name;
17           0..* concept EntityType entities;
18           0..* concept ProcessExecutionDocument executions;
19         }
20
21
22       //Identifier
```

```
23      concept Identifier{
24          1..1 string name deferredBy 2;
25          1..1 string address deferredBy 2;
26        }
27
28
29      concept ConceptIdentifier extends Identifier{
30          1..1 string repository;
31        }
32
33
34      //Key Concepts
35      concept Aspect
36        {
37          1..1 concept Identifier identifier;
38          1..1 string name;
39        }
40
41      concept Modality{
42        1..1 concept Identifier identifier;
43        1..1 string name;
44        1..1 concept Attribute attribute;
45      }
46
47
48      //ProcessExecution
49
50
51
52      concept ProcessExecutionDocument extends EntityType
53      {
54        1..1 concept Relation object;
55        0..* concept Relation aspect;
56        0..* concept ProcessExecutionSequence sequences;
57      }
58
59
60      //ER Concepts
61
62      abstract concept Attribute{
63        1..1 concept Identifier identifier;
64        1..1 string type;
65        boolean isNullable=false;
66      }
67
```

```
68
69        concept EntityType
70        {
71          1..1 string name;
72          1..1 concept Identifier identifier deferredBy 2;
73          0..* concept Attribute attribute;
74          0..* concept Relation relation;
75        }
76
77        concept Relation
78        {
79          0..1 string role;
80          0..1 concept Aspect aspect;
81          1..1 concept Identifier identifier;
82          1..1 concept EntityType source;
83          1..1 concept EntityType target;
84          1..1 enum Cardinality cardinality = Cardinality.one;
85          0..* concept Modality modality;//Attribute
86        }
87
88        //Technical Concepts
89        enum Cardinality {
90            one,
91            zeroOrOne,
92            oneOrMore,
93            zeroOrMore
94          }
95
96        concept ProcessExecutionSequence extends EntityType{
97          1..* concept ProcessExecutionDocument executions;
98          1..1 integer minExecutions = 0;
99          1..1 integer maxExecutions = 3000;
100       }
101
102     }
103
104   }
105
106 }
```

Listing C.1: M_2-Ebene von PODSL

C.2 Generische M_1-Ebene von PODSL

```
1  model PODSLM1Core {
2    uri "model:/www.ai4.uni-bayreuth.de/PODSLM1Core";
3    include "model:/www.ai4.uni-bayreuth.de/PODSLM2";
4
5    level M1 instanceOf PODSLM2.M2
6    {
7
8      package PODSLM1Core{
9        import PODSLM2.M2.PODSLM2.*;
10
11       /*
             ****************************************************************

12         Definition of Aspects
13
14         An aspect is a concept that carries a name and that has
15         an uri for referencing. Purpose of Aspects is to build a
               semantical
16         structure for documents and collections
17       ****************************************************************
             */
18       abstract Aspect BaseAspect
19       {
20         identifier=BaseAspectIdentifier;
21         name="Base Aspect";
22       }
23
24       concept FunctionalAspect extends BaseAspect
25       {
26         identifier=FunctionalAspectIdentifier;
27         name="Functional Aspect";
28       }
29
30       concept OrganizationalAspect extends BaseAspect
31       {
32         identifier=OrganizationalAspectIdentifier;
33         name="Organizational Aspect";
34       }
35
36       concept BehaviouralAspect extends BaseAspect
37       {
38         identifier=BehaviouralAspectIdentifier;
39         name="Behavioural Aspect";
40       }
```

```
41
42      concept OperationalAspect extends BaseAspect
43      {
44        identifier=OperationalAspectIdentifier;
45        name="Operational Aspect";
46      }
47
48      concept LocalAspect extends BaseAspect
49      {
50        identifier=LocalAspectIdentifier;
51        name="Local Aspect";
52      }
53
54      concept TemporalAspect extends BaseAspect
55      {
56        identifier=TemporalAspectIdentifier;
57        name="Temporal Aspect";
58      }
59
60      concept DataOrientedAspect extends BaseAspect
61      {
62        identifier=DataOrientedAspectIdentifier;
63        name="Data Oriented Aspect";
64      }
65
66      /*
          ****************************************************************

67        Definition of Modalities
68
69        A modality is a concept that carries a name and that has
70        an uri for referencing. Purpose of modalities is to assign
              qualities to relations.
71        To assign values to these qualities an attribute is used which
              can be a controlled
72        vocabulary (CV).
73      ****************************************************************
           */
74
75      abstract Modality BaseModality{
76        identifier=BaseModalityIdentifier;
77        name="BaseModality";
78      }
79
80      Modality PartWholeModality extends BaseModality{
```

```
81          identifier=PartWholeModalityIdentifier;
82          name="PartWholeModality";
83          attribute=remarks;
84        }
85
86        Modality BaseConversionModality{
87          identifier=BaseConversionModalityIdentifier;
88          name="BaseConversionModality";
89          attribute=remarks;
90        }
91
92        /*
              ****************************************************************

93          Definition of Relations
94
95          A relation is a concept that carries a name(role) and that has
96          an uri for referencing. Furthermore is has an source and a
                target
97          concept and can it can be assigned an cardinality.
98          To describe the relation it can be assigned to an aspect
99          and a modality.
100       ****************************************************************
              */
101
102       //Base
103       abstract Relation BaseRelation{
104         aspect=BaseAspect;
105         identifier=BaseRelationIdentifier;
106         source=BaseEntity;
107         target=BaseEntity;
108       }
109
110       abstract concept BaseOneRelation extends BaseRelation{
111         identifier=BaseOneRelationIdentifier;
112         role="1:1 Relation";
113         cardinality=enum one;
114       }
115
116       abstract concept BaseZeroOrOneRelation extends BaseRelation{
117         identifier=BaseZeroOrMoreRelationIdentifier;
118         role="optional 1:1 Relation";
119         cardinality=enum zeroOrOne;
120       }
121
```

```
122     abstract concept BaseOneOrMoreRelation extends BaseRelation{
123       identifier=BaseOneOrMoreRelationIdentifier;
124       role="1:n Relation, n>=1";
125       cardinality=enum oneOrMore;
126     }
127
128     abstract concept BaseZeroOrMoreRelation extends BaseRelation{
129       identifier=BaseZeroOrMoreRelationIdentifier;
130       role="1:n Relation, n>=0";
131       cardinality=enum zeroOrMore;
132     }
133
134     //Behavioural
135
136     concept SubProcessRelation extends BaseRelation{
137       role="is subprocess Of";
138       aspect=BehaviouralAspect;
139       identifier=SubProcessRelationIdentifier;
140       source=BaseProcessExecutionDocument;
141       target=BaseProcessExecutionDocument;
142     }
143
144     //Functional
145
146
147
148     concept ObjectRelation extends BaseOneRelation{
149       role="main object in a process";
150       identifier=ObjectRelationIdentifier;
151       aspect=FunctionalAspect;
152       source=BaseProcessExecutionDocument;
153       target=BaseEntity;
154     }
155
156
157     //Organizational
158
159     concept ResponsibleRelation extends BaseOneOrMoreRelation{
160       role="Responsible";
161       aspect=OrganizationalAspect;
162       identifier=ResponsibleRelationIdentifier;
163       source=BaseProcessExecutionDocument;
164       target=Person;
165     }
166
```

```
167       concept ProducerRelation extends BaseOneRelation{
168         role="Producer";
169         aspect=OrganizationalAspect;
170         identifier=ProducerRelationIdentifier;
171         source=Device;
172         target=OrganizationalBaseEntity;
173       }
174
175       concept OwnershipRelation extends BaseOneRelation{
176         role="Ownership";
177         aspect=OrganizationalAspect;
178         identifier=OwnershipRelationIdentifier;
179         source=BaseDataSet;
180         target=OrganizationalBaseEntity;
181       }
182
183
184       //Temporal
185
186       concept ExecutionTimeRelation extends BaseOneRelation{
187         role="ExecutionTime";
188         aspect=TemporalAspect;
189         identifier=ExecutionTimeRelationIdentifier;
190         target=DateTime;
191       }
192
193       concept TimeSpanRelation extends BaseOneRelation{
194         role="Timespan from start (source) to end (target)";
195         aspect=TemporalAspect;
196         identifier=TimeSpanRelationIdentifier;
197         source=DateTime;
198         target=DateTime;
199       }
200
201       concept DateOfBirthRelation extends BaseOneRelation{
202         role="DateOfBirth";
203         aspect=TemporalAspect;
204         identifier=DateOfBirthRelationIdentifier;
205         source=Person;
206         target=Date;
207       }
208
209       //Local
210
211       concept ExecutionLocalityRelation extends BaseOneOrMoreRelation{
```

```
212         role="ExecutionPlace";
213         aspect=LocalAspect;
214         identifier=ExecutionLocalityRelationIdentifier;
215         target=LocalBaseEntity;
216       }
217
218       concept StoragePlaceRelation extends BaseOneRelation{
219         role="StoragePlace";
220         aspect=LocalAspect;
221         identifier=StoragePlaceRelationIdentifier;
222         target=StoragePlace;
223       }
224
225       concept PhysicalStoragePlaceRelation extends StoragePlaceRelation
              {
226         role="PhysicalStoragePlace";
227         identifier=PhysicalStoragePlaceRelationIdentifier;
228         target=PhysicalStoragePlace;
229       }
230
231       concept DigitalStoragePlaceRelation extends StoragePlaceRelation{
232         role="DigitalStoragePlaceRelation";
233         identifier=DigitalStoragePlaceRelationIdentifier;
234         target=DigitalStoragePlace;
235       }
236
237       //Data-Oriented
238
239       concept DataSetsRelation extends BaseZeroOrMoreRelation{
240         role="Contained Data Sets";
241         aspect=DataOrientedAspect;
242         identifier=DataSetsRelationIdentifier;
243         source=BaseDataSet;
244         target=BaseDataSet;
245       }
246
247
248       concept ProcessExecutionDocumentsRelation extends
              BaseZeroOrMoreRelation{
249         role="Contained Process Executions";
250         aspect=DataOrientedAspect;
251         identifier=ProcessExecutionDocumentsRelationIdentifier;
252         source=BaseDataSet;
253         target=BaseProcessExecutionDocument;
254       }
```

```
255
256       concept ReferenceRelation extends BaseZeroOrMoreRelation{
257         role="associated Relation";
258         identifier=ReferenceRelationIdentifier;
259         source=BaseEntity;
260         target=ExternalReference;
261       }
262
263       concept ExternalDocumentRelation extends ReferenceRelation{
264         role="associated ExternalDocument";
265         identifier=ExternalDocumentRelationIdentifier;
266         source=BaseEntity;
267         target=ExternalDocument;
268       }
269
270
271
272       //Provenance
273
274       concept ProvenanceRelation extends BaseOneRelation{
275         role="ProvenenceTracking";
276         aspect=DataOrientedAspect;
277         identifier=ProvenanceRelationIdentifier;
278         source=BaseEntity;
279         target=ProvenanceTable;
280
281       }
282
283       concept OriginalRelation extends BaseOneRelation{
284         role="Link From ProvenaceTable to original version of entity";
285         aspect=DataOrientedAspect;
286         identifier=OriginalRelationIdentifier;
287         source=ProvenanceTable;
288         target=BaseEntity;
289
290       }
291
292       concept SiblingsRelation extends BaseZeroOrMoreRelation{
293         role="Link From ProvenaceTable of an entity to all derived
              versions of this entity";
294         aspect=DataOrientedAspect;
295         identifier=SiblingsRelationIdentifier;
296         source=ProvenanceTable;
297         target=BaseEntity;
298       }
```

```
299
300       concept ConversionRelation extends BaseZeroOrMoreRelation{
301         role="Link to the documentation of donversions of the original
              entity";
302         aspect=DataOrientedAspect;
303         identifier= ConversionRelationIdentifier;
304         source=ProvenanceTable;
305         target=ConversionProcessDocument;
306       }
307
308       concept ConversionOutputRelation extends BaseOneRelation{
309         role="Conversion from source to target";
310         aspect=FunctionalAspect;
311         identifier=ConversionOutputRelationIdentifier;
312         source=BaseEntity;//Original
313         target=BaseEntity;//Dervied Entity
314         modality=BaseConversionModality;
315       }
316
317       concept ConversionMethodRelation extends BaseOneOrMoreRelation{
318         role="Conversionmethod used in calculation";
319         aspect=OperationalAspect;
320         identifier=ConversionMethodRelationIdentifier;
321         source=ConversionProcessDocument;
322         target=BaseConversionMethod;//Dervied Entity
323       }
324
325       concept OriginalUnitRelation extends BaseOneRelation{
326         role="Original Unit in a UnitConversion";
327         aspect=OperationalAspect;
328         identifier=OriginalUnitRelationIdentifier;
329         source=UnitConversionMethod;
330         target=Unit;
331       }
332
333       concept TargetUnitRelation extends BaseOneRelation{
334         role="Original Unit in a UnitConversion";
335         aspect=OperationalAspect;
336         identifier=TargetUnitRelationIdentifier;
337         source=UnitConversionMethod;
338         target=Unit;
339       }
340
341       concept CreatorRelation extends BaseOneRelation{
342         role="Creator of the digital record";
```

```
343         aspect=OrganizationalAspect;
344         identifier=CreatorRelationIdentifier;
345         source=BaseEntity;
346         target=OrganizationalBaseEntity;
347       }
348
349       concept CreationTimeRelation extends BaseOneRelation{
350         role="Creation time of the digital record";
351         aspect=TemporalAspect;
352         identifier=CreationTimeRelationIdentifier;
353         source=BaseEntity;
354         target=DateTime;
355       }
356
357       //General
358
359       concept PartWholeRelation extends BaseZeroOrMoreRelation{
360         role="PartWholeRelation";
361         identifier=PartWholeRelationIdentifier;
362         source=BaseEntity;
363         target=BaseEntity;
364         modality=PartWholeModality;
365       }
366
367       /*Definition of ProcessExecutionDocuments
368        *
369        */
370
371       ProcessExecutionDocument BaseProcessExecutionDocument{
372         name="BaseProcessExecutionDocument";
373         identifier=BaseProcessExecutionDocumentIdentifier;
374         relation=ProvenanceRelation, CreatorRelation,
              CreationTimeRelation, ReferenceRelation;
375         attribute=remarks;
376       }
377
378
379       concept ConversionProcessDocument extends
            BaseProcessExecutionDocument{
380         identifier=ConversionProcessDocumentIdentifier;
381         object=BaseEntity;//Origin of Conversion
382         relation=ResponsibleRelation,ExecutionTimeRelation,
              ConversionOutputRelation,ConversionMethodRelation;
383       }
384
```

```
385       /*Definition of EntityTypes
386        * EntityTypes are used to descripe Aspect and are compatible
               with ER
387        * Includen in Reference-Inheritence System
388        */
389
390       abstract EntityType BaseEntity{
391         identifier=BaseEntityIdentifier;
392         name="BaseEntity";
393         relation=ProvenanceRelation, CreatorRelation,
               CreationTimeRelation;
394       }
395
396       //DataSet
397
398       concept BaseDataSet extends BaseEntity{
399         identifier=BaseDataSetIdentifier;
400         name="BaseDataSet";
401         relation=OwnershipRelation,ProcessExecutionDocumentsRelation,
               DataSetsRelation,ReferenceRelation;
402       }
403
404       //Functional Entities
405
406       abstract concept FunctionalBaseEntity extends BaseEntity{
407         identifier=FuntionalBaseEntityIdentifier;
408         name="FunctionalBaseEntity";
409       }
410
411       //Organizational Entities
412
413       abstract concept OrganizationalBaseEntity extends BaseEntity{
414         identifier=OrganizationalBaseEntityIdentifier;
415         name="OrganizationalBaseEntity";
416       }
417
418       concept Person extends OrganizationalBaseEntity{
419         name="Person";
420         identifier=PersonIdentifier;
421         relation=DateOfBirthRelation;
422         attribute=firstName,lastName;
423       }
424
425       concept Institute extends OrganizationalBaseEntity{
426         name="Intitute";
```

```
427         identifier=InstituteIdentifier;
428         attribute=entityName,shortName;
429       }
430
431       concept Company extends OrganizationalBaseEntity{
432         name="Company";
433         identifier=CompanyIdentifier;
434         attribute=entityName,shortName;
435       }
436
437       //Behavioural Entities
438
439       abstract concept BehaviouralBaseEntity extends BaseEntity{
440         identifier=BehaviouralBaseEntityIdentifier;
441         name="BehaviouralBaseEntity";
442       }
443
444
445       //Operational Entities
446
447       abstract concept OperationalBaseEntity extends BaseEntity{
448         identifier=OperationalBaseEntityIdentifier;
449         name="BehaviouralBaseEntity";
450       }
451
452       abstract concept Device extends OperationalBaseEntity{
453         identifier=DeviceIdentifier;
454         name="Device";
455         relation=ProducerRelation;
456       }
457
458       concept BaseConversionMethod extends OperationalBaseEntity{
459         name="BaseConversionMethod";
460         identifier=BaseConversionMethodIdentifier;
461       }
462
463       concept StringCVConversionMethod extends BaseConversionMethod{
464         name="StringCVConversion";
465         identifier=StringCVConversionMethodIdentifier;
466       }
467
468       concept StringNumberConversionMethod extends BaseConversionMethod
              {
469         name="StringNumberConversion";
470         identifier=StringNumberConversionMethodIdentifier;
```

```
471     }
472
473     concept NumberNumberConversionMethod extends BaseConversionMethod
            {
474       name="NumberNumberConversion";
475       identifier=NumberNumberConversionMethodIdentifier;
476       attribute=factor;
477     }
478
479     concept UnitConversionMethod extends NumberNumberConversionMethod
            {
480       name="UnitConversion";
481       identifier=UnitConversionMethodIdentifier;
482       relation=OriginalUnitRelation, TargetUnitRelation;
483     }
484
485
486
487     //Local Entities
488
489     abstract concept LocalBaseEntity extends BaseEntity{
490       identifier=LocalBaseEntityIdentifier;
491       name="LocalBaseEntity";
492     }
493
494     concept StoragePlace extends LocalBaseEntity{
495       identifier=StoragePlaceIdentifier;
496       name="StoragePlace";
497     }
498
499     concept PhysicalStoragePlace extends StoragePlace{
500       identifier=PhysicalStoragePlaceIdentifier;
501       name="PhysicalStoragePlace";
502       attribute=building,room,position;
503     }
504
505     concept DigitalStoragePlace extends StoragePlace{
506       identifier=DigitalStoragePlaceIdentifier;
507       name="DigitalStoragePlace";
508       attribute=fileName,path,repository;
509     }
510
511     //Temporal Entities
512
513     abstract concept TemporalBaseEntity extends BaseEntity{
```

```
514         identifier=TemporalBaseEntityIdentifier;
515         name="TemporalBaseEntity";
516       }
517
518       concept Date extends TemporalBaseEntity{
519         identifier=DateIdentifier;
520         name="Date";
521         attribute=day,month,year;
522       }
523
524       concept Time extends TemporalBaseEntity{
525         identifier=TimeIdentifier;
526         name="Time";
527         attribute=hour,minute,second,timezone;
528       }
529
530       concept DateTime extends Date,Time{
531         identifier=DateTimeIdentifier;
532         name="DateTime";
533       }
534
535
536       //DataOriented Entities
537
538       abstract concept DataOrientedBaseEntity extends BaseEntity{
539         identifier=DataOrientedBaseEntityIdentifier;
540         name="DataOrientedBaseEntity";
541       }
542
543       concept ProvenanceTable extends DataOrientedBaseEntity{
544         name="ProvenaceTable";
545         identifier=ProvenanceTableIdentifier;
546         relation=OriginalRelation,SiblingsRelation,ConversionRelation;
547         //Last Modified from ExecutionTime of ConversionRelation;
548       }
549
550       concept ExternalReference extends DataOrientedBaseEntity{
551         name="ExternalReference";
552         identifier=ExternalReferenceIdentifier;
553       }
554
555       concept ExternalDocument extends ExternalReference{
556         name="ExternalDocument";
557         identifier=ExternalDocumentIdentifier;
558         relation=StoragePlaceRelation;
```

```
559       }
560
561       concept WrittenDocument extends ExternalDocument{
562         name="WrittenDocument";
563         identifier=WrittenDocumentIdentifier;
564         attribute=language,remarks;
565         relation=PhysicalStoragePlaceRelation;
566       }
567
568       concept DigitalDocument extends ExternalDocument{
569         name="DigitalDocument";
570         identifier=DigitalDocumentIdentifier;
571         relation=DigitalStoragePlaceRelation;
572       }
573
574       concept Unit extends DataOrientedBaseEntity{
575         name="Unit";
576         identifier=UnitIdentifier;
577         attribute=entityName,symbol;
578       }
579
580       /*Definition of Attributes
581        *
582        */
583
584       //General
585       Attribute entityName{
586         string value;
587         type="string";
588         identifier=entityNameIdentifier;
589
590       }
591
592       Attribute entityCode{
593         string value;
594         type="string";
595         identifier=entityCodeIdentifier;
596
597       }
598
599       Attribute remarks{
600         string value;
601         type="string";
602         identifier=remarksIdentifier;
603         isNullable=true;
```

```
604         }
605
606         Attribute language{
607           string value;
608           type="string";
609           identifier=languageIdentifier;
610         }
611
612         //Organizational Attributes
613
614         Attribute firstName{
615           string value;
616           type="string";
617           identifier=firstNameIdentifier;
618
619         }
620         Attribute lastName{
621           string value;
622           type="string";
623           identifier=lastNameIdentifier;
624         }
625
626         Attribute shortName{
627           string value;
628           type="string";
629           identifier=shortNameIdentifier;
630           isNullable=true;
631         }
632
633
634
635         //Temporal Attributes
636
637         Attribute day{
638           integer value;
639           type="integer";
640           identifier=dayIdentifier;
641
642         }
643
644         Attribute month{
645           integer value;
646           type="integer";
647           identifier=monthIdentifier;
648         }
```

```
649
650       Attribute year{
651         integer value;
652         type="integer";
653         identifier=yearIdentifier;
654       }
655
656       Attribute hour{
657         integer value;
658         type="integer";
659         identifier=hourIdentifier;
660
661       }
662
663       Attribute minute{
664         integer value;
665         type="integer";
666         identifier=minuteIdentifier;
667       }
668
669       Attribute second{
670         integer value;
671         type="integer";
672         identifier=secondIdentifier;
673       }
674
675       Attribute timezone{
676         string value;
677         type="string";
678         identifier=timezoneIdentifier;
679       }
680
681       Attribute factor{
682         real value;
683         type="real";
684         identifier=factorIdentifier;
685       }
686
687       Attribute symbol{
688         string value;
689         type="string";
690         identifier=symbolIdentifier;
691       }
692
693       //Local Attributes
```

```
694
695         Attribute building{
696           string value;
697           type="string";
698           identifier=buildingIdentifier;
699         }
700
701         Attribute room{
702           string value;
703           type="string";
704           identifier=roomIdentifier;
705         }
706
707         Attribute position{
708           string value;
709           type="string";
710           identifier=positionIdentifier;
711           isNullable=true;
712         }
713
714         Attribute fileName{
715           string value;
716           type="string";
717           identifier=fileNameIdentifier;
718         }
719
720         Attribute path{
721           string value;
722           type="string";
723           identifier=pathIdentifier;
724         }
725
726         Attribute repository{
727           string value;
728           type="string";
729           identifier=repositoryIdentifier;
730         }
731
732
733         /*Definition of Controlled Vocabulary
734          *
735          */
736
737         /*Definition of Identifier
738          *
```

```
739         */
740
741
742       //AspectIdentifier
743
744       ConceptIdentifier BaseAspectIdentifier{
745         name="PODSLM1Core/BaseAspect";
746         repository="www.ai4.uni-bayreuth.de";
747         address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/BaseAspect"
              ;
748       }
749
750       ConceptIdentifier FunctionalAspectIdentifier{
751         name="PODSLM1Core/FunctionalAspect";
752         repository="www.ai4.uni-bayreuth.de";
753         address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
              FunctionalAspect";
754       }
755
756       ConceptIdentifier OrganizationalAspectIdentifier{
757         name="PODSLM1Core/OrganizationalAspect";
758         repository="www.ai4.uni-bayreuth.de";
759         address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
              OrganizationalAspect";
760       }
761
762       ConceptIdentifier BehaviouralAspectIdentifier{
763         name="PODSLM1Core/BehaviouralAspect";
764         repository="www.ai4.uni-bayreuth.de";
765         address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
              BehaviouralAspect";
766       }
767
768       ConceptIdentifier OperationalAspectIdentifier{
769         name="PODSLM1Core/OperationalAspect";
770         repository="www.ai4.uni-bayreuth.de";
771         address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
              OperationalAspect";
772       }
773
774       ConceptIdentifier LocalAspectIdentifier{
775         name="PODSLM1Core/LocalAspect";
776         repository="www.ai4.uni-bayreuth.de";
777         address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/LocalAspect
              ";
```

```
778         }
779
780         ConceptIdentifier TemporalAspectIdentifier{
781           name="PODSLM1Core/TemporalAspect";
782           repository="www.ai4.uni-bayreuth.de";
783           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  TemporalAspect";
784         }
785
786         ConceptIdentifier DataOrientedAspectIdentifier{
787           name="PODSLM1Core/DateOrientedAspect";
788           repository="www.ai4.uni-bayreuth.de";
789           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  DataOrientedAspect";
790         }
791
792         //ModalityIdentifier
793
794         ConceptIdentifier BaseModalityIdentifier{
795           name="PODSLM1Core/BaseModality";
796           repository="www.ai4.uni-bayreuth.de";
797           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  BaseModality";
798         }
799
800         ConceptIdentifier PartWholeModalityIdentifier{
801           name="PODSLM1Core/PartWholeModality";
802           repository="www.ai4.uni-bayreuth.de";
803           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  PartWholeModality";
804         }
805
806         ConceptIdentifier BaseConversionModalityIdentifier{
807           name="PODSLM1Core/BaseConversionModality";
808           repository="www.ai4.uni-bayreuth.de";
809           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  BaseConversionModality";
810         }
811
812         //RelationIdentifier
813         ConceptIdentifier BaseRelationIdentifier{
814           name="PODSLM1Core/BaseRelation";
815           repository="www.ai4.uni-bayreuth.de";
816           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  BaseRelation";
```

```
817         }
818
819         ConceptIdentifier BaseOneRelationIdentifier{
820           name="PODSLM1Core/BaseOneRelationy";
821           repository="www.ai4.uni-bayreuth.de";
822           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  BaseOneRelation";
823         }
824
825
826         ConceptIdentifier BaseOneOrMoreRelationIdentifier{
827           name="PODSLM1Core/BaseOneOrMoreRelation";
828           repository="www.ai4.uni-bayreuth.de";
829           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  BaseOneOrMoreRelation";
830         }
831
832         ConceptIdentifier BaseZeroOrMoreRelationIdentifier{
833           name="PODSLM1Core/BaseZeroOrMoreRelation";
834           repository="www.ai4.uni-bayreuth.de";
835           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  BaseZeroOrMoreRelation";
836         }
837
838
839         ConceptIdentifier SubProcessRelationIdentifier{
840           name="PODSLM1Core/SubProcessRelation";
841           repository="www.ai4.uni-bayreuth.de";
842           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  SubProcessRelation";
843         }
844
845
846         ConceptIdentifier PartWholeRelationIdentifier{
847           name="PODSLM1Core/PartWholeRelation";
848           repository="www.ai4.uni-bayreuth.de";
849           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  PartWholeRelation";
850         }
851
852         ConceptIdentifier ObjectRelationIdentifier{
853           name="PODSLM1Core/ObjectRelation";
854           repository="www.ai4.uni-bayreuth.de";
855           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  ObjectRelation";
```

```
856         }
857
858         ConceptIdentifier ResponsibleRelationIdentifier{
859           name="PODSLM1Core/ResponsibleRelation";
860           repository="www.ai4.uni-bayreuth.de";
861           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                ResponsibleRelation";
862         }
863
864         ConceptIdentifier OwnershipRelationIdentifier{
865           name="PODSLM1Core/OwnershipRelation";
866           repository="www.ai4.uni-bayreuth.de";
867           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                OwnershipRelation";
868         }
869
870         ConceptIdentifier ProducerRelationIdentifier{
871           name="PODSLM1Core/ProducerRelation";
872           repository="www.ai4.uni-bayreuth.de";
873           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                ProducerRelation";
874         }
875
876         ConceptIdentifier ExecutionTimeRelationIdentifier{
877           name="PODSLM1Core/ExecutionTimeRelation";
878           repository="www.ai4.uni-bayreuth.de";
879           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                ExecutionTimeRelation";
880         }
881
882         ConceptIdentifier TimeSpanRelationIdentifier{
883           name="PODSLM1Core/TimeSpanRelation";
884           repository="www.ai4.uni-bayreuth.de";
885           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                TimeSpanRelation";
886         }
887
888         ConceptIdentifier DateOfBirthRelationIdentifier{
889           name="PODSLM1Core/DateOfBirthRelation";
890           repository="www.ai4.uni-bayreuth.de";
891           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                DateOfBirthRelation";
892         }
893
894         ConceptIdentifier ExecutionLocalityRelationIdentifier{
```

```
895           name="PODSLM1Core/ExecutionLocalityRelation";
896           repository="www.ai4.uni-bayreuth.de";
897           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  ExecutionLocalityRelation";
898         }
899
900         ConceptIdentifier StoragePlaceRelationIdentifier{
901           name="PODSLM1Core/StoragePlaceRelation";
902           repository="www.ai4.uni-bayreuth.de";
903           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  StoragePlaceRelation";
904         }
905
906         ConceptIdentifier PhysicalStoragePlaceRelationIdentifier{
907           name="PODSLM1Core/PhysicalStoragePlaceRelation";
908           repository="www.ai4.uni-bayreuth.de";
909           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  PhysicalStoragePlaceRelation";
910         }
911
912         ConceptIdentifier DigitalStoragePlaceRelationIdentifier{
913           name="PODSLM1Core/DigitalStoragePlaceRelation";
914           repository="www.ai4.uni-bayreuth.de";
915           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  DigitalStoragePlaceRelation";
916         }
917
918         ConceptIdentifier DataSetsRelationIdentifier{
919           name="PODSLM1Core/DataSetsRelation";
920           repository="www.ai4.uni-bayreuth.de";
921           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  DataSetsRelation";
922         }
923
924         ConceptIdentifier ProcessExecutionDocumentsRelationIdentifier{
925           name="PODSLM1Core/ProcessExecutionDocumentsRelation";
926           repository="www.ai4.uni-bayreuth.de";
927           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  ProcessExecutionDocumentsRelation";
928         }
929
930         ConceptIdentifier ReferenceRelationIdentifier{
931           name="PODSLM1Core/ReferenceRelationIdentifier";
932           repository="www.ai4.uni-bayreuth.de";
```

```
933         address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                ReferenceRelationIdentifier";
934       }
935
936       ConceptIdentifier ExternalDocumentRelationIdentifier{
937         name="PODSLM1Core/ExternalDocumentRelation";
938         repository="www.ai4.uni-bayreuth.de";
939         address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                ExternalDocumentRelation";
940       }
941
942       ConceptIdentifier ProvenanceRelationIdentifier{
943         name="PODSLM1Core/ProvenanceRelation";
944         repository="www.ai4.uni-bayreuth.de";
945         address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                ProvenanceRelation";
946       }
947
948       ConceptIdentifier OriginalRelationIdentifier{
949         name="PODSLM1Core/OriginalRelation";
950         repository="www.ai4.uni-bayreuth.de";
951         address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                OriginalRelation";
952       }
953
954       ConceptIdentifier SiblingsRelationIdentifier{
955         name="PODSLM1Core/SiblingsRelation";
956         repository="www.ai4.uni-bayreuth.de";
957         address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                SiblingsRelation";
958       }
959
960       ConceptIdentifier ConversionRelationIdentifier{
961         name="PODSLM1Core/ConversionRelation";
962         repository="www.ai4.uni-bayreuth.de";
963         address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                ConversionRelation";
964       }
965
966       ConceptIdentifier ConversionOutputRelationIdentifier{
967         name="PODSLM1Core/ConversionOutputRelation";
968         repository="www.ai4.uni-bayreuth.de";
969         address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                ConversionOutputRelation";
970       }
```

```
971
972         ConceptIdentifier ConversionMethodRelationIdentifier{
973           name="PODSLM1Core/ConversionMethodRelation";
974           repository="www.ai4.uni-bayreuth.de";
975           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  ConversionMethodRelation";
976         }
977
978         ConceptIdentifier OriginalUnitRelationIdentifier{
979           name="PODSLM1Core/OriginalUnitRelation";
980           repository="www.ai4.uni-bayreuth.de";
981           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  OriginalUnitRelation";
982         }
983
984         ConceptIdentifier TargetUnitRelationIdentifier{
985           name="PODSLM1Core/TargetUnitRelationRelation";
986           repository="www.ai4.uni-bayreuth.de";
987           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  TargetUnitRelationRelation";
988         }
989
990         ConceptIdentifier CreationTimeRelationIdentifier{
991           name="PODSLM1Core/CreationTimeRelation";
992           repository="www.ai4.uni-bayreuth.de";
993           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  CreationTimeRelation";
994         }
995
996         ConceptIdentifier CreatorRelationIdentifier{
997           name="PODSLM1Core/CreatorRelation";
998           repository="www.ai4.uni-bayreuth.de";
999           address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  CreatorRelation";
1000        }
1001
1002        //EntityIdentifier
1003
1004        ConceptIdentifier BaseEntityIdentifier{
1005          name="PODSLM1Core/BaseEntity";
1006          repository="www.ai4.uni-bayreuth.de";
1007          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/BaseEntity"
                  ;
1008        }
1009
```

```
1010        ConceptIdentifier BaseDataSetIdentifier{
1011          name="PODSLM1Core/BaseDataSet";
1012          repository="www.ai4.uni-bayreuth.de";
1013          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/BaseDataSet
                  ";
1014        }
1015
1016        ConceptIdentifier FuntionalBaseEntityIdentifier{
1017          name="PODSLM1Core/FuntionalBaseEntity";
1018          repository="www.ai4.uni-bayreuth.de";
1019          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  FuntionalBaseEntity";
1020        }
1021
1022        ConceptIdentifier OrganizationalBaseEntityIdentifier{
1023          name="PODSLM1Core/OrganizationalBaseEntity";
1024          repository="www.ai4.uni-bayreuth.de";
1025          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  OrganizationalBaseEntity";
1026        }
1027
1028        ConceptIdentifier BehaviouralBaseEntityIdentifier{
1029          name="PODSLM1Core/BehaviouralBaseEntity";
1030          repository="www.ai4.uni-bayreuth.de";
1031          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  BehaviouralBaseEntity";
1032        }
1033
1034        ConceptIdentifier OperationalBaseEntityIdentifier{
1035          name="PODSLM1Core/OperationalBaseEntity";
1036          repository="www.ai4.uni-bayreuth.de";
1037          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  OperationalBaseEntity";
1038        }
1039
1040        ConceptIdentifier LocalBaseEntityIdentifier{
1041          name="PODSLM1Core/LocalBaseEntity";
1042          repository="www.ai4.uni-bayreuth.de";
1043          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  LocalBaseEntity";
1044        }
1045
1046        ConceptIdentifier TemporalBaseEntityIdentifier{
1047          name="PODSLM1Core/TemporalBaseEntity";
1048          repository="www.ai4.uni-bayreuth.de";
```

```
1049          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  TemporalBaseEntity";
1050        }
1051
1052        ConceptIdentifier DataOrientedBaseEntityIdentifier{
1053          name="PODSLM1Core/DataOrientedBaseEntity";
1054          repository="www.ai4.uni-bayreuth.de";
1055          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  DataOrientedBaseEntity";
1056        }
1057
1058        ConceptIdentifier PersonIdentifier{
1059          name="PODSLM1Core/Person";
1060          repository="www.ai4.uni-bayreuth.de";
1061          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/Person";
1062        }
1063
1064        ConceptIdentifier InstituteIdentifier{
1065          name="PODSLM1Core/Institute";
1066          repository="www.ai4.uni-bayreuth.de";
1067          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/Institute";
1068        }
1069
1070        ConceptIdentifier CompanyIdentifier{
1071          name="PODSLM1Core/Company";
1072          repository="www.ai4.uni-bayreuth.de";
1073          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/Company";
1074        }
1075
1076        ConceptIdentifier DeviceIdentifier{
1077          name="PODSLM1Core/Device";
1078          repository="www.ai4.uni-bayreuth.de";
1079          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/Device";
1080        }
1081
1082        ConceptIdentifier DateIdentifier{
1083          name="PODSLM1Core/Date";
1084          repository="www.ai4.uni-bayreuth.de";
1085          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/Date";
1086        }
1087
1088        ConceptIdentifier TimeIdentifier{
1089          name="PODSLM1Core/Time";
1090          repository="www.ai4.uni-bayreuth.de";
1091          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/Time";
```

```
1092        }
1093
1094        ConceptIdentifier DateTimeIdentifier{
1095          name="PODSLM1Core/DateTime";
1096          repository="www.ai4.uni-bayreuth.de";
1097          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/DateTime";
1098        }
1099
1100        ConceptIdentifier StoragePlaceIdentifier{
1101          name="PODSLM1Core/StoragePlace";
1102          repository="www.ai4.uni-bayreuth.de";
1103          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                StoragePlace";
1104        }
1105
1106        ConceptIdentifier PhysicalStoragePlaceIdentifier{
1107          name="PODSLM1Core/PhysicalStoragePlace";
1108          repository="www.ai4.uni-bayreuth.de";
1109          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                PhysicalStoragePlace";
1110        }
1111
1112        ConceptIdentifier DigitalStoragePlaceIdentifier{
1113          name="PODSLM1Core/DigitalStoragePlace";
1114          repository="www.ai4.uni-bayreuth.de";
1115          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                DigitalStoragePlace";
1116        }
1117
1118        ConceptIdentifier ProvenanceTableIdentifier{
1119          name="PODSLM1Core/ProvenanceTable";
1120          repository="www.ai4.uni-bayreuth.de";
1121          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                ProvenanceTable";
1122        }
1123
1124        ConceptIdentifier BaseConversionMethodIdentifier{
1125          name="PODSLM1Core/BaseConversionMethod";
1126          repository="www.ai4.uni-bayreuth.de";
1127          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                BaseConversionMethod";
1128        }
1129
1130        ConceptIdentifier StringCVConversionMethodIdentifier{
1131          name="PODSLM1Core/StringCVConversionMethod";
```

```
1132          repository="www.ai4.uni-bayreuth.de";
1133          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  StringCVConversionMethod";
1134        }
1135
1136        ConceptIdentifier StringNumberConversionMethodIdentifier{
1137          name="PODSLM1Core/StringNumberConversionMethod";
1138          repository="www.ai4.uni-bayreuth.de";
1139          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  StringNumberConversionMethod";
1140        }
1141
1142        ConceptIdentifier NumberNumberConversionMethodIdentifier{
1143          name="PODSLM1Core/NumberNumberConversionMethod";
1144          repository="www.ai4.uni-bayreuth.de";
1145          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  NumberNumberConversionMethod";
1146        }
1147
1148        ConceptIdentifier UnitConversionMethodIdentifier{
1149          name="PODSLM1Core/UnitConversionMethod";
1150          repository="www.ai4.uni-bayreuth.de";
1151          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  UnitConversionMethod";
1152        }
1153
1154        ConceptIdentifier UnitIdentifier{
1155          name="PODSLM1Core/Unit";
1156          repository="www.ai4.uni-bayreuth.de";
1157          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/Unit";
1158        }
1159
1160        ConceptIdentifier ExternalReferenceIdentifier{
1161          name="PODSLM1Core/ExternalReference";
1162          repository="www.ai4.uni-bayreuth.de";
1163          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  ExternalReference";
1164        }
1165
1166        ConceptIdentifier ExternalDocumentIdentifier{
1167          name="PODSLM1Core/ExternalDocument";
1168          repository="www.ai4.uni-bayreuth.de";
1169          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  ExternalDocument";
1170        }
```

```
1171
1172        ConceptIdentifier WrittenDocumentIdentifier{
1173          name="PODSLM1Core/WrittenDocument";
1174          repository="www.ai4.uni-bayreuth.de";
1175          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  WrittenDocument";
1176        }
1177
1178        ConceptIdentifier DigitalDocumentIdentifier{
1179          name="PODSLM1Core/DigitalDocument";
1180          repository="www.ai4.uni-bayreuth.de";
1181          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  DigitalDocument";
1182        }
1183
1184        //ProcessExecution
1185
1186        ConceptIdentifier BaseProcessExecutionDocumentIdentifier{
1187          name="PODSLM1Core/BaseProcessExecutionDocument";
1188          repository="www.ai4.uni-bayreuth.de";
1189          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  BaseProcessExecutionDocument";
1190        }
1191
1192        ConceptIdentifier ConversionProcessDocumentIdentifier{
1193          name="PODSLM1Core/ConversionProcessDocument";
1194          repository="www.ai4.uni-bayreuth.de";
1195          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/
                  ConversionProcessDocument";
1196        }
1197
1198
1199        //AttributeIdentifier
1200
1201        ConceptIdentifier entityNameIdentifier{
1202          name="PODSLM1Core/entityName";
1203          repository="www.ai4.uni-bayreuth.de";
1204          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/entityName"
                  ;
1205        }
1206
1207        ConceptIdentifier entityCodeIdentifier{
1208          name="PODSLM1Core/entityCode";
1209          repository="www.ai4.uni-bayreuth.de";
```

```
1210          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/entityCode"
                  ;
1211        }
1212
1213        ConceptIdentifier remarksIdentifier{
1214          name="PODSLM1Core/remarks";
1215          repository="www.ai4.uni-bayreuth.de";
1216          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/remarks";
1217        }
1218
1219        ConceptIdentifier languageIdentifier{
1220          name="PODSLM1Core/language";
1221          repository="www.ai4.uni-bayreuth.de";
1222          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/language";
1223        }
1224
1225        ConceptIdentifier firstNameIdentifier{
1226          name="PODSLM1Core/firstName";
1227          repository="www.ai4.uni-bayreuth.de";
1228          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/firstName";
1229        }
1230
1231        ConceptIdentifier lastNameIdentifier{
1232          name="PODSLM1Core/lastName";
1233          repository="www.ai4.uni-bayreuth.de";
1234          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/lastName";
1235        }
1236
1237        ConceptIdentifier shortNameIdentifier{
1238          name="PODSLM1Core/shortName";
1239          repository="www.ai4.uni-bayreuth.de";
1240          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/shortName";
1241        }
1242
1243        ConceptIdentifier dayIdentifier{
1244          name="PODSLM1Core/day";
1245          repository="www.ai4.uni-bayreuth.de";
1246          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/day";
1247        }
1248
1249        ConceptIdentifier monthIdentifier{
1250          name="PODSLM1Core/month";
1251          repository="www.ai4.uni-bayreuth.de";
1252          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/month";
1253        }
```

```
1254
1255        ConceptIdentifier yearIdentifier{
1256          name="PODSLM1Core/year";
1257          repository="www.ai4.uni-bayreuth.de";
1258          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/year";
1259        }
1260
1261        ConceptIdentifier hourIdentifier{
1262          name="PODSLM1Core/hour";
1263          repository="www.ai4.uni-bayreuth.de";
1264          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/hour";
1265        }
1266
1267        ConceptIdentifier minuteIdentifier{
1268          name="PODSLM1Core/minute";
1269          repository="www.ai4.uni-bayreuth.de";
1270          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/minute";
1271        }
1272
1273        ConceptIdentifier secondIdentifier{
1274          name="PODSLM1Core/second";
1275          repository="www.ai4.uni-bayreuth.de";
1276          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/second";
1277        }
1278
1279        ConceptIdentifier timezoneIdentifier{
1280          name="PODSLM1Core/timezone";
1281          repository="www.ai4.uni-bayreuth.de";
1282          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/timezone";
1283        }
1284
1285        ConceptIdentifier factorIdentifier{
1286          name="PODSLM1Core/factor";
1287          repository="www.ai4.uni-bayreuth.de";
1288          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/factor";
1289        }
1290
1291        ConceptIdentifier symbolIdentifier{
1292          name="PODSLM1Core/symbol";
1293          repository="www.ai4.uni-bayreuth.de";
1294          address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/symbol";
1295        }
1296
1297        ConceptIdentifier buildingIdentifier{
1298          name="PODSLM1Core/building";
```

```
1299            repository="www.ai4.uni-bayreuth.de";
1300            address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/building";
1301        }
1302
1303        ConceptIdentifier roomIdentifier{
1304            name="PODSLM1Core/room";
1305            repository="www.ai4.uni-bayreuth.de";
1306            address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/room";
1307        }
1308
1309        ConceptIdentifier positionIdentifier{
1310            name="PODSLM1Core/position";
1311            repository="www.ai4.uni-bayreuth.de";
1312            address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/position";
1313        }
1314
1315        ConceptIdentifier fileNameIdentifier{
1316            name="PODSLM1Core/fileName";
1317            repository="www.ai4.uni-bayreuth.de";
1318            address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/fileName";
1319        }
1320
1321        ConceptIdentifier pathIdentifier{
1322            name="PODSLM1Core/room";
1323            repository="www.ai4.uni-bayreuth.de";
1324            address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/path";
1325        }
1326
1327        ConceptIdentifier repositoryIdentifier{
1328            name="PODSLM1Core/repository";
1329            repository="www.ai4.uni-bayreuth.de";
1330            address="model:/www.ai4.uni-bayreuth.de/PODSLM1Core/repository"
                    ;
1331        }
1332
1333    }
1334
1335  }
1336
1337 }
```

Listing C.2: $M_1 - Core$ von PODSL

C.3 PODSL-Biodiv

```
1  model PODSLM1Biodiv {
2    uri "model:/www.ai4.uni-bayreuth.de/PODSLM1Biodiv";
3    include "model:/www.ai4.uni-bayreuth.de/PODSLM2";
4    include "model:/www.ai4.uni-bayreuth.de/PODSLM1Core";
5
6    level M1 alignedWith PODSLM1Core.M1
7    {
8      package PODSLM1Biodiv{
9      import PODSLM2.M2.PODSLM2.*;
10     import PODSLM1Core.M1.PODSLM1Core.*;
11
12     /*
13      * Modalities
14      */
15
16     concept SecurityOfDeterminationModality extends BaseModality{
17       name="SecurityOfDetermination";
18       identifier=SecurityOfDeterminationModalityIdentifier;
19       attribute=securityOfDetermination;
20     }
21
22     concept VerificationOfDeterminationModality extends BaseModality{
23       name="VerificationOfDetermination";
24       identifier=VerificationOfDeterminationModalityIdentifier;
25       attribute=verificationStatus;
26     }
27
28     concept FormOfCoexistenceModality extends BaseModality{
29       name="FormOfCoexistenceModality";
30       identifier=FormOfCoexistenceModalityIdentifier;
31       attribute=formOfCoexistence;
32     }
33
34     concept OwnershipModality extends BaseModality{
35       name="OwnershipModality";
36       identifier=OwnershipModalityIdentifier;
37       attribute=typeOfOwnership;
38     }
39
40     concept AffiliationModality extends BaseModality{
41       name="AffiliationModality";
42       identifier=AffiliationModalityIdentifier;
```

```
43        attribute=typeOfAffiliation;
44      }
45
46      concept LocalitySurfaceDescriptionModality extends BaseModality{
47        name="LocalitySurfaceDescriptionModality";
48        identifier=LocalitySurfaceDescriptionModalityIdentifier;
49        attribute=typeOfSurface;
50      }
51
52
53      /*
54       * Relations
55       */
56
57      //Specific Object Relations
58
59      concept ObservationObjectRelation extends ObjectRelation{
60        role="main object in a observation";
61        identifier=ObjectObservationRelationIdentifier;
62        source=BaseProcessExecutionDocument;
63        target=ObservationObject;
64      }
65
66      concept BiologicalObjectRelation extends ObservationObjectRelation{
67        role="BiologicalObject in a observation";
68        identifier=BiologicalObjectRelationIdentifier;
69        source=BaseProcessExecutionDocument;
70        target=BiologicalObject;
71      }
72
73      concept LocalityObjectRelation extends ObservationObjectRelation{
74        role="LocalityObject in a observation";
75        identifier=LocalityObjectRelationIdentifier;
76        source=BaseProcessExecutionDocument;
77        target=LocalityObject;
78      }
79
80      concept SpecimenRelation extends ObjectRelation{
81        role="Object from which a Specimen is taken";
82        identifier=SpecimenRelationIdentifier;
83        source=BaseProcessExecutionDocument;
84        target=ObservationObject;
85      }
86
87      concept SiteRelation extends BaseOneRelation{
```

```
88        role="Site which is observed in a process";
89        identifier=SiteRelationIdentifier;
90        source=BaseProcessExecutionDocument;
91        target=Site;
92      }
93
94
95      //Taxon
96
97      concept AcceptedNameTaxonRelation extends BaseZeroOrOneRelation{
98        role="accepted Name of Taxon";
99        identifier= AcceptedNameTaxonRelationIdentifier;
100       source=TaxonLink;
101       target=BaseTaxon;
102     }
103
104     concept ParentNameTaxonRelation extends BaseZeroOrOneRelation{
105       role="parent Name of Taxon";
106       identifier= ParentNameTaxonRelationIdentifier;
107       source=TaxonLink;
108       target=BaseTaxon;
109     }
110
111     concept OriginalNameTaxonRelation extends BaseZeroOrOneRelation{
112       role="original Name of Taxon";
113       identifier= OriginalNameTaxonRelationIdentifier;
114       source=TaxonLink;
115       target=BaseTaxon;
116     }
117
118     //Organizational
119
120     concept InstituteCollectionRelation extends BaseOneOrMoreRelation{
121       role="is custodian Of";
122       identifier=InstituteCollectionRelationIdentifier;
123       aspect=OrganizationalAspect;
124       source=Institute;
125       target=Collection;
126     }
127
128
129
130     concept InstitutePersonRelation extends BaseOneOrMoreRelation{
131       role="Institute has affiliations with persons";
132       identifier=InstitutePersonRelationIdentifier;
```

```
133       aspect=OrganizationalAspect;
134       source=Institute;
135       target=Person;
136       modality=AffiliationModality;
137     }
138
139     concept InstituteITSystemRelation extends BaseOneOrMoreRelation{
140       role="Institute operates ITSystems";
141       identifier=InstituteITSystemRelationIdentifier;
142       aspect=OrganizationalAspect;
143       source=Institute;
144       target=ITSystem;
145     }
146
147
148     concept ResponsibleScientistRelation extends ResponsibleRelation{
149       role="Scientist responsible for a process";
150       identifier=ResponsibleScientistRelationIdentifier;
151       target=Scientist;
152     }
153
154     concept ResponsibleLaboratoryTechnicianRelation extends
            ResponsibleRelation{
155       role="LaboratoryTechnician responsible for a process";
156       identifier=ResponsibleLaboratoryTechnicianRelationIdentifier;
157       target=LaboratoryTechnician;
158     }
159
160     concept ResponsibleGathererRelation extends
            ResponsibleScientistRelation{
161       role="Scientist as a Gatherer in a process";
162       identifier=ResponsibleGathererRelationIdentifier;
163       target=Gatherer;
164     }
165
166     concept ResponsibleCuratorRelation extends
            ResponsibleScientistRelation{
167       role="Scientist as a Curator in a process";
168       identifier=ResponsibleCuratorRelationIdentifier;
169       target=Curator;
170     }
171
172     concept ResponsibleITSystemRelation extends ResponsibleRelation{
173       role="ITSystem responsible for a process";
174       identifier=ResponsibleITSystemRelationIdentifier;
```

```
175       source=BaseProcessExecutionDocument;
176       target=ITSystem;
177     }
178
179
180     concept AuthorRelation extends BaseOneOrMoreRelation{
181       role="hasAuthor";
182       identifier=AuthorRelationIdentifier;
183       source=WorkOfReference;
184       target=Person;
185     }
186
187     concept DataSetCollectionRelation extends BaseOneRelation{
188       role="belongs to";
189       identifier=DataSetCollectionRelationIdentifier;
190       source=BaseDataSet;
191       target=Collection;
192     }
193
194     concept DataSetLicenseRelation extends BaseOneRelation{
195       role="has license";
196       identifier=DataSetLicenseRelationIdentifier;
197       source=BaseDataSet;
198       target=License;
199     }
200
201     concept AccessRightsRelation extends BaseOneOrMoreRelation{
202       role="target has access rights";
203       identifier=AccessRightsRelationIdentifier;
204       source=BiodivDataSet;
205       target=Person;
206     }
207
208
209     //Functional
210     concept TaxonDeterminationRelation extends BaseOneRelation{
211       role="TaxonDetermination";
212       identifier=TaxonDeterminationRelationIdentifier;
213       aspect=FunctionalAspect;
214       source=BiologicalObject;
215       target=BaseTaxon;
216       modality=SecurityOfDeterminationModality,
              VerificationOfDeterminationModality;
217     }
218
```

```
219
220     concept FormOfCoExistenceRelation extends BaseOneOrMoreRelation{
221       role="FormOfCoexistence";
222       identifier=FormOfCoexistenceRelationIdentifier;
223       aspect=FunctionalAspect;
224       source=BiologicalObject;
225       target=BiologicalObject;
226       modality=FormOfCoexistenceModality;
227     }
228
229
230     concept PreparatingRelation extends BaseOneRelation{
231       role="Preparating";
232       identifier=PreparatingRelationIdentifier;
233       aspect=FunctionalAspect;
234       source=Specimen;
235       target=SpecimenPreparating;
236     }
237
238     //Measurement Relations
239
240     concept MeasurementValueRelation extends BaseOneRelation{
241       role="MeasurementDetermination";
242       identifier=MeasurementValueRelationIdentifier;
243       aspect=FunctionalAspect;
244       source=ObservationObject;
245       target=BaseMeasurement;
246     }
247
248
249     concept CoordinateDeterminationRelation extends
            MeasurementValueRelation{
250       role="CoordinateDetermination";
251       identifier=CoordinateDeterminationRelationIdentifier;
252       aspect=FunctionalAspect;
253       target=Coordinates;
254       modality=VerificationOfDeterminationModality;
255     }
256
257
258     concept NumericMeasurementValueRelation extends
            MeasurementValueRelation{
259       role="Assignment of a numerical value";
260       identifier=NumericMeasurementValueRelationIdentifier;
261       aspect=FunctionalAspect;
```

```
262       target=NumericMeasurement;
263
264     }
265
266     concept LocalitySurfaceDescriptionRelation extends
            MeasurementValueRelation{
267       role="Description of the Surface of the Locality";
268       identifier=LocalitySurfaceDescriptionRelationIdentifier;
269       target=SurfaceDescription;
270       modality=LocalitySurfaceDescriptionModality;
271     }
272
273
274     concept TagAssignmentRelation extends MeasurementValueRelation{
275       role="TagAssignment";
276       identifier=TagAssignmentRelationIdentifier;
277       aspect=FunctionalAspect;
278       target=Tag;
279
280     }
281
282     concept ObjectDescriptionRelation extends MeasurementValueRelation{
283       role="ObjectDescription";
284       identifier=ObjectDescriptionRelationIdentifier;
285       aspect=FunctionalAspect;
286       target=Description;
287     }
288
289
290
291     concept MeasurementMethodRelation extends BaseOneRelation{
292       role="Method is assignend to Measurement";
293       identifier=MeasurementMethodRelationIdentifier;
294       source=BaseMeasurement;
295       target=BaseMethod;
296     }
297
298
299     //Temporal
300
301     //Local
302
303     concept ExecutionHabitatRelation extends ExecutionLocalityRelation{
304       role="ExecutionHabitat";
305       identifier=ExecutionHabitatRelationIdentifier;
```

```
306       aspect=LocalAspect;
307       target=Description;
308     }
309
310     concept ExternalInstituteRelation extends BaseOneRelation{
311       role="ExternalIntitute in a process";
312       identifier=ExternalInstituteRelationIdentifier;
313       aspect=LocalAspect;
314       source=BaseProcessExecutionDocument;
315       target=Institute;
316     }
317
318
319     concept ParentGeographyRelation extends BaseZeroOrOneRelation{
320       role="parent Name of a Locality";
321       identifier= ParentGeographyRelationIdentifier;
322       source=ParentGeography;
323       target=LocalityDescription;
324     }
325
326     //DataOriented
327
328     concept WorkOfReferenceRelation extends BaseZeroOrMoreRelation{
329       role="Reference in literature";
330       identifier=WorkOfReferenceRelationIdentifier;
331       aspect=OperationalAspect;
332       source=BaseEntity;
333       target=WorkOfReference;
334     }
335
336
337     concept MultimediaDocumentRelation extends BaseOneOrMoreRelation{
338       role="Link toMultiMedia Documentation";
339       identifier=MultimediaDocumentRelationIdentifier;
340       aspect=DataOrientedAspect;
341       source=ObservationObject;
342       target=MultimediaDocument;
343     }
344
345     concept WrittenDocumentationRelation extends BaseOneRelation{
346       role="Link to Written Documentation";
347       identifier=WrittenDocumentationRelationIdentifier;
348       aspect=DataOrientedAspect;
349       source=BaseProcessExecutionDocument;
350       target=WrittenDocumentation;
```

```
351     }
352
353     concept DNAAnalysisRelation extends BaseOneRelation{
354       role="Link to DNAAnalysis";
355       identifier=DNAAnalysisRelationIdentifier;
356       aspect=DataOrientedAspect;
357       source=Specimen;
358       target=DNAAnalysis;
359     }
360
361     concept RelatedProcessDocumentsRelation extends
          BaseOneOrMoreRelation{
362       role="Link to execution of another process";
363       identifier=RelatedProcessDocumentsRelationIdentifier;
364       aspect=DataOrientedAspect;
365       source=BaseProcessExecutionDocument;
366       target=BaseProcessExecutionDocument;
367
368     }
369
370     concept SpecimenRecordRelation extends WrittenDocumentationRelation
          {
371       role="Link to Specimen Record";
372       identifier=SpecimenRecordRelationIdentifier;
373       aspect=DataOrientedAspect;
374       source=Specimen;
375       target=SpecimenRecord;
376
377     }
378
379     concept ObservationRelation extends BaseOneRelation{
380       role="ObservationRelation";
381       identifier=ObservationRelationIdentifier;
382       aspect=DataOrientedAspect;
383       source=BaseProcessExecutionDocument;
384       target=Observation;
385     }
386
387     concept SpecimenGatheringRelation extends BaseOneRelation{
388       role="SpecimenGatheringRelation";
389       identifier=SpecimenGatheringRelationIdentifier;
390       aspect=DataOrientedAspect;
391       source=BaseProcessExecutionDocument;
392       target=SpecimenGathering;
393     }
```

```
394
395     concept SpecimenArchivingRelation extends BaseOneRelation{
396       role="SpecimenArchivingRelation";
397       identifier=SpecimenArchivingRelationIdentifier;
398       aspect=DataOrientedAspect;
399       source=BaseProcessExecutionDocument;
400       target=SpecimenArchiving;
401     }
402
403     concept SpecimenCollectionRelation extends BaseOneRelation{
404       role="Specimen is in Collection";
405       identifier=SpecimenCollectionRelationIdentifier;
406       aspect=DataOrientedAspect;
407       source=Specimen;
408       target=Collection;
409     }
410
411
412     //Operational
413     abstract concept MethodRelation extends BaseOneRelation{
414       role="Used Method";
415       identifier=MethodRelationIdentifier;
416       aspect=OperationalAspect;
417       source=BaseProcessExecutionDocument;
418       target=BaseMethod;
419     }
420
421     concept SamplingMethodRelation extends MethodRelation{
422       role="Description of Sampling Method";
423       identifier=SamplingMethodRelationIdentifier;
424       target=SamplingMethod;
425     }
426
427     concept UsedDeviceRelation extends BaseOneOrMoreRelation{
428       role="Device used in Process";
429       identifier=UsedDeviceRelationIdentifier;
430       aspect=OperationalAspect;
431       source=BaseProcessExecutionDocument;
432       target=Device;
433     }
434
435     concept MapCoordinatesRelation extends MapRelation{
436       role="Map to display Coordinates";
437       aspect=OperationalAspect;
438       identifier=MapCoordinatesRelationIdentifier;
```

```
439       source=Coordinates;
440       target=Map;
441     }
442     concept TKCoordinatesRelation extends MapRelation{
443       role="TK25 to display Coordinates";
444       identifier=TKCoordinatesRelationIdentifier;
445       target=TK;
446     }
447
448     //General
449
450     concept ObservationObjectPartWholeRelation extends
            PartWholeRelation{
451       role="ObservationObjectPartWholeRelation";
452       identifier=ObservationObjectPartWholeRelationIdentifier;
453       source=ObservationObject;
454       target=ObservationObject;
455     }
456
457     concept MapRelation extends BaseOneRelation{
458       role="Relation to a Map";
459       identifier=MapRelationIdentifier;
460       source=BaseEntity;
461       target=Map;
462     }
463
464     //subProcesses
465
466     concept IdentificationSubProcess extends SubProcessRelation{
467       role="Identification is a subprocess in a composite process";
468       identifier=IdentificationSubProcessIdentifier;
469       aspect=FunctionalAspect;
470       target=Identification;
471       cardinality= enum one;
472     }
473
474     concept GeoReferencingSubProcess extends SubProcessRelation{
475       role="GeoReferencing is a subprocess in a composite process";
476       identifier=GeoReferencingSubProcessIdentifier;
477       aspect=FunctionalAspect;
478       target=GeoReferencing;
479       cardinality= enum one;
480     }
481
482     concept SpecimenGatheringSubProcess extends SubProcessRelation{
```

```
483       role="SpecimenGathering is a subprocess in a composite process";
484       identifier=SpecimenGatheringSubProcessIdentifier;
485       aspect=FunctionalAspect;
486       target=SpecimenGathering;
487       cardinality= enum one;
488     }
489
490     concept ObservationSubProcess extends SubProcessRelation{
491       role="Observation is a subprocess in a composite process";
492       identifier=ObservationSubProcessIdentifier;
493       aspect=FunctionalAspect;
494       target=Observation;
495       cardinality= enum one;
496     }
497
498     concept CoExistenceObservationSubProcess extends SubProcessRelation
          {
499       role="CoExistenceObservation is a subprocess in a composite
            process";
500       identifier=CoExistenceObservationSubProcessIdentifier;
501       aspect=FunctionalAspect;
502       target=CoExistenceObservation;
503       cardinality= enum one;
504     }
505
506     concept SpecimenMonitoringSubProcess extends SubProcessRelation{
507       role="SpecimenMonitoring is a subprocess in a composite process";
508       identifier=SpecimenMonitoringSubProcessIdentifier;
509       aspect=FunctionalAspect;
510       target=SpecimenMonitoring;
511       cardinality= enum one;
512     }
513
514     concept SpecimenArchivingSubProcess extends SubProcessRelation{
515       role="SpecimenArchiving is a subprocess in a composite process";
516       identifier=SpecimenArchivingSubProcessIdentifier;
517       aspect=FunctionalAspect;
518       target=SpecimenArchiving;
519       cardinality= enum one;
520     }
521
522     concept MultimediaRecordingSubProcess extends SubProcessRelation{
523       role="MultimediaRecording is a subprocess in a composite process"
            ;
524       identifier=MultimediaRecordingSubProcessIdentifier;
```

```
525       aspect=FunctionalAspect;
526       target=MultimediaRecording;
527       cardinality= enum oneOrMore;
528     }
529
530     concept MultimediaUpdateSubProcess extends SubProcessRelation{
531       role="MultimediaUpdate is a subprocess in a composite process";
532       identifier=MultimediaUpdateSubProcessIdentifier;
533       aspect=FunctionalAspect;
534       target=MultimediaStoragePlaceUpdate;
535       cardinality= enum one;
536     }
537
538     concept MultimediaObservationSubProcess extends SubProcessRelation{
539       role="MultimediaObservation is a subprocess in a composite
              process";
540       identifier=MultimediaObservationSubProcessIdentifier;
541       aspect=FunctionalAspect;
542       target=MultimediaObservation;
543       cardinality= enum oneOrMore;
544     }
545
546     concept RemoveSpecimenSubProcess extends SubProcessRelation{
547       role="RemoveSpecimen is a subprocess in a composite process";
548       identifier=RemoveSpecimenSubProcessIdentifier;
549       aspect=FunctionalAspect;
550       target=RemoveSpecimen;
551       cardinality= enum one;
552     }
553
554     concept DNAAnalysingSubProcess extends SubProcessRelation{
555       role="DNAAnalysing is a subprocess in a composite process";
556       identifier=DNAAnalysingSubProcessIdentifier;
557       aspect=FunctionalAspect;
558       target=DNAAnalysing;
559       cardinality= enum one;
560     }
561
562     concept AddSpecimenSubProcess extends SubProcessRelation{
563       role="AddSpecimen is a subprocess in a composite process";
564       identifier=AddSpecimenSubProcessIdentifier;
565       aspect=FunctionalAspect;
566       target=AddSpecimen;
567       cardinality= enum one;
568     }
```

```
569
570     concept BaseMeasuringSubProcess extends SubProcessRelation{
571       role="BaseMeasuring is a subprocess in a composite process";
572       identifier=BaseMeasuringSubProcessIdentifier;
573       aspect=FunctionalAspect;
574       target=BaseMeasuring;
575       cardinality= enum one;
576     }
577
578     //Multiple Execution
579
580     concept BaseMeasuringMultipleSubProcess extends SubProcessRelation{
581       role="BaseMeasuring is a subprocess in a composite process (
              possible multiple execution)";
582       identifier=BaseMeasuringMultipleSubProcessIdentifier;
583       aspect=FunctionalAspect;
584       target=BaseMeasuring;
585       cardinality= enum oneOrMore;
586     }
587
588     concept BaseMonitoringMultipleSubProcess extends SubProcessRelation
          {
589       role="BaseMonitoring is a subprocess in a composite process (
              possible multiple execution)";
590       identifier=BaseMonitoringMultipleSubProcessIdentifier;
591       aspect=FunctionalAspect;
592       target=BaseMonitoring;
593       cardinality= enum oneOrMore;
594     }
595
596     concept CoExistenceObservationMultipleSubProcess extends
          SubProcessRelation{
597       role="CoExistenceObservation is a subprocess in a composite
              process  (possible multiple execution)";
598       identifier=CoExistenceObservationMultipleSubProcessIdentifier;
599       aspect=FunctionalAspect;
600       target=CoExistenceObservation;
601       cardinality= enum oneOrMore;
602     }
603
604     //Subprocesses for data connections
605
606     concept ConnectSpecimenArchivingWithSpecimenGatheringSubProcess
          extends SubProcessRelation{
```

```
607       role="ConnectSpecimenArchivingWithSpecimenGathering is a
              subprocess in a composite process";
608       identifier=
              ConnectSpecimenArchivingWithSpecimenGatheringSubProcessIdentifier
              ;
609       aspect=DataOrientedAspect;
610       target=ConnectSpecimenArchivingWithSpecimenGathering;
611       cardinality= enum one;
612     }
613
614     concept ConnectSpecimenGatheringWithSpecimenArchivingSubProcess
          extends SubProcessRelation{
615       role="ConnectSpecimenGatheringWithSpecimenArchiving is a
              subprocess in a composite process";
616       identifier=
              ConnectSpecimenGatheringWithSpecimenArchivingSubProcessIdentifier
              ;
617       aspect=DataOrientedAspect;
618       target=ConnectSpecimenGatheringWithSpecimenArchiving;
619       cardinality= enum one;
620     }
621
622     concept ConnectSpecimenRecordWithDNAAnalysisSubProcess extends
          SubProcessRelation{
623       role="ConnectSpecimenRecordWithDNAAnalysisSubProcess is a
              subprocess in a composite process";
624       identifier=
              ConnectSpecimenRecordWithDNAAnalysisSubProcessIdentifier;
625       aspect=DataOrientedAspect;
626       target=ConnectSpecimenRecordWithDNAAnalysis;
627       cardinality= enum one;
628     }
629
630     /*
631      * ProcessExecutionDocuments
632     */
633
634
635     //Monitoring
636
637     concept BaseMonitoring extends BaseProcessExecutionDocument{
638       name="BaseMonitoring";
639       identifier=BaseMonitoringIdentifier;
640       object=ObservationObjectRelation;
```

```
641       relation= ResponsibleScientistRelation, ExecutionTimeRelation,
              ExecutionLocalityRelation, WorkOfReferenceRelation,
642       RelatedProcessDocumentsRelation,MethodRelation;
643     }
644
645
646     concept Identification extends BaseProcessExecutionDocument{ //DwC
647       name="Identification";
648       identifier=IdentificationIdentifier;
649       object=BiologicalObjectRelation;
650       relation=ResponsibleScientistRelation,TaxonDeterminationRelation,
651       ExecutionTimeRelation,WorkOfReferenceRelation;
652     }
653
654     concept CoExistenceObservation extends BaseMonitoring{// UC3:
          Connect Observations PED4,PED5
655       name="CoExistenceObservation";
656       identifier=CoExistenceObservationIdentifier;
657       relation=FormOfCoExistenceRelation;
658     }
659
660     //Specimen
661     concept SpecimenGathering extends BaseMonitoring{ //UC2: Beleg in
          digitaler Form
662       name="SpecimenGathering";
663       identifier=SpecimenGatheringIdentifier;
664       object=SpecimenRelation;
665       relation=ResponsibleGathererRelation,SpecimenRecordRelation;
666     }
667
668     concept SpecimenArchiving extends BaseProcessExecutionDocument{ //
          UC2: PED3
669       name="SpecimenArchiving";
670       identifier=SpecimenArchivingIdentifier;
671       object=SpecimenRelation;
672       relation=ExecutionTimeRelation,ExecutionLocalityRelation,
              ResponsibleCuratorRelation,SpecimenCollectionRelation,
673       SpecimenRecordRelation;
674     }
675
676     concept SpecimenPreparating extends BaseProcessExecutionDocument{//
          DwC
677       name="SpecimenPreparating";
678       identifier=SpecimenPreparatingIdentifier;
679       object=SpecimenRelation;
```

```
680       relation=ExecutionTimeRelation,ExecutionLocalityRelation,
              ResponsibleScientistRelation,MethodRelation,
681       SpecimenRecordRelation;
682     }
683
684     concept AddSpecimen extends BaseProcessExecutionDocument{//UC4:
            PED7
685       name="AddSpecimen";
686       identifier=AddSpecimenIdentifier;
687       object=SpecimenRelation;
688       relation=SpecimenCollectionRelation, ResponsibleCuratorRelation,
              ExecutionTimeRelation, ExternalInstituteRelation,
689       SpecimenRecordRelation;
690     }
691
692     concept RemoveSpecimen extends BaseProcessExecutionDocument{// UC4:
            PED7
693       name="RemoveSpecimen";
694       identifier=RemoveSpecimenIdentifier;
695       object=SpecimenRelation;
696       relation=SpecimenCollectionRelation, ResponsibleCuratorRelation,
              ExecutionTimeRelation, ExternalInstituteRelation,
697       SpecimenRecordRelation;
698     }
699
700     //Multimedia
701
702     concept MultimediaRecording extends BaseMonitoring{// UC3: PED6
703       name="MultimediaRecording";
704       identifier=MultimediaRecordingIdentifier;
705       relation=UsedDeviceRelation,MultimediaDocumentRelation;// UC3:
              Foto
706     }
707
708     concept MultimediaStoragePlaceUpdate extends
            BaseProcessExecutionDocument{// UC3: PED6
709       name="MultimediaStoragePlaceUpdate";
710       identifier=MultimediaStoragePlaceUpdateIdentifier;
711       relation=ExecutionLocalityRelation,ExecutionTimeRelation,
              ResponsibleITSystemRelation;
712       object=MultimediaDocumentRelation;// UC3: Foto
713     }
714
715
716     //Measuring
```

```
717
718     concept BaseMeasuring extends BaseProcessExecutionDocument{//UC5
719       name="BaseMeasuring";
720       identifier=BaseMeasuringIdentifier;
721       object=ObservationObjectRelation;
722       relation=ExecutionTimeRelation,ExecutionLocalityRelation,
              ResponsibleScientistRelation,MeasurementMethodRelation,
723       WorkOfReferenceRelation,MeasurementValueRelation;
724     }
725
726     concept NumericMeasuring extends BaseMeasuring{//UC5
727       name="LengthMeasuring";
728       identifier=NumericMeasuringIdentifier;
729       object=ObservationObjectRelation;
730       relation=NumericMeasurementValueRelation;
731     }
732
733
734     concept Tagging extends BaseMeasuring{//DwC
735       name="Tagging";
736       identifier=TaggingIdentifier;
737       relation=TagAssignmentRelation;
738     }
739
740     concept Describing extends BaseMeasuring{//DwC
741       name="Describing";
742       identifier=DescribingIdentifier;
743       relation=ObjectDescriptionRelation;
744     }
745
746     concept GeoReferencing extends BaseMeasuring{// UC1
747       name="GeoReferencing";
748       identifier=GeoReferencingIdentifier;
749       relation=CoordinateDeterminationRelation,UsedDeviceRelation,
              MethodRelation;
750     }
751
752
753     concept SurfaceMeasuring extends BaseMeasuring{//DwC
754       name="SurfaceMeasuring";
755       identifier=SurfaceMeasuringIdentifier;
756       object=LocalityObjectRelation;
757       relation=LocalitySurfaceDescriptionRelation;
758     }
759
```

```
760     concept DNAAnalysing extends BaseMeasuring{// UC5
761       name="DNAAnalysing";
762       identifier=DNAAnalysingIdentifier;
763       object=SpecimenRelation;
764       relation=ExternalInstituteRelation,
              ResponsibleLaboratoryTechnicianRelation, DNAAnalysisRelation;
765     }
766
767     //Data Connecting
768
769
770     concept ConnectSpecimenArchivingWithSpecimenGathering extends
            BaseProcessExecutionDocument{ //UC2: PED2+PED3
771       name="ConnectSpecimenArchivingWithSpecimenGathering";
772       identifier=
              ConnectSpecimenArchivingWithSpecimenGatheringIdentifier;
773       object=SpecimenArchivingRelation;
774       relation=ExecutionTimeRelation,ExecutionLocalityRelation,
              ResponsibleITSystemRelation,
775       SpecimenGatheringRelation;
776     }
777
778     concept ConnectSpecimenGatheringWithSpecimenArchiving extends
            BaseProcessExecutionDocument{ //UC2: PED2+PED3
779       name="ConnectSpecimenGatheringWithSpecimenArchiving";
780       identifier=
              ConnectSpecimenGatheringWithSpecimenArchivingIdentifier;
781       object=SpecimenGatheringRelation;
782       relation=ExecutionTimeRelation,ExecutionLocalityRelation,
              ResponsibleITSystemRelation,
783       SpecimenArchivingRelation;
784     }
785
786     concept ConnectObservationWithMultimediaRecording extends
            BaseProcessExecutionDocument{//UC3: (PED4,PED5)+PED6
787       name="ConnectObservationWithMultimediaRecording";
788       identifier=ConnectObservationWithMultimediaRecordingIdentifier;
789       object=ObservationRelation;
790       relation=ExecutionTimeRelation,ExecutionLocalityRelation,
              ResponsibleRelation,
791       MultimediaDocumentRelation;
792     }
793
794     concept ConnectSpecimenGatheringWithMultimediaRecording extends
            BaseProcessExecutionDocument{ //UC3: (PED4,PED5)+PED6
```

```
795       name="ConnectSpecimenGatheringWithMultimediaRecording";
796       identifier=
              ConnectSpecimenGatheringWithMultimediaRecordingIdentifier;
797       object=SpecimenGatheringRelation;
798       relation=ExecutionTimeRelation,ExecutionLocalityRelation,
              ResponsibleRelation,
799       MultimediaDocumentRelation;
800     }
801
802     concept ConnectSpecimenRecordWithDNAAnalysis extends
            BaseProcessExecutionDocument{ //UC4: PED7
803       name="ConnectSpecimenRecorfWithDNAAnalysis";
804       identifier=ConnectSpecimenRecordWithDNAAnalysisIdentifier;
805       object=SpecimenRecordRelation;
806       relation=ExecutionTimeRelation,ExecutionLocalityRelation,
              ResponsibleRelation,
807       DNAAnalysisRelation;
808     }
809
810     //Composite Processes
811     concept SiteInspection extends BaseProcessExecutionDocument{// DwC=
          Complete Monitoring of a site
812       name="SiteInspection";
813       identifier=SiteInspectionIdentifier;
814       object=SiteRelation;
815       relation= BaseMonitoringMultipleSubProcess,
              BaseMeasuringMultipleSubProcess,
816       TimeSpanRelation,ExecutionTimeRelation,ExecutionHabitatRelation,
817       ResponsibleScientistRelation,WrittenDocumentationRelation,
              SamplingMethodRelation;
818     }
819
820     concept EcologicalMeasuring extends BaseProcessExecutionDocument{
          //=UC5
821       name="EcologicalMeasuring";
822       identifier=EcologicalMeasuringIdentifier;
823       object=SiteRelation;
824       relation=BaseMeasuringMultipleSubProcess,
              CoExistenceObservationMultipleSubProcess;
825     }
826
827     concept Observation extends BaseMonitoring{//=UC1: PED1; UC3:PED4,
          PED5
828       name="Observation";
829       identifier=ObservationIdentifier;
```

```
830       object=BiologicalObjectRelation;
831       relation=IdentificationSubProcess,GeoReferencingSubProcess;
832     }
833
834     concept SpecimenMonitoring extends BaseMonitoring{//=UC2: PED2
835       name="SpecimenMonitoring";
836       identifier=SpecimenMonitoringIdentifier;
837       object=SpecimenRelation;
838       relation=SpecimenGatheringSubProcess,ObservationSubProcess;
839     }
840
841     concept SpecimenGatheringWithObservation extends BaseMonitoring {//
            =UC2
842       name="SpecimenGatheringWithObservation";
843       identifier=SpecimenGatheringWithObservationIdentifier;
844       object=SpecimenRelation;
845       relation=SpecimenMonitoringSubProcess,SpecimenArchivingSubProcess
              ,
846       ConnectSpecimenArchivingWithSpecimenGatheringSubProcess,
847       ConnectSpecimenGatheringWithSpecimenArchivingSubProcess;
848     }
849
850     concept MultimediaObservation extends BaseMonitoring{
851       name="MultimediaObservation";
852       identifier=MultimediaObservationIdentifier;
853       relation=ObservationSubProcess,MultimediaRecordingSubProcess,
              MultimediaUpdateSubProcess;
854     }
855
856     concept MultimediaCoExistenceObservation extends BaseMonitoring{//=
            UC3
857       name="MultimediaObservation";
858       identifier=MultimediaCoExistenceObservationIdentifier;
859       relation=CoExistenceObservationSubProcess,
              MultimediaRecordingSubProcess,MultimediaUpdateSubProcess,
              MultimediaObservationSubProcess;
860     }
861
862
863     concept SpecimenDNAAnalysing extends BaseProcessExecutionDocument{
            //=UC4
864       name="SpecimenDNAAnalysing";
865       identifier=SpecimenDNAAnalysingIdentifier;
866       object=SpecimenRelation;
```

```
867       relation=RemoveSpecimenSubProcess,DNAAnalysingSubProcess,
              AddSpecimenSubProcess,
868       ConnectSpecimenRecordWithDNAAnalysisSubProcess;
869     }
870
871
872
873     /*Definition of EntityTypes
874      * EntityTypes are used to descripe Aspect and are compatible with
             ER
875      * Includen in Reference-Inheritence System
876      */
877
878     //DataSet
879
880     concept BiodivDataSet extends BaseDataSet{
881       name="BiodivDataSet";
882       identifier=BiodivDataSetIdentifier;
883       attribute=language,entityName,informationWithHeld;
884       relation=DataSetCollectionRelation,DataSetLicenseRelation,
              AccessRightsRelation,
885       WorkOfReferenceRelation,ExternalDocumentRelation;
886     }
887
888     //Objects
889
890     concept ObservationObject extends FunctionalBaseEntity{
891       name="ObservationObject";
892       identifier=ObservationObjectIdentifier;
893       relation=ObservationObjectPartWholeRelation;
894     }
895
896     concept BiologicalObject extends ObservationObject{
897       name="BiologicalObject";
898       identifier=BiologicalObjectIdentifier;
899     }
900
901     concept ExtendedBiologicalObject extends BiologicalObject{
902       name="ExtendedBiologicalObject";
903       identifier=ExtendedBiologicalObjectIdentifier;
904       attribute=typeStatus,establishedMeans;
905     }
906
907     concept LocalityObject extends ObservationObject,LocalBaseEntity{
908       name="LocalityObject";
```

```
909       identifier=LocalityObjectIdentifier;
910     }
911
912     //Specimen
913
914     concept Specimen extends BiologicalObject{
915       name="Specimen";
916       identifier=SpecimenIdentifier;
917       relation=SpecimenRecordRelation;
918     }
919
920     concept SpecimenRecord extends DataOrientedBaseEntity{
921       name="SpecimenRecord";
922       identifier=SpecimenRecordIdentifier;
923       attribute=fieldNumber,specimenInCollectionCode,disposition;
924     }
925
926     //Taxon
927
928     concept BaseTaxon extends FunctionalBaseEntity{
929       name="BaseTaxon";
930       identifier=BaseTaxonIdentifier;
931       attribute=scientificName,scientificNameID,nomenclaturalCode,
              taxonRemarks;
932     }
933
934     concept Taxon extends BaseTaxon{
935       name="Taxon";
936       identifier=TaxonIdentifier;
937       attribute=vernacularName,taxonomicStatus;
938     }
939
940     concept TaxonLink extends BaseTaxon{
941       name="TaxonLink";
942       identifier=TaxonLinkIdentifier;
943       attribute=nomenclaturalStatus;
944       relation=ParentNameTaxonRelation,OriginalNameTaxonRelation,
              AcceptedNameTaxonRelation;
945     }
946
947     concept TaxonRanks extends BaseTaxon{
948       name="TaxonRanks";
949       identifier=TaxonRanksIdentifier;
950       attribute=kingdom,phylum,class,order,family,genus,subgenus,
              specificEpithet,
```

```
951       taxonRank, verbatimTaxonRank, scientificNameAuthorship;
952     }
953
954     concept TaxonPublication extends BaseTaxon{
955       name="TaxonPublication";
956       identifier=TaxonPublicationIdentifier;
957       relation=WorkOfReferenceRelation;
958     }
959
960     concept ExtendedTaxon extends Taxon, TaxonRanks{
961       name="ExtendedTaxon";
962       identifier=ExtendedTaxonIdentifier;
963     }
964
965     concept FullTaxon extends ExtendedTaxon, TaxonPublication, TaxonLink{
966       name="FullTaxon";
967       identifier=FullTaxonIdentifier;
968     }
969
970     //Measurement
971
972
973
974     concept BaseMeasurement extends FunctionalBaseEntity{
975       name="BaseMeasurement";
976       identifier=BaseMeasurementIdentifier;
977       attribute=measurementType, remarks;
978       relation=MeasurementMethodRelation;
979     }
980
981
982     concept NumericMeasurement extends BaseMeasurement{
983       name="NumericMeasurement";
984       identifier=NumericMeasurementIdentifier;
985       attribute=numericMeasurementValue, measurementUnit,
              measurementAccuracy;
986
987     }
988
989     concept Tag extends BaseMeasurement{
990       name="Tag";
991       identifier=TagIdentifier;
992       attribute=tagValue;
993     }
994
```

```
995     concept Description extends BaseMeasurement{
996       name="Description";
997       identifier=DescriptionIdentifier;
998       attribute=description;
999     }
1000
1001    concept ExtendedDescription extends Description{
1002      name="Description";
1003      identifier=ExtendedDescriptionIdentifier;
1004      attribute=verbatimDescription;
1005    }
1006
1007    abstract concept LocalityMeasurement extends BaseMeasurement{
1008      name="LocalityMeasurement";
1009      identifier=LocalityMeasurementIdentifier;
1010    }
1011
1012    abstract concept Coordinates extends LocalityMeasurement{
1013      name="Coordinates";
1014      identifier=CoordinatesIdentifier;
1015    }
1016
1017    concept VerbatimCoordinates extends Coordinates{
1018      name="VerbatimCoordinates";
1019      identifier=VerbatimCoordinatesIdentifier;
1020      attribute=verbatimLatitude,verbatimLongitude,coordinateSystem,
              spatialReferenceSystem;
1021    }
1022
1023    concept DecimalCoordinates extends Coordinates{
1024      name="DecimalCoordinates";
1025      identifier=DecimalCoordinatesIdentifier;
1026      attribute=longitude,latitude,coordinateSystem,
              spatialReferenceSystem;
1027    }
1028
1029    abstract concept CoordinatesWithPrecision extends Coordinates{
1030      name="CoordinatesWithPrecision";
1031      identifier=CoordinatesWithPrecisionIdentifier;
1032      attribute=coordinatePrecision, uncertainty, pointRadiusSpatialFit
              ;
1033    }
1034
1035    concept DecimalCoordinatesWithPrecision extends DecimalCoordinates,
          CoordinatesWithPrecision{
```

```
1036        name="DecimalCoordinatesWithPrecision";
1037        identifier=DecimalCoordinatesWithPrecisionIdentifier;
1038      }
1039
1040      concept MTBCoordinates extends Coordinates{
1041        name="Coordinates with TK25 = Messtischblättern (MTB)";
1042        identifier=MTBCoordinatesIdentifier;
1043        attribute=rightValue, heightValue;
1044        relation=TKCoordinatesRelation;
1045      }
1046
1047      concept WKTArea extends Coordinates{
1048        name="WKTArea";
1049        identifier=WKTAreaIdentifier;
1050        attribute=wKT, spatialReferenceSystem , pointRadiusSpatialFit;
1051      }
1052
1053
1054      concept SurfaceDescription extends BaseMeasurement{
1055        name="SurfaceDescription";
1056        identifier=SurfaceDescriptionIdentifier;
1057      }
1058
1059      concept VerbatimSurfaceDescription extends SurfaceDescription{
1060        name="VerbatimSurfaceDescription";
1061        identifier=VerbatimSurfaceDescriptionIdentifier;
1062        attribute=description;
1063      }
1064
1065      concept UnitSurfaceDescription extends SurfaceDescription{
1066        name="UnitSurfaceDescription";
1067        identifier=UnitSurfaceDescriptionIdentifier;
1068        attribute=minValue, maxValue, measurementUnit;
1069      }
1070
1071      //Local Entities
1072
1073      concept Site extends LocalBaseEntity{
1074        name="Site";
1075        identifier=SiteIdentifier;
1076        attribute=entityName, description;
1077      }
1078
1079      concept LocalityDescription extends LocalBaseEntity, Description{
1080        name="LocalityDescription";
```

```
1081        identifier=LocalityDescriptionIdentifier;
1082        attribute=continent;
1083      }
1084
1085      concept MarineLocality extends LocalityDescription{
1086        name="MarineLocality";
1087        identifier=MarineLocalityIdentifier;
1088        attribute=waterBody;
1089      }
1090
1091      concept IslandLocality extends LocalityDescription{
1092        name="IslandLocality";
1093        identifier=IslandLocalityIdentifier;
1094        attribute=island,islandGroup;
1095      }
1096
1097      concept PoliticalLocality extends LocalityDescription{
1098        name="PoliticalLocality";
1099        identifier=PoliticalLocalityIdentifier;
1100        attribute=country,countrycode,stateProvince,county,municipality;
1101      }
1102
1103
1104      concept ParentGeography extends LocalityDescription{
1105        name="ParentGeography";
1106        identifier=ParentGeographyIdentifier;
1107        relation=ParentGeographyRelation;
1108      }
1109
1110      concept FullGeography extends MarineLocality, IslandLocality,
              PoliticalLocality,ParentGeography{
1111        name="FullGeography";
1112        identifier=FullGeographyIdentifier;
1113      }
1114
1115
1116      //OperationalEntities
1117
1118
1119      concept BaseMethod extends OperationalBaseEntity{
1120        name="BaseMethod";
1121        identifier=BaseMethodIdentifier;
1122        attribute=description,remarks;
1123      }
1124
```

```
1125    concept SamplingMethod extends BaseMethod{
1126      name="SamplingtMethod";
1127      identifier=SamplingMethodIdentifier;
1128      attribute=samplingProtocol,samplingEffort;
1129    }
1130
1131    //Organizational Entities
1132
1133    concept Collection extends OrganizationalBaseEntity{
1134      name="Collection";
1135      identifier=CollectionIdentifier;
1136      relation=InstituteCollectionRelation,DataSetCollectionRelation;
1137    }
1138
1139    concept License extends OrganizationalBaseEntity{
1140      name="License";
1141      identifier=LicenseIdentifier;
1142    }
1143
1144    concept Scientist extends Person{
1145      name="Scientist";
1146      identifier=ScientistIdentifier;
1147      relation=InstitutePersonRelation;
1148    }
1149
1150    concept LaboratoryTechnician extends Person{
1151      name="Scientist";
1152      identifier=LaboratoryTechnicianIdentifier;
1153      relation=InstitutePersonRelation;
1154    }
1155
1156    concept Curator extends Scientist{
1157      name="Curator";
1158      identifier=CuratorIdentifier;
1159    }
1160
1161    concept Gatherer extends Scientist{
1162      name="Gatherer";
1163      identifier=GathererIdentifier;
1164    }
1165
1166    concept ITSystem extends OrganizationalBaseEntity{
1167      name="ITSystem";
1168      identifier=ITSystemIdentifier;
1169      attribute=entityName;
```

```
1170        relation=InstituteITSystemRelation;
1171      }
1172
1173      //Behavioural Entities
1174
1175      //Operational Entities
1176
1177      concept Camera extends Device{
1178        name="Camera";
1179        identifier=CameraIdentifier;
1180      }
1181
1182      concept AudioRecorder extends Device{
1183        name="AudioRecorder";
1184        identifier=AudioRecorderIdentifier;
1185      }
1186
1187      concept GPSRecorder extends Device{
1188        name="GPSRecorder";
1189        identifier=GPSRecorderIdentifier;
1190      }
1191
1192      concept Map extends OperationalBaseEntity{
1193        name="Map";
1194        identifier=MapIdentifier;
1195        attribute=entityName;
1196      }
1197
1198      concept TK extends Map{
1199        name="Topographical Map";
1200        identifier=TKIdentifier;
1201        attribute=entityName,entityCode,resolution;
1202      }
1203
1204      //Temporal Entities
1205
1206      //DataOriented Entities
1207
1208      concept WorkOfReference extends ExternalDocument{
1209        name="WorkOfReference";
1210        identifier=WorkOfReferenceIdentifier;
1211        attribute=yearOfPublication;
1212        relation=AuthorRelation;
1213      }
1214
```

```
1215    concept MultimediaDocument extends DigitalDocument{
1216      name="MultimediaDocument";
1217      identifier=MultimediaDocumentIdentifier;
1218    }
1219
1220    concept WrittenDocumentation extends WrittenDocument{
1221      name="MultimediaDocument";
1222      identifier=WrittenDocumentationIdentifier;
1223      attribute=fieldNumber;
1224    }
1225
1226    concept Picture extends MultimediaDocument{
1227      name="Picture";
1228      identifier=PictureIdentifier;
1229      attribute=mimeType;
1230    }
1231
1232    concept Audio extends MultimediaDocument{
1233      name="Audio";
1234      identifier=AudioIdentifier;
1235    }
1236
1237    concept Video extends MultimediaDocument{
1238      name="Video";
1239      identifier=VideoIdentifier;
1240    }
1241
1242    concept DNAAnalysis extends ExternalDocument{
1243      name="DNAAnalysis";
1244      identifier=DNAAnalysisIdentifier;
1245    }
1246
1247    //Controlled Vocabulary
1248
1249    enum nomenclaturalCodeVocabulary{ //Reference:  http://rs.tdwg.org/
          dwc/terms/index.htm#nomenclaturalCode
1250      BioCode,
1251      ICBN,
1252      ICNB,
1253      ICNCP,
1254      ICZN,
1255      ICVCN
1256    }
1257
```

```
1258    enum taxonRankVocabulary{ //Reference: http://rs.tdwg.org/dwc/terms
            /index.htm#taxonRank
1259      kingdom,
1260      subkingdom,
1261      phylum,
1262      subphylum,
1263      class,
1264      subclass,
1265      order,
1266      suborder,
1267      family,
1268      subfamily,
1269      tribe,
1270      subtribe,
1271      genus,
1272      subgenus,
1273      section,
1274      subsection,
1275      series,
1276      subseries,
1277      species,
1278      subspecies,
1279      varitey,
1280      subvariety,
1281      form,
1282      subform
1283    }
1284
1285    enum taxonomicStatusVocabulary{ //Reference: http://rs.tdwg.org/dwc
            /terms/index.htm#taxonomicStatus
1286      accepted,
1287      valid,
1288      synonym,
1289      homotypicSynonym,
1290      heterotypicSynonym,
1291      propParteSynonym,
1292      missapplied
1293    }
1294
1295    enum taxonLinkVocabulary{
1296      parent,
1297      original,
1298      accepted
1299    }
1300
```

```
1301    enum sexVocabulary{ //Reference: http://rs.tdwg.org/dwc/terms/index
            .htm#sex
1302      unknowable,
1303      undetermined,
1304      female,
1305      male,
1306      hermaphrodite
1307    }
1308
1309
1310    enum lifeStageVocabulary{ //Reference: http://rs.tdwg.org/dwc/terms
            /index.htm#lifeStage
1311      zygote,
1312      embryo,
1313      larva,
1314      juvenile,
1315      adult,
1316      sporophyte,
1317      spore,
1318      gametophyte,
1319      gamete,
1320      pupa
1321    }
1322
1323    enum establishedMeansVocabulary{ //Reference: http://rs.tdwg.org/
            dwc/terms/index.htm#establishmentMeans
1324      native,
1325      introduced,
1326      naturalised,
1327      invasive,
1328      managed,
1329      uncertain
1330    }
1331
1332    enum occurenceStatusVocabulary{ //Reference: http://rs.tdwg.org/dwc
            /terms/index.htm#occurrenceStatus
1333      present,
1334      absent,
1335      common,
1336      irregular,
1337      rare,
1338      doubtful
1339    }
1340
```

```
1341    enum typeStatusVocabulary{//Reference: http://rs.tdwg.org/dwc/terms
            /index.htm#typeStatus
1342      holotype,
1343      paratype,
1344      neotype,
1345      syntype,
1346      lectotype,
1347      paralectotype,
1348      hapantotype
1349    }
1350
1351    enum surfaceRelation{
1352      elevation,
1353      depth,
1354      distanceAboveSurface
1355    }
1356
1357    enum formOfCoexistenceVocabulary{ //Reference: DiversityCollection:
            CollUnitRelationType_Enum
1358      association,
1359      childOf,
1360      parentOf,
1361      endophyticIn,
1362      foundOn,
1363      growingOn,
1364      isolatedFrom,
1365      lichenization,
1366      mutantOf,
1367      mutualism,
1368      mycorrhizaOf,
1369      parasiticOn,
1370      pollinatorOf,
1371      predatorOf,
1372      foodOf,
1373      saprophyticOn,
1374      sibling
1375    }
1376
1377
1378    /*
1379     * Attributes
1380     */
1381
1382
1383    //BiodivDataSet Attributes
```

```
1384
1385      Attribute informationWithHeld{
1386        string value;
1387        type="string";
1388        identifier=informationWithHeldIdentifier;
1389        isNullable=true;
1390      }
1391
1392      //BiologicalObject Attributes
1393      Attribute typeStatus{
1394        string value;
1395        type="typeStatusVocabulary";
1396        identifier=typeStatusIdentifier;
1397        isNullable=true;
1398      }
1399
1400      Attribute tagValue{
1401        string value;
1402        type="tagValue";
1403        identifier=tagValueIdentifier;
1404        isNullable=true;
1405      }
1406
1407      Attribute establishedMeans extends tagValue{
1408        string value;
1409        type="establishedMeansVocabulary";
1410        identifier=establishedMeansIdentifier;
1411        isNullable=true;
1412      }
1413
1414      Attribute sex extends tagValue{
1415        string value;
1416        type="sexVocabulary";
1417        identifier=sexIdentifier;
1418        isNullable=true;
1419      }
1420
1421      Attribute lifeStage extends tagValue{
1422        string value;
1423        type="lifeStageVocabulary";
1424        identifier=lifeStageIdentifier;
1425        isNullable=true;
1426      }
1427
1428      Attribute reproductiveCondition extends tagValue{
```

```
1429        string value;
1430        type="reproductiveConditionVocabulary";
1431        identifier=reproductiveConditionIdentifier;
1432        isNullable=true;
1433      }
1434
1435      Attribute behavior extends tagValue{
1436        string value;
1437        type="behaviorVocabulary";
1438        identifier=behaviorIdentifier;
1439        isNullable=true;
1440      }
1441
1442      Attribute occurrenceStatus extends tagValue{
1443        string value;
1444        type="occurrenceStatusVocabulary";
1445        identifier=occurrenceStatusIdentifier;
1446        isNullable=true;
1447      }
1448
1449      Attribute formOfCoexistence{
1450        string value;
1451        type="formOfCoexistenceVocabulary";
1452        identifier=formOfCoexistenceIdentifier;
1453        isNullable=true;
1454      }
1455
1456      Attribute verificationStatus{
1457        string value;
1458        type="verificationStatus";
1459        identifier=verificationStatusIdentifier;
1460        isNullable=true;
1461      }
1462
1463      Attribute securityOfDetermination{
1464        string value;
1465        type="string";
1466        identifier=securityOfDeterminationIdentifier;
1467      }
1468
1469      //Specimen Attributes
1470
1471      Attribute fieldNumber{
1472        string value;
1473        type="string";
```

```
1474        identifier=fieldNumberIdentifier;
1475      }
1476
1477      Attribute specimenInCollectionCode{
1478        string value;
1479        type="string";
1480        identifier=specimenInCollectionCodeIdentifier;
1481        isNullable=true;
1482      }
1483
1484      Attribute disposition{
1485        string value;
1486        type="string";
1487        identifier=dispositionIdentifier;
1488      }
1489
1490      //ModalityAttributes
1491      Attribute typeOfOwnership{
1492        string value;
1493        type="string";
1494        identifier=typeOfOwnershipIdentifier;
1495        isNullable=true;
1496      }
1497
1498      Attribute typeOfAffiliation{
1499        string value;
1500        type="string";
1501        identifier=typeOfAffiliationIdentifier;
1502        isNullable=true;
1503      }
1504
1505      Attribute typeOfSurface{
1506        string value;
1507        type="description of surface relation";
1508        identifier=typeOfSurfaceIdentifier;
1509        isNullable=true;
1510      }
1511
1512
1513      //Locality Attributes
1514
1515      Attribute continent{
1516        string value;
1517        type="string";
1518        identifier=continentIdentifier;
```

```
1519        isNullable=true;
1520      }
1521
1522      Attribute waterBody{
1523        string value;
1524        type="string";
1525        identifier=waterBodyIdentifier;
1526        isNullable=true;
1527      }
1528
1529      Attribute island{
1530        string value;
1531        type="string";
1532        identifier=islandIdentifier;
1533        isNullable=true;
1534      }
1535
1536      Attribute islandGroup{
1537        string value;
1538        type="string";
1539        identifier=islandGroupIdentifier;
1540        isNullable=true;
1541      }
1542
1543      Attribute country{
1544        string value;
1545        type="string";
1546        identifier=countryIdentifier;
1547        isNullable=true;
1548      }
1549
1550      Attribute countrycode{
1551        string value;
1552        type="string";
1553        identifier=countryCodeIdentifier;
1554        isNullable=true;
1555      }
1556
1557      Attribute stateProvince{
1558        string value;
1559        type="string";
1560        identifier=stateProvinceIdentifier;
1561        isNullable=true;
1562      }
1563
```

```
1564      Attribute county{
1565        string value;
1566        type="string";
1567        identifier=countyIdentifier;
1568        isNullable=true;
1569      }
1570
1571      Attribute municipality{
1572        string value;
1573        type="string";
1574        identifier=municipalityIdentifier;
1575        isNullable=true;
1576      }
1577
1578      Attribute latitude{
1579        real value;
1580        type="real";
1581        identifier=latitudeIdentifier;
1582      }
1583
1584      Attribute longitude{
1585        real value;
1586        type="real";
1587        identifier=longitudeIdentifier;
1588      }
1589
1590      Attribute altitude{
1591        real value;
1592        type="real";
1593        identifier=altitudeIdentifier;
1594        isNullable=true;
1595      }
1596
1597      Attribute verbatimLatitude{
1598        string value;
1599        type="string";
1600        identifier=verbatimLatitudeIdentifier;
1601      }
1602
1603      Attribute verbatimLongitude{
1604        string value;
1605        type="string";
1606        identifier=verbatimLongitudeIdentifier;
1607      }
1608
```

```
1609      Attribute spatialReferenceSystem{
1610        string value;
1611        type="string";
1612        identifier=spatialReferenceSystemIdentifier;
1613      }
1614
1615      Attribute coordinatePrecision{
1616        string value;
1617        type="real";
1618        identifier=coordinatePrecisionIdentifier;
1619        isNullable=true;
1620      }
1621
1622      Attribute uncertainty{
1623        string value;
1624        type="real";
1625        identifier=uncertaintyIdentifier;
1626        isNullable=true;
1627      }
1628
1629      Attribute coordinateSystem{
1630        string value;
1631        type="string";
1632        identifier=coordinateSystemIdentifier;
1633      }
1634
1635      Attribute wKT{
1636        string value;
1637        type="string";
1638        identifier=wKTIdentifier;
1639      }
1640
1641
1642      Attribute pointRadiusSpatialFit{
1643        string value;
1644        type="string";
1645        identifier=pointRadiusSpatialFitIdentifier;
1646        isNullable=true;
1647      }
1648
1649      Attribute minValue{
1650        string value;
1651        type="real";
1652        identifier=minValueIdentifier;
1653        isNullable=true;
```

```
1654      }
1655
1656      Attribute maxValue{
1657        string value;
1658        type="real";
1659        identifier=maxValueIdentifier;
1660        isNullable=true;
1661      }
1662
1663      //MeasurementAttributes
1664
1665
1666      Attribute numericMeasurementValue{
1667        string value;
1668        type="real";
1669        identifier=numericMeasurementValueIdentifier;
1670      }
1671
1672      Attribute measurementType{
1673        string value;
1674        type="string";
1675        identifier=measurementTypeIdentifier;
1676      }
1677
1678      Attribute measurementUnit{
1679        string value;
1680        type="string";
1681        identifier=measurementUnitIdentifier;
1682      }
1683
1684      Attribute measurementAccuracy{
1685        string value;
1686        type="string";
1687        identifier=measurementAccuracyIdentifier;
1688        isNullable=true;
1689      }
1690
1691      Attribute description{
1692        string value;
1693        type="string";
1694        identifier=descriptionIdentifier;
1695      }
1696
1697      Attribute verbatimDescription{
1698        string value;
```

```
1699        type="string";
1700        identifier=verbatimDescriptionIdentifier;
1701      }
1702
1703      //Sampling Attributes
1704
1705      Attribute samplingProtocol{
1706        string value;
1707        type="string";
1708        identifier=samplingProtocolIdentifier;
1709        isNullable=true;
1710      }
1711
1712      Attribute samplingEffort{
1713        string value;
1714        type="string";
1715        identifier=samplingEffortIdentifier;
1716        isNullable=true;
1717      }
1718
1719      //Taxon Attributes
1720
1721
1722      Attribute scientificNameID{
1723        string value;
1724        type="string";
1725        identifier=scientificNameIDIdentifier;
1726      }
1727
1728      Attribute acceptedNameUsageID{
1729        string value;
1730        type="string";
1731        identifier=acceptedNameUsageIDIdentifier;
1732      }
1733
1734
1735      Attribute parentNameUsageID{
1736        string value;
1737        type="string";
1738        identifier=parentNameUsageIDIdentifier;
1739        isNullable=true;
1740      }
1741
1742      Attribute originalNameUsageID{
1743        string value;
```

```
1744        type="string";
1745        identifier=originalNameUsageIDIdentifier;
1746        isNullable=true;
1747      }
1748
1749      Attribute nameAccordingToID{
1750        string value;
1751        type="string";
1752        identifier=nameAccordingToIDIdentifier;
1753        isNullable=true;
1754      }
1755
1756      Attribute namePublishedInID{
1757        string value;
1758        type="string";
1759        identifier=namePublishedInIDIdentifier;
1760      }
1761
1762      Attribute taxonConceptID{
1763        string value;
1764        type="string";
1765        identifier=taxonConceptIDIdentifier;
1766      }
1767
1768      Attribute scientificName{
1769        string value;
1770        type="string";
1771        identifier=scientificNameIdentifier;
1772      }
1773
1774      Attribute acceptedNameUsage{
1775        string value;
1776        type="string";
1777        identifier=acceptedNameUsageIdentifier;
1778      }
1779
1780      Attribute parentNameUsage{
1781        string value;
1782        type="string";
1783        identifier=parentNameUsageIdentifier;
1784        isNullable=true;
1785      }
1786
1787      Attribute originalNameUsage{
1788        string value;
```

```
1789        type="string";
1790        identifier=originalNameUsageIdentifier;
1791        isNullable=true;
1792      }
1793
1794      Attribute higherClassification{
1795        string value;
1796        type="string";
1797        identifier=higherClassificationIdentifier;
1798      }
1799
1800      Attribute kingdom{
1801        string value;
1802        type="string";
1803        identifier=kingdomIdentifier;
1804      }
1805
1806      Attribute phylum{
1807        string value;
1808        type="string";
1809        identifier=phylumIdentifier;
1810      }
1811
1812      Attribute class{
1813        string value;
1814        type="string";
1815        identifier=classIdentifier;
1816      }
1817
1818      Attribute order{
1819        string value;
1820        type="string";
1821        identifier=orderIdentifier;
1822      }
1823
1824      Attribute family{
1825        string value;
1826        type="string";
1827        identifier=familyIdentifier;
1828        isNullable=true;
1829      }
1830
1831
1832      Attribute genus{
1833        string value;
```

```
1834        type="string";
1835        identifier=genusIdentifier;
1836        isNullable=true;
1837      }
1838
1839      Attribute subgenus{
1840        string value;
1841        type="string";
1842        identifier=subgenusIdentifier;
1843        isNullable=true;
1844      }
1845
1846      Attribute specificEpithet{
1847        string value;
1848        type="string";
1849        identifier=specificEpithetIdentifier;
1850        isNullable=true;
1851      }
1852
1853      Attribute taxonRank{
1854        string value;
1855        type="taxonRankVocabulary";
1856        identifier=taxonRankIdentifier;
1857      }
1858
1859      Attribute verbatimTaxonRank{
1860        string value;
1861        type="string";
1862        identifier=verbatimTaxonRankIdentifier;
1863        isNullable=true;
1864      }
1865
1866      Attribute scientificNameAuthorship{
1867        string value;
1868        type="string";
1869        identifier=scientificNameAuthorshipIdentifier;
1870        isNullable=true;
1871      }
1872
1873
1874      Attribute vernacularName{
1875        string value;
1876        type="string";
1877        identifier=vernacularNameIdentifier;
1878        isNullable=true;
```

```
1879      }
1880
1881      Attribute nomenclaturalCode{
1882        string value;
1883        type="nomenclaturalCodeVocabulary";
1884        identifier=nomenclaturalCodeIdentifier;
1885        isNullable=true;
1886      }
1887
1888      Attribute taxonomicStatus{
1889        string value;
1890        type="taxonomicStatusVocaubulary";
1891        identifier=taxonomicStatusIdentifier;
1892        isNullable=true;
1893      }
1894
1895      Attribute nomenclaturalStatus{
1896        string value;
1897        type="string";
1898        identifier=nomenclaturalStatusIdentifier;
1899        isNullable=true;
1900      }
1901
1902      Attribute taxonRemarks{
1903        string value;
1904        type="string";
1905        identifier=taxonRemarksIdentifier;
1906        isNullable=true;
1907      }
1908
1909      Attribute taxonLink{
1910        string value;
1911        type="taxonLinkVocabulary";
1912        identifier=taxonLinkIdentifier;
1913        isNullable=true;
1914      }
1915
1916      //Operational
1917
1918      Attribute rightValue{
1919        string value;
1920        type="real";
1921        identifier=rightValueIdentifier;
1922      }
1923
```

```
1924      Attribute heightValue{
1925        string value;
1926        type="real";
1927        identifier=heightValueIdentifier;
1928      }
1929
1930      Attribute resolution{
1931        string value;
1932        type="string";
1933        identifier=resolutionIdentifier;
1934      }
1935
1936      //DataOriented
1937
1938      Attribute yearOfPublication{
1939        integer value;
1940        type="integer";
1941        identifier=yearOfPublicationIdentifier;
1942      }
1943
1944      Attribute mimeType{
1945        string value;
1946        type="string";
1947        identifier=mimeTypeIdentifier;
1948      }
1949
1950
1951
1952      /*
1953       * Identifier
1954       */
1955
1956      //ModalityIdentifier
1957
1958      ConceptIdentifier SecurityOfDeterminationModalityIdentifier{
1959        name="PODSLBiodiv/SecurityOfDeterminationModality";
1960        repository="www.ai4.uni-bayreuth.de";
1961        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              SecurityOfDeterminationModality";
1962      }
1963
1964      ConceptIdentifier VerificationOfDeterminationModalityIdentifier{
1965        name="PODSLBiodiv/erificationOfDeterminationModality";
1966        repository="www.ai4.uni-bayreuth.de";
```

```
1967        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                erificationOfDeterminationModality";
1968      }
1969
1970      ConceptIdentifier FormOfCoexistenceModalityIdentifier{
1971        name="PODSLBiodiv/FormOfCoexistenceModality";
1972        repository="www.ai4.uni-bayreuth.de";
1973        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                FormOfCoexistenceModality";
1974      }
1975
1976      ConceptIdentifier OwnershipModalityIdentifier{
1977        name="PODSLBiodiv/OwnershipModality";
1978        repository="www.ai4.uni-bayreuth.de";
1979        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                OwnershipModality";
1980      }
1981
1982      ConceptIdentifier AffiliationModalityIdentifier{
1983        name="PODSLBiodiv/AffiliationModality";
1984        repository="www.ai4.uni-bayreuth.de";
1985        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                AffiliationModality";
1986      }
1987
1988      ConceptIdentifier LocalitySurfaceDescriptionModalityIdentifier{
1989        name="PODSLBiodiv/LocalitySurfaceDescriptionModality";
1990        repository="www.ai4.uni-bayreuth.de";
1991        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                LocalitySurfaceDescriptionModality";
1992      }
1993
1994
1995
1996      //RelationIdentifier
1997
1998      ConceptIdentifier ObjectObservationRelationIdentifier{
1999        name="PODSLBiodiv/ObjectObservationRelation";
2000        repository="www.ai4.uni-bayreuth.de";
2001        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                ObjectObservationRelation";
2002      }
2003
2004      ConceptIdentifier BiologicalObjectRelationIdentifier{
2005        name="PODSLBiodiv/BiologicalObjectRelation";
```

```
2006        repository="www.ai4.uni-bayreuth.de";
2007        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                BiologicalObjectRelation";
2008      }
2009
2010      ConceptIdentifier LocalityObjectRelationIdentifier{
2011        name="PODSLBiodiv/LocalityObjectRelation";
2012        repository="www.ai4.uni-bayreuth.de";
2013        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                LocalityObjectRelation";
2014      }
2015
2016      ConceptIdentifier SpecimenRelationIdentifier{
2017        name="PODSLBiodiv/SpecimenRelation";
2018        repository="www.ai4.uni-bayreuth.de";
2019        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                SpecimenRelation";
2020      }
2021
2022      ConceptIdentifier SiteRelationIdentifier{
2023        name="PODSLBiodiv/SiteRelation";
2024        repository="www.ai4.uni-bayreuth.de";
2025        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/SiteRelation"
                ;
2026      }
2027
2028      ConceptIdentifier AcceptedNameTaxonRelationIdentifier{
2029        name="PODSLBiodiv/AcceptedNameTaxonRelation";
2030        repository="www.ai4.uni-bayreuth.de";
2031        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                AcceptedNameTaxonRelation";
2032      }
2033
2034      ConceptIdentifier ParentNameTaxonRelationIdentifier{
2035        name="PODSLBiodiv/ParentNameTaxonRelation";
2036        repository="www.ai4.uni-bayreuth.de";
2037        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                ParentNameTaxonRelation";
2038      }
2039
2040      ConceptIdentifier OriginalNameTaxonRelationIdentifier{
2041        name="PODSLBiodiv/OriginalNameTaxonRelation";
2042        repository="www.ai4.uni-bayreuth.de";
2043        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                OriginalNameTaxonRelation";
```

```
2044      }
2045
2046      ConceptIdentifier InstituteCollectionRelationIdentifier{
2047        name="PODSLBiodiv/InstituteCollectionRelation";
2048        repository="www.ai4.uni-bayreuth.de";
2049        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              InstituteCollectionRelation";
2050      }
2051
2052      ConceptIdentifier SpecimenCollectionRelationIdentifier{
2053        name="PODSLBiodiv/SpecimenCollectionRelation";
2054        repository="www.ai4.uni-bayreuth.de";
2055        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              SpecimenCollectionRelation";
2056      }
2057
2058      ConceptIdentifier InstitutePersonRelationIdentifier{
2059        name="PODSLBiodiv/InstitutePersonRelation";
2060        repository="www.ai4.uni-bayreuth.de";
2061        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              InstitutePersonRelation";
2062      }
2063
2064      ConceptIdentifier InstituteITSystemRelationIdentifier{
2065        name="PODSLBiodiv/InstituteITSystemRelation";
2066        repository="www.ai4.uni-bayreuth.de";
2067        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              InstituteITSystemRelation";
2068      }
2069
2070      ConceptIdentifier ResponsibleScientistRelationIdentifier{
2071        name="PODSLBiodiv/ResponsibleScientistRelation";
2072        repository="www.ai4.uni-bayreuth.de";
2073        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              ResponsibleScientistRelation";
2074      }
2075
2076      ConceptIdentifier ResponsibleLaboratoryTechnicianRelationIdentifier
            {
2077        name="PODSLBiodiv/ResponsibleLaboratoryTechnicianRelation";
2078        repository="www.ai4.uni-bayreuth.de";
2079        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              ResponsibleLaboratoryTechnicianRelation";
2080      }
2081
```

```
2082    ConceptIdentifier ResponsibleGathererRelationIdentifier{
2083      name="PODSLBiodiv/ResponsibleGathererRelation";
2084      repository="www.ai4.uni-bayreuth.de";
2085      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              ResponsibleGathererRelation";
2086    }
2087
2088    ConceptIdentifier ResponsibleCuratorRelationIdentifier{
2089      name="PODSLBiodiv/ResponsibleCuratorRelation";
2090      repository="www.ai4.uni-bayreuth.de";
2091      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              ResponsibleCuratorRelation";
2092    }
2093
2094    ConceptIdentifier ResponsibleITSystemRelationIdentifier{
2095      name="PODSLBiodiv/ResponsibleITSystemRelation";
2096      repository="www.ai4.uni-bayreuth.de";
2097      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              ResponsibleITSystemRelation";
2098    }
2099
2100
2101    ConceptIdentifier AuthorRelationIdentifier{
2102      name="PODSLBiodiv/AuthorRelation";
2103      repository="www.ai4.uni-bayreuth.de";
2104      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              AuthorRelation";
2105    }
2106
2107    ConceptIdentifier DataSetCollectionRelationIdentifier{
2108      name="PODSLBiodiv/DataSetCollectionRelation";
2109      repository="www.ai4.uni-bayreuth.de";
2110      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              DataSetCollectionRelation";
2111    }
2112
2113    ConceptIdentifier DataSetLicenseRelationIdentifier{
2114      name="PODSLBiodiv/DataSetLicenseRelation";
2115      repository="www.ai4.uni-bayreuth.de";
2116      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              DataSetLicenseRelation";
2117    }
2118
2119    ConceptIdentifier AccessRightsRelationIdentifier{
2120      name="PODSLBiodiv/AccessRightsRelation";
```

```
2121        repository="www.ai4.uni-bayreuth.de";
2122        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                AccessRightsRelation";
2123      }
2124
2125
2126      ConceptIdentifier TaxonDeterminationRelationIdentifier{
2127        name="PODSLBiodiv/TaxonDeterminationRelation";
2128        repository="www.ai4.uni-bayreuth.de";
2129        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                TaxonDeterminationRelation";
2130      }
2131
2132      ConceptIdentifier FormOfCoexistenceRelationIdentifier{
2133        name="PODSLBiodiv/FormOfCoexistenceRelation";
2134        repository="www.ai4.uni-bayreuth.de";
2135        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                FormOfCoexistenceRelation";
2136      }
2137
2138      ConceptIdentifier MeasurementValueRelationIdentifier{
2139        name="PODSLBiodiv/MeasurementValueRelation";
2140        repository="www.ai4.uni-bayreuth.de";
2141        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                MeasurementValueRelation";
2142      }
2143
2144
2145      ConceptIdentifier LocalitySurfaceDescriptionRelationIdentifier{
2146        name="PODSLBiodiv/LocalitySurfaceDescriptionRelation";
2147        repository="www.ai4.uni-bayreuth.de";
2148        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                LocalitySurfaceDescriptionRelation";
2149      }
2150
2151
2152      ConceptIdentifier TagAssignmentRelationIdentifier{
2153        name="PODSLBiodiv/TagAssignmentRelation";
2154        repository="www.ai4.uni-bayreuth.de";
2155        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                TagAssignmentRelation";
2156      }
2157
2158      ConceptIdentifier ObjectDescriptionRelationIdentifier{
2159        name="PODSLBiodiv/ObjectDescriptionRelation";
```

```
2160        repository="www.ai4.uni-bayreuth.de";
2161        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                ObjectDescriptionRelation";
2162      }
2163
2164      ConceptIdentifier CoordinateDeterminationRelationIdentifier{
2165        name="PODSLBiodiv/CoordinateDeterminationRelation";
2166        repository="www.ai4.uni-bayreuth.de";
2167        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                CoordinateDeterminationRelation";
2168      }
2169
2170      ConceptIdentifier MeasurementMethodRelationIdentifier{
2171        name="PODSLBiodiv/MeasurementMethodRelation";
2172        repository="www.ai4.uni-bayreuth.de";
2173        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                MeasurementMethodRelation";
2174      }
2175
2176      ConceptIdentifier ExecutionHabitatRelationIdentifier{
2177        name="PODSLBiodiv/ExecutionHabitatRelation";
2178        repository="www.ai4.uni-bayreuth.de";
2179        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                ExecutionHabitatRelation";
2180      }
2181
2182      ConceptIdentifier ExternalInstituteRelationIdentifier{
2183        name="PODSLBiodiv/ExternalInstituteRelation";
2184        repository="www.ai4.uni-bayreuth.de";
2185        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                ExternalInstituteRelation";
2186      }
2187
2188      ConceptIdentifier ParentGeographyRelationIdentifier{
2189        name="PODSLBiodiv/ParentGeographyRelationIdentifier";
2190        repository="www.ai4.uni-bayreuth.de";
2191        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                ParentGeographyRelationIdentifier";
2192      }
2193
2194      ConceptIdentifier WorkOfReferenceRelationIdentifier{
2195        name="PODSLBiodiv/WorkOfReferenceRelation";
2196        repository="www.ai4.uni-bayreuth.de";
2197        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                WorkOfReferenceRelation";
```

```
2198      }
2199
2200      ConceptIdentifier MultimediaDocumentRelationIdentifier{
2201        name="PODSLBiodiv/MultimediaDocumentRelation";
2202        repository="www.ai4.uni-bayreuth.de";
2203        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              MultimediaDocumentRelation";
2204      }
2205
2206      ConceptIdentifier WrittenDocumentationRelationIdentifier{
2207        name="PODSLBiodiv/WrittenDocumentationRelation";
2208        repository="www.ai4.uni-bayreuth.de";
2209        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              WrittenDocumentationRelation";
2210      }
2211
2212      ConceptIdentifier DNAAnalysisRelationIdentifier{
2213        name="PODSLBiodiv/DNAAnalysisRelation";
2214        repository="www.ai4.uni-bayreuth.de";
2215        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              DNAAnalysisRelation";
2216      }
2217
2218      ConceptIdentifier RelatedProcessDocumentsRelationIdentifier{
2219        name="PODSLBiodiv/RelatedProcessDocumentsRelation";
2220        repository="www.ai4.uni-bayreuth.de";
2221        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              RelatedProcessDocumentsRelation";
2222      }
2223
2224      ConceptIdentifier SpecimenRecordRelationIdentifier{
2225        name="PODSLBiodiv/SpecimenRecordRelation";
2226        repository="www.ai4.uni-bayreuth.de";
2227        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              SpecimenRecordRelation";
2228      }
2229
2230      ConceptIdentifier ObservationRelationIdentifier{
2231        name="PODSLBiodiv/ObservationRelation";
2232        repository="www.ai4.uni-bayreuth.de";
2233        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              ObservationRelation";
2234      }
2235
2236      ConceptIdentifier SpecimenGatheringRelationIdentifier{
```

```
2237        name="PODSLBiodiv/SpecimenGatheringRelation";
2238        repository="www.ai4.uni-bayreuth.de";
2239        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                SpecimenGatheringRelation";
2240      }
2241
2242      ConceptIdentifier SpecimenArchivingRelationIdentifier{
2243        name="PODSLBiodiv/SpecimenArchivingRelation";
2244        repository="www.ai4.uni-bayreuth.de";
2245        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                SpecimenArchivingRelation";
2246      }
2247
2248      ConceptIdentifier MethodRelationIdentifier{
2249        name="PODSLBiodiv/MethodRelation";
2250        repository="www.ai4.uni-bayreuth.de";
2251        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                MethodRelation";
2252      }
2253
2254      ConceptIdentifier SamplingMethodRelationIdentifier{
2255        name="PODSLBiodiv/SamplingMethodRelation";
2256        repository="www.ai4.uni-bayreuth.de";
2257        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                SamplingMethodRelation";
2258      }
2259
2260      ConceptIdentifier PreparatingRelationIdentifier{
2261        name="PODSLBiodiv/PreparatingRelation";
2262        repository="www.ai4.uni-bayreuth.de";
2263        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                PreparatingRelation";
2264      }
2265
2266      ConceptIdentifier NumericMeasurementValueRelationIdentifier{
2267        name="PODSLBiodiv/NumericMeasurementValueRelation";
2268        repository="www.ai4.uni-bayreuth.de";
2269        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                NumericMeasurementValueRelation";
2270      }
2271
2272
2273      ConceptIdentifier UsedDeviceRelationIdentifier{
2274        name="PODSLBiodiv/UsedDeviceRelation";
2275        repository="www.ai4.uni-bayreuth.de";
```

```
2276        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                UsedDeviceRelation";
2277      }
2278
2279
2280      ConceptIdentifier ObservationObjectPartWholeRelationIdentifier{
2281        name="PODSLBiodiv/ObservationObjectPartWholeRelation";
2282        repository="www.ai4.uni-bayreuth.de";
2283        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                ObservationObjectPartWholeRelation";
2284      }
2285
2286      ConceptIdentifier MapRelationIdentifier{
2287        name="PODSLBiodiv/MapRelation";
2288        repository="www.ai4.uni-bayreuth.de";
2289        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/MapRelation";
2290      }
2291
2292      ConceptIdentifier MapCoordinatesRelationIdentifier{
2293        name="PODSLBiodiv/MapCoordinatesRelation";
2294        repository="www.ai4.uni-bayreuth.de";
2295        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                MapCoordinatesRelation";
2296      }
2297
2298      ConceptIdentifier TKCoordinatesRelationIdentifier{
2299        name="PODSLBiodiv/TKCoordinatesRelation";
2300        repository="www.ai4.uni-bayreuth.de";
2301        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                TKCoordinatesRelation";
2302      }
2303
2304      //SubProcesses
2305
2306      ConceptIdentifier IdentificationSubProcessIdentifier{
2307        name="PODSLBiodiv/IdentificationSubProcess";
2308        repository="www.ai4.uni-bayreuth.de";
2309        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                IdentificationSubProcess";
2310      }
2311
2312      ConceptIdentifier GeoReferencingSubProcessIdentifier{
2313        name="PODSLBiodiv/GeoReferencingSubProcess";
2314        repository="www.ai4.uni-bayreuth.de";
```

```
2315        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                GeoReferencingSubProcess";
2316      }
2317
2318      ConceptIdentifier SpecimenGatheringSubProcessIdentifier{
2319        name="PODSLBiodiv/SpecimenGatheringSubProcesss";
2320        repository="www.ai4.uni-bayreuth.de";
2321        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                SpecimenGatheringSubProcess";
2322      }
2323
2324      ConceptIdentifier ObservationSubProcessIdentifier{
2325        name="PODSLBiodiv/ObservationSubProcess";
2326        repository="www.ai4.uni-bayreuth.de";
2327        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                ObservationSubProcess";
2328      }
2329
2330      ConceptIdentifier CoExistenceObservationSubProcessIdentifier{
2331        name="PODSLBiodiv/CoExistenceObservationSubProcess";
2332        repository="www.ai4.uni-bayreuth.de";
2333        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                CoExistenceObservationSubProcess";
2334      }
2335
2336      ConceptIdentifier SpecimenMonitoringSubProcessIdentifier{
2337        name="PODSLBiodiv/SpecimenMonitoringSubProcess";
2338        repository="www.ai4.uni-bayreuth.de";
2339        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                SpecimenMonitoringSubProcess";
2340      }
2341
2342      ConceptIdentifier SpecimenArchivingSubProcessIdentifier{
2343        name="PODSLBiodiv/SpecimenArchivingSubProcess";
2344        repository="www.ai4.uni-bayreuth.de";
2345        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                SpecimenArchivingSubProcess";
2346      }
2347
2348      ConceptIdentifier MultimediaRecordingSubProcessIdentifier{
2349        name="PODSLBiodiv/MultimediaRecordingSubProcessSubProcess";
2350        repository="www.ai4.uni-bayreuth.de";
2351        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                MultimediaRecordingSubProcess";
2352      }
```

```
2353
2354      ConceptIdentifier MultimediaUpdateSubProcessIdentifier{
2355        name="PODSLBiodiv/MultimediaUpdateSubProcess";
2356        repository="www.ai4.uni-bayreuth.de";
2357        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                MultimediaUpdateSubProcess";
2358      }
2359
2360      ConceptIdentifier MultimediaObservationSubProcessIdentifier{
2361        name="PODSLBiodiv/MultimediaObservationSubProcess";
2362        repository="www.ai4.uni-bayreuth.de";
2363        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                MultimediaObservationSubProcess";
2364      }
2365
2366      ConceptIdentifier RemoveSpecimenSubProcessIdentifier{
2367        name="PODSLBiodiv/RemoveSpecimenSubProcess";
2368        repository="www.ai4.uni-bayreuth.de";
2369        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                RemoveSpecimenSubProcess";
2370      }
2371
2372      ConceptIdentifier DNAAnalysingSubProcessIdentifier{
2373        name="PODSLBiodiv/DNAAnalysingSubProcess";
2374        repository="www.ai4.uni-bayreuth.de";
2375        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                DNAAnalysingSubProcess";
2376      }
2377
2378      ConceptIdentifier AddSpecimenSubProcessIdentifier{
2379        name="PODSLBiodiv/AddSpecimenSubProcess";
2380        repository="www.ai4.uni-bayreuth.de";
2381        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                AddSpecimenSubProcesss";
2382      }
2383
2384      ConceptIdentifier BaseMeasuringSubProcessIdentifier{
2385        name="PODSLBiodiv/BaseMeasuringSubProcess";
2386        repository="www.ai4.uni-bayreuth.de";
2387        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                BaseMeasuringSubProcess";
2388      }
2389
2390      //MultipleExecution
2391
```

```
2392    ConceptIdentifier BaseMonitoringMultipleSubProcessIdentifier{
2393      name="PODSLBiodiv/BaseMonitoringMultipleSubProcess";
2394      repository="www.ai4.uni-bayreuth.de";
2395      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              BaseMonitoringMultipleSubProcess";
2396    }
2397
2398    ConceptIdentifier BaseMeasuringMultipleSubProcessIdentifier{
2399      name="PODSLBiodiv/BaseMeasuringMultipleSubProcess";
2400      repository="www.ai4.uni-bayreuth.de";
2401      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              BaseMeasuringMultipleSubProcess";
2402    }
2403
2404    ConceptIdentifier
          CoExistenceObservationMultipleSubProcessIdentifier{
2405      name="PODSLBiodiv/CoExistenceObservationMultipleSubProcess";
2406      repository="www.ai4.uni-bayreuth.de";
2407      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              CoExistenceObservationMultipleSubProcess";
2408    }
2409
2410    //Subprocesses for data connections
2411
2412    ConceptIdentifier
          ConnectSpecimenArchivingWithSpecimenGatheringSubProcessIdentifier
          {
2413      name="PODSLBiodiv/
              ConnectSpecimenArchivingWithSpecimenGatheringSubProcess";
2414      repository="www.ai4.uni-bayreuth.de";
2415      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              ConnectSpecimenArchivingWithSpecimenGatheringSubProcess";
2416    }
2417
2418    ConceptIdentifier
          ConnectSpecimenGatheringWithSpecimenArchivingSubProcessIdentifier
          {
2419      name="PODSLBiodiv/
              ConnectSpecimenGatheringWithSpecimenArchivingSubProcess";
2420      repository="www.ai4.uni-bayreuth.de";
2421      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              ConnectSpecimenGatheringWithSpecimenArchivingSubProcess";
2422    }
2423
```

```
2424    ConceptIdentifier
            ConnectSpecimenRecordWithDNAAnalysisSubProcessIdentifier{
2425      name="PODSLBiodiv/ConnectSpecimenRecordWithDNAAnalysisSubProcess"
              ;
2426      repository="www.ai4.uni-bayreuth.de";
2427      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              ConnectSpecimenRecordWithDNAAnalysisSubProcess";
2428    }
2429
2430    //ProcessExecutionDocuments
2431
2432    ConceptIdentifier IdentificationIdentifier{
2433      name="PODSLBiodiv/Identification";
2434      repository="www.ai4.uni-bayreuth.de";
2435      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              Identification";
2436    }
2437
2438    ConceptIdentifier BaseMonitoringIdentifier{
2439      name="PODSLBiodiv/BaseMonitoring";
2440      repository="www.ai4.uni-bayreuth.de";
2441      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              BaseMonitoring";
2442    }
2443
2444    ConceptIdentifier ObservationIdentifier{
2445      name="PODSLBiodiv/Observation";
2446      repository="www.ai4.uni-bayreuth.de";
2447      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/Observation";
2448    }
2449
2450    ConceptIdentifier CoExistenceObservationIdentifier{
2451      name="PODSLBiodiv/CoExistenceObservation";
2452      repository="www.ai4.uni-bayreuth.de";
2453      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              CoExistenceObservation";
2454    }
2455
2456    //Specimen
2457
2458    ConceptIdentifier SpecimenGatheringIdentifier{
2459      name="PODSLBiodiv/SpecimenGathering";
2460      repository="www.ai4.uni-bayreuth.de";
2461      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              SpecimenGathering";
```

```
2462      }
2463
2464      ConceptIdentifier SpecimenArchivingIdentifier{
2465        name="PODSLBiodiv/SpecimenArchiving";
2466        repository="www.ai4.uni-bayreuth.de";
2467        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              SpecimenArchiving";
2468      }
2469
2470      ConceptIdentifier SpecimenPreparatingIdentifier{
2471        name="PODSLBiodiv/SpecimenPreparating";
2472        repository="www.ai4.uni-bayreuth.de";
2473        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              SpecimenPreparating";
2474      }
2475
2476
2477      ConceptIdentifier AddSpecimenIdentifier{
2478        name="PODSLBiodiv/AddSpecimen";
2479        repository="www.ai4.uni-bayreuth.de";
2480        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/AddSpecimen";
2481      }
2482
2483      ConceptIdentifier RemoveSpecimenIdentifier{
2484        name="PODSLBiodiv/RemoveSpecimen";
2485        repository="www.ai4.uni-bayreuth.de";
2486        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              RemoveSpecimen";
2487      }
2488
2489      //Multimedia
2490
2491      ConceptIdentifier MultimediaRecordingIdentifier{
2492        name="PODSLBiodiv/MultimediaRecording";
2493        repository="www.ai4.uni-bayreuth.de";
2494        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              MultimediaRecording";
2495      }
2496
2497      ConceptIdentifier MultimediaStoragePlaceUpdateIdentifier{
2498        name="PODSLBiodiv/MultimediaStoragePlaceUpdate";
2499        repository="www.ai4.uni-bayreuth.de";
2500        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              MultimediaStoragePlaceUpdate";
2501      }
```

```
2502
2503      //Measuring
2504
2505      ConceptIdentifier BaseMeasuringIdentifier{
2506        name="PODSLBiodiv/BaseMeasuring";
2507        repository="www.ai4.uni-bayreuth.de";
2508        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/BaseMeasuring
              ";
2509      }
2510      ConceptIdentifier NumericMeasuringIdentifier{
2511        name="PODSLBiodiv/NumericMeasuring";
2512        repository="www.ai4.uni-bayreuth.de";
2513        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              NumericMeasuring";
2514      }
2515
2516      ConceptIdentifier TaggingIdentifier{
2517        name="PODSLBiodiv/Tagging";
2518        repository="www.ai4.uni-bayreuth.de";
2519        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/Tagging";
2520      }
2521
2522      ConceptIdentifier DescribingIdentifier{
2523        name="PODSLBiodiv/Describing";
2524        repository="www.ai4.uni-bayreuth.de";
2525        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/Describing";
2526      }
2527
2528      ConceptIdentifier GeoReferencingIdentifier{
2529        name="PODSLBiodiv/GeoReferencing";
2530        repository="www.ai4.uni-bayreuth.de";
2531        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              GeoReferencing";
2532      }
2533
2534      ConceptIdentifier SurfaceMeasuringIdentifier{
2535        name="PODSLBiodiv/SurfaceMeasuring";
2536        repository="www.ai4.uni-bayreuth.de";
2537        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              SurfaceMeasuring";
2538      }
2539
2540      ConceptIdentifier DNAAnalysingIdentifier{
2541        name="PODSLBiodiv/DNAAnalysing";
2542        repository="www.ai4.uni-bayreuth.de";
```

```
2543        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/DNAAnalysing"
                ;
2544    }
2545
2546    //Data Connecting
2547
2548    ConceptIdentifier
            ConnectSpecimenArchivingWithSpecimenGatheringIdentifier{
2549        name="PODSLBiodiv/ConnectSpecimenArchivingWithSpecimenGathering";
2550        repository="www.ai4.uni-bayreuth.de";
2551        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                ConnectSpecimenArchivingWithSpecimenGathering";
2552    }
2553
2554    ConceptIdentifier
            ConnectSpecimenGatheringWithSpecimenArchivingIdentifier{
2555        name="PODSLBiodiv/ConnectSpecimenGatheringWithSpecimenArchiving";
2556        repository="www.ai4.uni-bayreuth.de";
2557        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                ConnectSpecimenGatheringWithSpecimenArchiving";
2558    }
2559
2560    ConceptIdentifier
            ConnectObservationWithMultimediaRecordingIdentifier{
2561        name="PODSLBiodiv/ConnectObservationWithMultimediaRecording";
2562        repository="www.ai4.uni-bayreuth.de";
2563        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                ConnectObservationWithMultimediaRecording";
2564    }
2565
2566    ConceptIdentifier
            ConnectSpecimenGatheringWithMultimediaRecordingIdentifier{
2567        name="PODSLBiodiv/ConnectSpecimenGatheringWithMultimediaRecording
                ";
2568        repository="www.ai4.uni-bayreuth.de";
2569        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                ConnectSpecimenGatheringWithMultimediaRecording";
2570    }
2571
2572    ConceptIdentifier ConnectSpecimenRecordWithDNAAnalysisIdentifier{
2573        name="PODSLBiodiv/ConnectSpecimenRecordWithDNAAnalysis";
2574        repository="www.ai4.uni-bayreuth.de";
2575        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                ConnectSpecimenRecordWithDNAAnalysis";
2576    }
```

```
2577
2578      //Composite
2579
2580      ConceptIdentifier SiteInspectionIdentifier{
2581        name="PODSLBiodiv/SiteInspection";
2582        repository="www.ai4.uni-bayreuth.de";
2583        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              SiteInspection";
2584      }
2585
2586      ConceptIdentifier EcologicalMeasuringIdentifier{
2587        name="PODSLBiodiv/EcologicalMeasuring";
2588        repository="www.ai4.uni-bayreuth.de";
2589        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              EcologicalMeasuring";
2590      }
2591
2592      ConceptIdentifier SpecimenDNAAnalysingIdentifier{
2593        name="PODSLBiodiv/SpecimenDNAAnalysing";
2594        repository="www.ai4.uni-bayreuth.de";
2595        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              SpecimenDNAAnalysing";
2596      }
2597
2598      ConceptIdentifier SpecimenMonitoringIdentifier{
2599        name="PODSLBiodiv/SpecimenMonitoring";
2600        repository="www.ai4.uni-bayreuth.de";
2601        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              SpecimenMonitoring";
2602      }
2603
2604      ConceptIdentifier SpecimenGatheringWithObservationIdentifier{
2605        name="PODSLBiodiv/SpecimenGatheringWithObservation";
2606        repository="www.ai4.uni-bayreuth.de";
2607        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              SpecimenGatheringWithObservation";
2608      }
2609
2610      ConceptIdentifier MultimediaObservationIdentifier{
2611        name="PODSLBiodiv/MultimediaObservation";
2612        repository="www.ai4.uni-bayreuth.de";
2613        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              MultimediaObservation";
2614      }
2615
```

```
2616      ConceptIdentifier MultimediaCoExistenceObservationIdentifier{
2617        name="PODSLBiodiv/MultimediaCoExistenceObservation";
2618        repository="www.ai4.uni-bayreuth.de";
2619        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              MultimediaCoExistenceObservation";
2620      }
2621
2622      //EntityIdentifier
2623
2624      //DataSet
2625
2626      ConceptIdentifier BiodivDataSetIdentifier{
2627        name="PODSLBiodiv/BiodivDataSet";
2628        repository="www.ai4.uni-bayreuth.de";
2629        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/BiodivDataSet
              ";
2630      }
2631
2632      //Objects
2633
2634      ConceptIdentifier ObservationObjectIdentifier{
2635        name="PODSLBiodiv/ObservationObject";
2636        repository="www.ai4.uni-bayreuth.de";
2637        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              ObservationObject";
2638      }
2639
2640      ConceptIdentifier BiologicalObjectIdentifier{
2641        name="PODSLBiodiv/BiologicalObject";
2642        repository="www.ai4.uni-bayreuth.de";
2643        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              BiologicalObject";
2644      }
2645
2646      ConceptIdentifier ExtendedBiologicalObjectIdentifier{
2647        name="PODSLBiodiv/ExtendedBiologicalObject";
2648        repository="www.ai4.uni-bayreuth.de";
2649        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              ExtendedBiologicalObject";
2650      }
2651
2652      ConceptIdentifier LocalityObjectIdentifier{
2653        name="PODSLBiodiv/LocalityObject";
2654        repository="www.ai4.uni-bayreuth.de";
```

```
2655        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                LocalityObject";
2656      }
2657
2658      //Specimen
2659
2660      ConceptIdentifier SpecimenIdentifier{
2661        name="PODSLBiodiv/Specimen";
2662        repository="www.ai4.uni-bayreuth.de";
2663        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/Specimen";
2664      }
2665
2666      ConceptIdentifier SpecimenRecordIdentifier{
2667        name="PODSLBiodiv/SpecimenRecord";
2668        repository="www.ai4.uni-bayreuth.de";
2669        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                SpecimenRecord";
2670      }
2671
2672      //Taxon
2673
2674      ConceptIdentifier BaseTaxonIdentifier{
2675        name="PODSLBiodiv/BaseTaxon";
2676        repository="www.ai4.uni-bayreuth.de";
2677        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/BaseTaxon";
2678      }
2679
2680      ConceptIdentifier TaxonRanksIdentifier{
2681        name="PODSLBiodiv/TaxonRanks";
2682        repository="www.ai4.uni-bayreuth.de";
2683        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/TaxonRanks";
2684      }
2685
2686      ConceptIdentifier TaxonIdentifier{
2687        name="PODSLBiodiv/Taxon";
2688        repository="www.ai4.uni-bayreuth.de";
2689        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/Taxon";
2690      }
2691
2692      ConceptIdentifier TaxonLinkIdentifier{
2693        name="PODSLBiodiv/TaxonLink";
2694        repository="www.ai4.uni-bayreuth.de";
2695        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/TaxonLink";
2696      }
2697
```

```
2698    ConceptIdentifier ExtendedTaxonIdentifier{
2699      name="PODSLBiodiv/ExtendedTaxon";
2700      repository="www.ai4.uni-bayreuth.de";
2701      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/ExtendedTaxon
              ";
2702    }
2703
2704    ConceptIdentifier FullTaxonIdentifier{
2705      name="PODSLBiodiv/FullTaxon";
2706      repository="www.ai4.uni-bayreuth.de";
2707      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/FullTaxon";
2708    }
2709
2710    ConceptIdentifier TaxonPublicationIdentifier{
2711      name="PODSLBiodiv/TaxonPublication";
2712      repository="www.ai4.uni-bayreuth.de";
2713      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              TaxonPublication";
2714    }
2715
2716
2717
2718
2719    //MeasurementAttribute Identifier
2720
2721    ConceptIdentifier BaseMeasurementIdentifier{
2722      name="PODSLBiodiv/BaseMeasurement";
2723      repository="www.ai4.uni-bayreuth.de";
2724      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              BaseMeasurement";
2725    }
2726
2727    ConceptIdentifier MeasurementMethodIdentifier{
2728      name="PODSLBiodiv/MeasurementMethod";
2729      repository="www.ai4.uni-bayreuth.de";
2730      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              MeasurementMethod";
2731    }
2732
2733    ConceptIdentifier NumericMeasurementIdentifier{
2734      name="PODSLBiodiv/NumericMeasurement";
2735      repository="www.ai4.uni-bayreuth.de";
2736      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              NumericMeasurement";
2737    }
```

```
2738
2739      ConceptIdentifier TagIdentifier{
2740        name="PODSLBiodiv/Tag";
2741        repository="www.ai4.uni-bayreuth.de";
2742        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/Tag";
2743      }
2744
2745      ConceptIdentifier DescriptionIdentifier{
2746        name="PODSLBiodiv/Description";
2747        repository="www.ai4.uni-bayreuth.de";
2748        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/Description";
2749      }
2750
2751      ConceptIdentifier ExtendedDescriptionIdentifier{
2752        name="PODSLBiodiv/ExtendedDescription";
2753        repository="www.ai4.uni-bayreuth.de";
2754        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              ExtendedDescription";
2755      }
2756
2757      ConceptIdentifier LocalityMeasurementIdentifier{
2758        name="PODSLBiodiv/LocalityMeasurement";
2759        repository="www.ai4.uni-bayreuth.de";
2760        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              LocalityMeasurement";
2761      }
2762
2763      ConceptIdentifier CoordinatesIdentifier{
2764        name="PODSLBiodiv/Coordinates";
2765        repository="www.ai4.uni-bayreuth.de";
2766        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/Coordinates";
2767      }
2768
2769      ConceptIdentifier VerbatimCoordinatesIdentifier{
2770        name="PODSLBiodiv/VerbatimCoordinates";
2771        repository="www.ai4.uni-bayreuth.de";
2772        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              VerbatimCoordinates";
2773      }
2774
2775      ConceptIdentifier DecimalCoordinatesIdentifier{
2776        name="PODSLBiodiv/DecimalCoordinates";
2777        repository="www.ai4.uni-bayreuth.de";
2778        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              DecimalCoordinates";
```

```
2779      }
2780
2781      ConceptIdentifier CoordinatesWithPrecisionIdentifier{
2782        name="PODSLBiodiv/CoordinatesWithPrecision";
2783        repository="www.ai4.uni-bayreuth.de";
2784        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              CoordinatesWithPrecision";
2785      }
2786
2787      ConceptIdentifier DecimalCoordinatesWithPrecisionIdentifier{
2788        name="PODSLBiodiv/DecimalCoordinatesWithPrecision";
2789        repository="www.ai4.uni-bayreuth.de";
2790        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              DecimalCoordinatesWithPrecision";
2791      }
2792
2793      ConceptIdentifier MTBCoordinatesIdentifier{
2794        name="PODSLBiodiv/MTBCoordinates";
2795        repository="www.ai4.uni-bayreuth.de";
2796        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              MTBCoordinates";
2797      }
2798
2799      ConceptIdentifier WKTAreaIdentifier{
2800        name="PODSLBiodiv/WKTArea";
2801        repository="www.ai4.uni-bayreuth.de";
2802        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/WKTArea";
2803      }
2804
2805      ConceptIdentifier SurfaceDescriptionIdentifier{
2806        name="PODSLBiodiv/SurfaceDescription";
2807        repository="www.ai4.uni-bayreuth.de";
2808        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              SurfaceDescription";
2809      }
2810
2811      ConceptIdentifier VerbatimSurfaceDescriptionIdentifier{
2812        name="PODSLBiodiv/VerbatimSurfaceDescription";
2813        repository="www.ai4.uni-bayreuth.de";
2814        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              VerbatimSurfaceDescription";
2815      }
2816
2817      ConceptIdentifier UnitSurfaceDescriptionIdentifier{
2818        name="PODSLBiodiv/UnitSurfaceDescription";
```

```
2819        repository="www.ai4.uni-bayreuth.de";
2820        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                UnitSurfaceDescription";
2821      }
2822
2823      //Local
2824      ConceptIdentifier SiteIdentifier{
2825        name="PODSLBiodiv/Site";
2826        repository="www.ai4.uni-bayreuth.de";
2827        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/Site";
2828      }
2829
2830      ConceptIdentifier LocalityDescriptionIdentifier{
2831        name="PODSLBiodiv/LocalityDescription";
2832        repository="www.ai4.uni-bayreuth.de";
2833        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                LocalityDescription";
2834      }
2835
2836      ConceptIdentifier MarineLocalityIdentifier{
2837        name="PODSLBiodiv/MarineLocality";
2838        repository="www.ai4.uni-bayreuth.de";
2839        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                MarineLocality";
2840      }
2841
2842      ConceptIdentifier IslandLocalityIdentifier{
2843        name="PODSLBiodiv/IslandLocality";
2844        repository="www.ai4.uni-bayreuth.de";
2845        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                IslandLocality";
2846      }
2847
2848      ConceptIdentifier PoliticalLocalityIdentifier{
2849        name="PODSLBiodiv/PoliticalLocality";
2850        repository="www.ai4.uni-bayreuth.de";
2851        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                PoliticalLocality";
2852      }
2853
2854      ConceptIdentifier ParentGeographyIdentifier{
2855        name="PODSLBiodiv/ParentGeography";
2856        repository="www.ai4.uni-bayreuth.de";
2857        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                ParentGeography";
```

```
2858      }
2859
2860      ConceptIdentifier FullGeographyIdentifier{
2861        name="PODSLBiodiv/FullGeography";
2862        repository="www.ai4.uni-bayreuth.de";
2863        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/FullGeography
              ";
2864      }
2865
2866      //Methods
2867
2868      ConceptIdentifier BaseMethodIdentifier{
2869        name="PODSLBiodiv/BaseMethod";
2870        repository="www.ai4.uni-bayreuth.de";
2871        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/BaseMethod";
2872      }
2873
2874      ConceptIdentifier SamplingMethodIdentifier{
2875        name="PODSLBiodiv/SamplingMethod";
2876        repository="www.ai4.uni-bayreuth.de";
2877        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              SamplingMethod";
2878      }
2879
2880      //Organizational
2881
2882      ConceptIdentifier CollectionIdentifier{
2883        name="PODSLBiodiv/Collection";
2884        repository="www.ai4.uni-bayreuth.de";
2885        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/Collection";
2886      }
2887
2888      ConceptIdentifier LicenseIdentifier{
2889        name="PODSLBiodiv/License";
2890        repository="www.ai4.uni-bayreuth.de";
2891        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/License";
2892      }
2893
2894      ConceptIdentifier ScientistIdentifier{
2895        name="PODSLBiodiv/Scientist";
2896        repository="www.ai4.uni-bayreuth.de";
2897        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/Scientist";
2898      }
2899
2900      ConceptIdentifier LaboratoryTechnicianIdentifier{
```

```
2901        name="PODSLBiodiv/LaboratoryTechnician";
2902        repository="www.ai4.uni-bayreuth.de";
2903        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                LaboratoryTechnician";
2904      }
2905
2906      ConceptIdentifier CuratorIdentifier{
2907        name="PODSLBiodiv/Curator";
2908        repository="www.ai4.uni-bayreuth.de";
2909        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/Curator";
2910      }
2911
2912      ConceptIdentifier GathererIdentifier{
2913        name="PODSLBiodiv/Gatherer";
2914        repository="www.ai4.uni-bayreuth.de";
2915        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/Gatherer";
2916      }
2917
2918      ConceptIdentifier ITSystemIdentifier{
2919        name="PODSLBiodiv/ITSystem";
2920        repository="www.ai4.uni-bayreuth.de";
2921        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/ITSystem";
2922      }
2923
2924      //Operational
2925      ConceptIdentifier CameraIdentifier{
2926        name="PODSLBiodiv/Camera";
2927        repository="www.ai4.uni-bayreuth.de";
2928        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/Camera";
2929      }
2930
2931      ConceptIdentifier AudioRecorderIdentifier{
2932        name="PODSLBiodiv/AudioRecorder";
2933        repository="www.ai4.uni-bayreuth.de";
2934        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/AudioRecorder
                ";
2935      }
2936
2937      ConceptIdentifier GPSRecorderIdentifier{
2938        name="PODSLBiodiv/GPSRecorder";
2939        repository="www.ai4.uni-bayreuth.de";
2940        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/GPSRecorder";
2941      }
2942
2943      ConceptIdentifier MapIdentifier{
```

```
2944        name="PODSLBiodiv/Map";
2945        repository="www.ai4.uni-bayreuth.de";
2946        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/Map";
2947      }
2948
2949      ConceptIdentifier TKIdentifier{
2950        name="PODSLBiodiv/TK";
2951        repository="www.ai4.uni-bayreuth.de";
2952        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/TK";
2953      }
2954
2955      //DataOriented
2956
2957      ConceptIdentifier WorkOfReferenceIdentifier{
2958        name="PODSLBiodiv/WorkOfReferenceIdentifier";
2959        repository="www.ai4.uni-bayreuth.de";
2960        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              WorkOfReferenceIdentifier";
2961      }
2962
2963      ConceptIdentifier MultimediaDocumentIdentifier{
2964        name="PODSLBiodiv/MultimediaDocument";
2965        repository="www.ai4.uni-bayreuth.de";
2966        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              MultimediaDocument";
2967      }
2968
2969      ConceptIdentifier WrittenDocumentationIdentifier{
2970        name="PODSLBiodiv/WrittenDocumentation";
2971        repository="www.ai4.uni-bayreuth.de";
2972        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              WrittenDocumentation";
2973      }
2974
2975
2976      ConceptIdentifier PictureIdentifier{
2977        name="PODSLBiodiv/Picture";
2978        repository="www.ai4.uni-bayreuth.de";
2979        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/Picture";
2980      }
2981
2982      ConceptIdentifier AudioIdentifier{
2983        name="PODSLBiodiv/Audio";
2984        repository="www.ai4.uni-bayreuth.de";
2985        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/Audio";
```

```
2986      }
2987
2988      ConceptIdentifier VideoIdentifier{
2989        name="PODSLBiodiv/Video";
2990        repository="www.ai4.uni-bayreuth.de";
2991        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/Video";
2992      }
2993
2994      ConceptIdentifier DNAAnalysisIdentifier{
2995        name="PODSLBiodiv/DNAAnalysis";
2996        repository="www.ai4.uni-bayreuth.de";
2997        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/DNAAnalysis";
2998      }
2999
3000      //AttributeIdentifier
3001
3002      //BiodivDataSet
3003      ConceptIdentifier informationWithHeldIdentifier{
3004        name="PODSLBiodiv/informationWithHeld";
3005        repository="www.ai4.uni-bayreuth.de";
3006        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              informationWithHeld";
3007      }
3008
3009
3010      //BiologicalObject Attributes
3011
3012      ConceptIdentifier typeStatusIdentifier{
3013        name="PODSLBiodiv/typeStatus";
3014        repository="www.ai4.uni-bayreuth.de";
3015        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/typeStatus";
3016      }
3017
3018      ConceptIdentifier tagValueIdentifier{
3019        name="PODSLBiodiv/tagValueIdentifier";
3020        repository="www.ai4.uni-bayreuth.de";
3021        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              tagValueIdentifier";
3022      }
3023
3024      ConceptIdentifier establishedMeansIdentifier{
3025        name="PODSLBiodiv/establishedMeans";
3026        repository="www.ai4.uni-bayreuth.de";
3027        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              establishedMeans";
```

```
3028      }
3029
3030      ConceptIdentifier sexIdentifier{
3031        name="PODSLBiodiv/sex";
3032        repository="www.ai4.uni-bayreuth.de";
3033        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/sex";
3034      }
3035
3036      ConceptIdentifier lifeStageIdentifier{
3037        name="PODSLBiodiv/lifeStage";
3038        repository="www.ai4.uni-bayreuth.de";
3039        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/lifeStage";
3040      }
3041
3042      ConceptIdentifier reproductiveConditionIdentifier{
3043        name="PODSLBiodiv/reproductiveCondition";
3044        repository="www.ai4.uni-bayreuth.de";
3045        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              reproductiveCondition";
3046      }
3047
3048      ConceptIdentifier behaviorIdentifier{
3049        name="PODSLBiodiv/behavior";
3050        repository="www.ai4.uni-bayreuth.de";
3051        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/behavior";
3052      }
3053
3054      ConceptIdentifier occurrenceStatusIdentifier{
3055        name="PODSLBiodiv/occurrenceStatus";
3056        repository="www.ai4.uni-bayreuth.de";
3057        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              occurrenceStatus";
3058      }
3059
3060
3061      ConceptIdentifier formOfCoexistenceIdentifier{
3062        name="PODSLBiodiv/formOfCoexistence";
3063        repository="www.ai4.uni-bayreuth.de";
3064        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              formOfCoexistence";
3065      }
3066
3067      ConceptIdentifier verificationStatusIdentifier{
3068        name="PODSLBiodiv/verificationStatus";
3069        repository="www.ai4.uni-bayreuth.de";
```

```
3070        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                verificationStatus";
3071      }
3072
3073      ConceptIdentifier fieldNumberIdentifier{
3074        name="PODSLBiodiv/fieldNumber";
3075        repository="www.ai4.uni-bayreuth.de";
3076        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/fieldNumber";
3077      }
3078
3079      ConceptIdentifier specimenInCollectionCodeIdentifier{
3080        name="PODSLBiodiv/specimenInCollectionCode";
3081        repository="www.ai4.uni-bayreuth.de";
3082        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                specimenInCollectionCode";
3083      }
3084
3085      ConceptIdentifier dispositionIdentifier{
3086        name="PODSLBiodiv/disposition";
3087        repository="www.ai4.uni-bayreuth.de";
3088        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/disposition";
3089      }
3090
3091      //Modal
3092      ConceptIdentifier securityOfDeterminationIdentifier{
3093        name="PODSLBiodiv/securityOfDetermination";
3094        repository="www.ai4.uni-bayreuth.de";
3095        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                securityOfDetermination";
3096      }
3097
3098      ConceptIdentifier typeOfOwnershipIdentifier{
3099        name="PODSLBiodiv/typeOfOwnership";
3100        repository="www.ai4.uni-bayreuth.de";
3101        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                typeOfOwnership";
3102      }
3103
3104      ConceptIdentifier typeOfAffiliationIdentifier{
3105        name="PODSLBiodiv/typeOfAffiliation";
3106        repository="www.ai4.uni-bayreuth.de";
3107        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                typeOfAffiliation";
3108      }
3109
```

```
3110      ConceptIdentifier typeOfSurfaceIdentifier{
3111        name="PODSLBiodiv/typeOfSurface";
3112        repository="www.ai4.uni-bayreuth.de";
3113        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/typeOfSurface
              ";
3114      }
3115
3116      //Local
3117
3118      ConceptIdentifier continentIdentifier{
3119        name="PODSLBiodiv/continent";
3120        repository="www.ai4.uni-bayreuth.de";
3121        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/continent";
3122      }
3123
3124      ConceptIdentifier waterBodyIdentifier{
3125        name="PODSLBiodiv/waterBody";
3126        repository="www.ai4.uni-bayreuth.de";
3127        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/waterBody";
3128      }
3129
3130      ConceptIdentifier islandIdentifier{
3131        name="PODSLBiodiv/island";
3132        repository="www.ai4.uni-bayreuth.de";
3133        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/island";
3134      }
3135
3136      ConceptIdentifier islandGroupIdentifier{
3137        name="PODSLBiodiv/islandGroup";
3138        repository="www.ai4.uni-bayreuth.de";
3139        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/islandGroup";
3140      }
3141
3142      ConceptIdentifier countryIdentifier{
3143        name="PODSLBiodiv/country";
3144        repository="www.ai4.uni-bayreuth.de";
3145        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/country";
3146      }
3147
3148      ConceptIdentifier countryCodeIdentifier{
3149        name="PODSLBiodiv/countryCode";
3150        repository="www.ai4.uni-bayreuth.de";
3151        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/countryCode";
3152      }
3153
```

```
3154    ConceptIdentifier stateProvinceIdentifier{
3155      name="PODSLBiodiv/stateProvince";
3156      repository="www.ai4.uni-bayreuth.de";
3157      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/stateProvince
              ";
3158    }
3159
3160    ConceptIdentifier countyIdentifier{
3161      name="PODSLBiodiv/county";
3162      repository="www.ai4.uni-bayreuth.de";
3163      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/county";
3164    }
3165
3166    ConceptIdentifier municipalityIdentifier{
3167      name="PODSLBiodiv/municipality";
3168      repository="www.ai4.uni-bayreuth.de";
3169      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/municipality"
              ;
3170    }
3171
3172
3173    ConceptIdentifier latitudeIdentifier{
3174      name="PODSLBiodiv/latitude";
3175      repository="www.ai4.uni-bayreuth.de";
3176      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/latitude";
3177    }
3178
3179    ConceptIdentifier longitudeIdentifier{
3180      name="PODSLBiodiv/longitude";
3181      repository="www.ai4.uni-bayreuth.de";
3182      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/longitude";
3183    }
3184
3185    ConceptIdentifier altitudeIdentifier{
3186      name="PODSLBiodiv/Altitude";
3187      repository="www.ai4.uni-bayreuth.de";
3188      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/altitude";
3189    }
3190
3191    ConceptIdentifier verbatimLatitudeIdentifier{
3192      name="PODSLBiodiv/verbatimLatitude";
3193      repository="www.ai4.uni-bayreuth.de";
3194      address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              verbatimLatitude";
3195    }
```

```
3196
3197      ConceptIdentifier verbatimLongitudeIdentifier{
3198        name="PODSLBiodiv/verbatimLongitude";
3199        repository="www.ai4.uni-bayreuth.de";
3200        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              verbatimLongitude";
3201      }
3202
3203      ConceptIdentifier spatialReferenceSystemIdentifier{
3204        name="PODSLBiodiv/spatialReferenceSystem";
3205        repository="www.ai4.uni-bayreuth.de";
3206        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              spatialReferenceSystem";
3207      }
3208
3209      ConceptIdentifier coordinatePrecisionIdentifier{
3210        name="PODSLBiodiv/coordinatePrecision";
3211        repository="www.ai4.uni-bayreuth.de";
3212        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              coordinatePrecision";
3213      }
3214
3215      ConceptIdentifier uncertaintyIdentifier{
3216        name="PODSLBiodiv/uncertainty";
3217        repository="www.ai4.uni-bayreuth.de";
3218        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/uncertainty";
3219      }
3220
3221      ConceptIdentifier coordinateSystemIdentifier{
3222        name="PODSLBiodiv/coordinateSystem";
3223        repository="www.ai4.uni-bayreuth.de";
3224        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              coordinateSystem";
3225      }
3226
3227      ConceptIdentifier wKTIdentifier{
3228        name="PODSLBiodiv/wKT";
3229        repository="www.ai4.uni-bayreuth.de";
3230        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/wKT";
3231      }
3232
3233
3234      ConceptIdentifier pointRadiusSpatialFitIdentifier{
3235        name="PODSLBiodiv/pointRadiusSpatialFit";
3236        repository="www.ai4.uni-bayreuth.de";
```

```
3237        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                pointRadiusSpatialFit";
3238      }
3239
3240      ConceptIdentifier minValueIdentifier{
3241        name="PODSLBiodiv/minValue";
3242        repository="www.ai4.uni-bayreuth.de";
3243        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/minValue";
3244      }
3245
3246      ConceptIdentifier maxValueIdentifier{
3247        name="PODSLBiodiv/maxValue";
3248        repository="www.ai4.uni-bayreuth.de";
3249        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/maxValue";
3250      }
3251
3252
3253      //Taxon
3254      ConceptIdentifier scientificNameIDIdentifier{
3255        name="PODSLBiodiv/scientificNameID";
3256        repository="www.ai4.uni-bayreuth.de";
3257        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                scientificNameID";
3258      }
3259
3260
3261      ConceptIdentifier acceptedNameUsageIDIdentifier{
3262        name="PODSLBiodiv/acceptedNameUsageID";
3263        repository="www.ai4.uni-bayreuth.de";
3264        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                acceptedNameUsageID";
3265      }
3266
3267
3268      ConceptIdentifier parentNameUsageIDIdentifier{
3269        name="PODSLBiodiv/parentNameUsageID";
3270        repository="www.ai4.uni-bayreuth.de";
3271        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                parentNameUsageID";
3272      }
3273
3274      ConceptIdentifier originalNameUsageIDIdentifier{
3275        name="PODSLBiodiv/originalNameUsageID";
3276        repository="www.ai4.uni-bayreuth.de";
```

```
3277        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                originalNameUsageID";
3278      }
3279
3280      ConceptIdentifier nameAccordingToIDIdentifier{
3281        name="PODSLBiodiv/nameAccordingToID";
3282        repository="www.ai4.uni-bayreuth.de";
3283        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                nameAccordingToID";
3284      }
3285
3286      ConceptIdentifier namePublishedInIDIdentifier{
3287        name="PODSLBiodiv/namePublishedInID";
3288        repository="www.ai4.uni-bayreuth.de";
3289        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                namePublishedInID";
3290      }
3291
3292      ConceptIdentifier taxonConceptIDIdentifier{
3293        name="PODSLBiodiv/taxonConceptID";
3294        repository="www.ai4.uni-bayreuth.de";
3295        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                taxonConceptID";
3296      }
3297
3298      ConceptIdentifier scientificNameIdentifier{
3299        name="PODSLBiodiv/scientificName";
3300        repository="www.ai4.uni-bayreuth.de";
3301        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                scientificName";
3302      }
3303
3304      ConceptIdentifier acceptedNameUsageIdentifier{
3305        name="PODSLBiodiv/acceptedNameUsage";
3306        repository="www.ai4.uni-bayreuth.de";
3307        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                acceptedNameUsage";
3308      }
3309
3310      ConceptIdentifier parentNameUsageIdentifier{
3311        name="PODSLBiodiv/parentNameUsage";
3312        repository="www.ai4.uni-bayreuth.de";
3313        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                parentNameUsage";
3314      }
```

```
3315
3316      ConceptIdentifier originalNameUsageIdentifier{
3317        name="PODSLBiodiv/originalNameUsage";
3318        repository="www.ai4.uni-bayreuth.de";
3319        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              originalNameUsage";
3320      }
3321
3322      ConceptIdentifier nameAccordingToIdentifier{
3323        name="PODSLBiodiv/nameAccordingTo";
3324        repository="www.ai4.uni-bayreuth.de";
3325        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              nameAccordingTo";
3326      }
3327
3328      ConceptIdentifier namePublishedInIdentifier{
3329        name="PODSLBiodiv/namePublishedIn";
3330        repository="www.ai4.uni-bayreuth.de";
3331        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              namePublishedIn";
3332      }
3333
3334      ConceptIdentifier namePublishedInYearIdentifier{
3335        name="PODSLBiodiv/namePublishedInYear";
3336        repository="www.ai4.uni-bayreuth.de";
3337        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              namePublishedInYear";
3338      }
3339
3340      ConceptIdentifier higherClassificationIdentifier{
3341        name="PODSLBiodiv/higherClassification";
3342        repository="www.ai4.uni-bayreuth.de";
3343        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              higherClassification";
3344      }
3345
3346      ConceptIdentifier kingdomIdentifier{
3347        name="PODSLBiodiv/kingdomr";
3348        repository="www.ai4.uni-bayreuth.de";
3349        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/kingdom";
3350      }
3351
3352      ConceptIdentifier phylumIdentifier{
3353        name="PODSLBiodiv/phylum";
3354        repository="www.ai4.uni-bayreuth.de";
```

```
3355        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/phylum";
3356      }
3357
3358      ConceptIdentifier classIdentifier{
3359        name="PODSLBiodiv/class";
3360        repository="www.ai4.uni-bayreuth.de";
3361        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/class";
3362      }
3363
3364      ConceptIdentifier orderIdentifier{
3365        name="PODSLBiodiv/order";
3366        repository="www.ai4.uni-bayreuth.de";
3367        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/order";
3368      }
3369
3370      ConceptIdentifier familyIdentifier{
3371        name="PODSLBiodiv/family";
3372        repository="www.ai4.uni-bayreuth.de";
3373        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/family";
3374      }
3375
3376      ConceptIdentifier genusIdentifier{
3377        name="PODSLBiodiv/genus";
3378        repository="www.ai4.uni-bayreuth.de";
3379        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/genus";
3380      }
3381
3382      ConceptIdentifier subgenusIdentifier{
3383        name="PODSLBiodiv/subgenus";
3384        repository="www.ai4.uni-bayreuth.de";
3385        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/subgenus";
3386      }
3387
3388      ConceptIdentifier specificEpithetIdentifier{
3389        name="PODSLBiodiv/specificEpithet";
3390        repository="www.ai4.uni-bayreuth.de";
3391        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              specificEpithet";
3392      }
3393
3394      ConceptIdentifier taxonRankIdentifier{
3395        name="PODSLBiodiv/taxonRank";
3396        repository="www.ai4.uni-bayreuth.de";
3397        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/taxonRank";
3398      }
```

```
3399
3400      ConceptIdentifier verbatimTaxonRankIdentifier{
3401        name="PODSLBiodiv/verbatimTaxonRank";
3402        repository="www.ai4.uni-bayreuth.de";
3403        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              verbatimTaxonRank";
3404      }
3405
3406      ConceptIdentifier scientificNameAuthorshipIdentifier{
3407        name="PODSLBiodiv/scientificNameAuthorship";
3408        repository="www.ai4.uni-bayreuth.de";
3409        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              scientificNameAuthorship";
3410      }
3411
3412      ConceptIdentifier vernacularNameIdentifier{
3413        name="PODSLBiodiv/vernacularName";
3414        repository="www.ai4.uni-bayreuth.de";
3415        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              vernacularName";
3416      }
3417
3418      ConceptIdentifier nomenclaturalCodeIdentifier{
3419        name="PODSLBiodiv/nomenclaturalCode";
3420        repository="www.ai4.uni-bayreuth.de";
3421        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              nomenclaturalCode";
3422      }
3423
3424      ConceptIdentifier taxonomicStatusIdentifier{
3425        name="PODSLBiodiv/taxonomicStatus";
3426        repository="www.ai4.uni-bayreuth.de";
3427        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              taxonomicStatus";
3428      }
3429
3430      ConceptIdentifier nomenclaturalStatusIdentifier{
3431        name="PODSLBiodiv/nomenclaturalStatus";
3432        repository="www.ai4.uni-bayreuth.de";
3433        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              nomenclaturalStatus";
3434      }
3435
3436      ConceptIdentifier taxonRemarksIdentifier{
3437        name="PODSLBiodiv/taxonRemarks";
```

```
3438        repository="www.ai4.uni-bayreuth.de";
3439        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/taxonRemarks"
                ;
3440      }
3441
3442      ConceptIdentifier taxonLinkIdentifier{
3443        name="PODSLBiodiv/taxonLink";
3444        repository="www.ai4.uni-bayreuth.de";
3445        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/taxonLink";
3446      }
3447
3448      //Maps
3449
3450      ConceptIdentifier rightValueIdentifier{
3451        name="PODSLBiodiv/rightValue";
3452        repository="www.ai4.uni-bayreuth.de";
3453        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/rightValue";
3454      }
3455
3456      ConceptIdentifier heightValueIdentifier{
3457        name="PODSLBiodiv/heightValue";
3458        repository="www.ai4.uni-bayreuth.de";
3459        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/heightValue";
3460      }
3461
3462      ConceptIdentifier resolutionIdentifier{
3463        name="PODSLBiodiv/resolution";
3464        repository="www.ai4.uni-bayreuth.de";
3465        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/resolution";
3466      }
3467
3468
3469      //DataOriented
3470
3471      ConceptIdentifier yearOfPublicationIdentifier{
3472        name="PODSLBiodiv/yearOfPublication";
3473        repository="www.ai4.uni-bayreuth.de";
3474        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
                yearOfPublication";
3475      }
3476
3477      ConceptIdentifier mimeTypeIdentifier{
3478        name="PODSLBiodiv/mimeType";
3479        repository="www.ai4.uni-bayreuth.de";
3480        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/mimeType";
```

```
3481      }
3482
3483
3484      //Measurement
3485      ConceptIdentifier measurementValueIdentifier{
3486        name="PODSLBiodiv/measurementValue";
3487        repository="www.ai4.uni-bayreuth.de";
3488        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              measurementValue";
3489      }
3490
3491      ConceptIdentifier numericMeasurementValueIdentifier{
3492        name="PODSLBiodiv/numericMeasurementValue";
3493        repository="www.ai4.uni-bayreuth.de";
3494        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              numericMeasurementValue";
3495      }
3496
3497
3498
3499      ConceptIdentifier measurementTypeIdentifier{
3500        name="PODSLBiodiv/measurementType";
3501        repository="www.ai4.uni-bayreuth.de";
3502        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              measurementType";
3503      }
3504      ConceptIdentifier measurementUnitIdentifier{
3505        name="PODSLBiodiv/measurementUnit";
3506        repository="www.ai4.uni-bayreuth.de";
3507        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              measurementUnit";
3508      }
3509      ConceptIdentifier measurementAccuracyIdentifier{
3510        name="PODSLBiodiv/measurementAccuracy";
3511        repository="www.ai4.uni-bayreuth.de";
3512        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              measurementAccuracy";
3513      }
3514
3515      ConceptIdentifier descriptionIdentifier{
3516        name="PODSLBiodiv/description";
3517        repository="www.ai4.uni-bayreuth.de";
3518        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/description";
3519      }
3520
```

```
3521
3522      ConceptIdentifier verbatimDescriptionIdentifier{
3523        name="PODSLBiodiv/verbatimDescription";
3524        repository="www.ai4.uni-bayreuth.de";
3525        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              verbatimDescription";
3526      }
3527
3528      //Sampling
3529
3530      ConceptIdentifier samplingProtocolIdentifier{
3531        name="PODSLBiodiv/samplingProtocol";
3532        repository="www.ai4.uni-bayreuth.de";
3533        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              samplingProtocol";
3534      }
3535      ConceptIdentifier samplingEffortIdentifier{
3536        name="PODSLBiodiv/samplingEffort";
3537        repository="www.ai4.uni-bayreuth.de";
3538        address="model:/www.ai4.uni-bayreuth.de/PODSLBiodiv/
              samplingEffort";
3539      }
3540
3541
3542    }
3543
3544 }
3545 }
```

Listing C.3: PODSL-Biodiv

Literaturverzeichnis

[1] AKN. BMDE variable descriptions version 1.38. `http://www.avianknowledge.net/content/about/bmde-variable-descriptions`, 2013. Online; accessed 29-June-2013.

[2] S. Allamaraju. *RESTful Web Services Cookbook*. O'Reilly Media, 2010.

[3] G. Alonso, F. Casati, H. Kuno, and V. Machiriju. *Web Services – Concepts, Architectures and Applications*. Springer, 2004.

[4] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludascher, and S. Mock. Kepler: an extensible system for design and execution of scientific workflows. In *Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on*, pages 423 – 424, june 2004.

[5] Altova. Altova SchemaAgent. `http://www.altova.com/schemaagent/schema-management.html`, 2012. Online; accessed 14-July-2012.

[6] Éamonn Ó Tuama, V. Chavan, S. Gaiji, and T. Robertson. Gbif's global biodiversity resources discovery system. In *Proceedings of the TDWG 2008*, TDWG 08, page 107, 2009.

[7] C. Atkinson. Meta-modeling for distributed object environments. In *Proceedings of the 1st International Conference on Enterprise Distributed Object Computing*, EDOC '97, pages 90–101, Washington, DC, USA, 1997. IEEE Computer Society.

[8] D. Basci and S. Misra. Entropy metric for XML DTD documents. *SIGSOFT Softw. Eng. Notes*, 33:5:1–5:6, July 2008.

[9] D. Basci and S. Misra. Measuring and evaluating a design complexity metric for XML schema documents. *Journal of Information Science and Engineering*, 25(5):1405–1425, Sept. 2009.

[10] C. Batini, S. Ceri, and S. Navathe. *Conceptual database design : an entity-relationship approach*. Benjamin/Cummings, Redwood City, Cal., 1992. by Carlo Batini ; Stefano Ceri ; Shamkant B. Navathe.

[11] A. Bauer and H. Günzel. *Data Warehouse Systeme*. dpunkt.verlag, Heidelberg, 2008.

[12] C. Bauer and G. King. *Java Persistance with Hibernate*. Dreamtech Press, 2006.

[13] BDI. Biodiversitätsabteilung des Botanischen Gartens und Botanischen Museums Berlin-Dahlem. `http://www.bgbm.org/BioDivInf`, 2013. Online; accessed 22-May-2013.

[14] W. Berendsohn. Biodiversity informatics. `http://www.bgbm.org/BioDivInf/def-e.htm`, 2011. Online; accessed 22-August-2011; Preprint of an article to be published in the Proceedings of the Second National Colloquium on Global Change Research.

[15] W. Berendsohn and et al. Mapping between ABCD v. 2.06b concepts and those used in the DarwinCore (DwC) and its extensions. `http://www.bgbm.org/TDWG/CODATA/Schema/Mappings/DwCAndExtensions.htm`, 2007. Online; accessed 29-January-2013.

[16] BioCASE. BioCASE provider software. `http://www.biocase.org/products/provider_software/index.shtml`, 2011. Online; accessed 23-October-2012.

[17] BioCASE. A Biological Collection Access Service for Europe. `http://www.biocase.org`, 2013. Online; accessed 23-May-2013.

[18] BioCASE. CommonABCD2Concepts. `http://wiki.bgbm.org/bps/index.php/CommonABCD2Concepts`, 2013. Online; accessed 10-June-2013.

[19] BioCASE. Pywrapper-wiki. `http://wiki.bgbm.org/bps/index.php`, 2013. Online; accessed 10-June-2013.

[20] A. D. Birrell and B. J. Nelson. Implementing remote procedure calls. *ACM Trans. Comput. Syst.*, 2(1):39–59, Feb. 1984.

[21] F. Bisby, J. Coddington, J. Thorpe, et al. *Characterization of biodiversity*, pages 21–106. Cambridge University Press, Cambridge, 1995.

[22] M. E. Board. *Millennium Ecosystem Assesment: Ecosystems and human well-being*. Island Press Washington, DC, 2005.

[23] G. Booch, J. Rumbaugh, and I. Jacobson. *The unified modeling language user guide*. Addison-Wesley Professional, 2005.

[24] N. u. R. Bundesministerium für Umwelt. Nationale strategie zur biologischen vielfalt. `http://www.bmu.de/themen/natur-arten/naturschutz-biologische-vielfalt/nationale-strategie/`, 2007. Online; accessed 16-June-2013; beschlossen vom Bundeskabinett am 07.November 2007.

[25] P. Buneman, S. Khanna, and T. Wang-Chiew. Why and where: A characterization of data provenance. *Database Theory – ICDT 2001*, pages 316–330, 2001.

[26] C. Bussler. *Organisationsverwaltung in Workflow-Management-Systemen*. DUV, 1998.

[27] A. Campbell. Save those molecules! molecular biodiversity and life. *Journal of applied ecology*, 40(2):193–203, 2003.

[28] CBD. The convention on biological diversity. `https://www.cbd.int/convention/`, 1992. opened for signature at the Earth Summit in Rio de Janeiro on 5 June 1992;Online; accessed 16-August-2011.

[29] CBD. Homepage of the convention on biological diversity. `https://www.cbd.int`, 2013. Online; accessed 16-March-2013.

[30] S. Chaudhuri and U. Dayal. An overview of data warehousing and olap technology. *SIGMOD Rec.*, 26:65–74, March 1997.

[31] P. Chen. The entity-relationship model—toward a unified view of data. *ACM Transactions on database systems*, 1(1):9–36, 1976.

[32] T. Clark, P. Sammut, and J. Willans. Applied metamodelling: a foundation for language driven development. `http://eprints.mdx.ac.uk/6060/1/Clark-Applied_Metamodelling_%28Second_Edition%29%5B1%5D.pdf`, 2008. Online; accessed 17-February-2013.

[33] G. Cochrane, I. Karsch-Mizrachi, and Y. Nakamura. The international nucleotide sequence database collaboration. *Nucleic Acids Research*, 2010.

[34] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13:377–387, June 1970.

[35] COL. Catalogue of life. `http://www.catalogueoflife.org`, 2011. Online; accessed 24-August-2011.

[36] C. Cornelius, A. Leingärtner, B. Hoiss, J. Krauss, I. Steffan-Dewenter, and A. Menzel. Phenological response of grassland species to manipulative snowmelt and drought along an altitudinal gradient. *Journal of experimental botany*, 64(1):241–251, 2013.

[37] R. Costanza, R. d'Arge, R. De Groot, S. Farber, M. Grasso, B. Hannon, K. Limburg, S. Naeem, R. O'Neill, J. Paruelo, et al. The value of the world's ecosystem services and natural capital. *Nature*, 387(6630):253–260, 1997.

[38] R. Costello. XML schema: Best practices. `http://www.xfront.com/BestPracticesHomepage.html`, 2012. Online; accessed 20-July-2012.

[39] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana. Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI. *Internet Computing, IEEE*, 6(2):86–93, Mar. 2002.

[40] O. Cure, S. Jablonski, F. Jochaud, M. Rehman, and B. Volz. Semantic data integration in the dalton system. In *Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on*, pages 234 –241, april 2008.

[41] M. Dallwitz. A general system for coding taxonomic descriptions. *Taxon*, 29:41–46, February 1980.

[42] R. Dasmann. *A different kind of country.* Macmillan New York, 1968.

[43] dataONE. dataONE Architecture. `http://mule1.dataone.org/ArchitectureDocs-current/index.html#/`, 2012. Online; accessed 22-November-2012.

[44] dataONE. dataONE Homepage. `http://www.dataone.org/`, 2012. Online; accessed 15-November-2012.

[45] R. De Giovanni. TDWG Task Group Homepage. `http://www.tdwg.org/activities/tapir/`, 2009. Online; accessed 23-August-2012.

[46] R. De Giovanni and C. Copp. TAPIR - TDWG access protocol for information retrieval protocol specification - version 1.0. `http://www.tdwg.org/dav/`

subgroups/tapir/1.0/docs/tdwg_tapir_specification_2010-05-05.htm, 2010. Online; accessed 23-August-2012.

[47] W. De la Mare. Abrupt mid-twentieth-century decline in antarctic sea-ice extent from whaling records. *Nature*, 389(6646):57–60, 1997.

[48] K. De Queiroz. Ernst mayr and the modern concept of species. *Proceedings of the National Academy of Sciences of the United States of America*, 102(Suppl 1):6600, 2005.

[49] DFG. Empfehlungen der Kommission 'Selbstkontrolle in der Wissenschaft', Vorschläge zur Sicherung guter wissenschaftlicher Praxis. http://www.dfg.de/download/pdf/dfg_im_profil/reden_stellungnahmen/download/empfehlung_wiss_praxis_0198.pdf, 1998. Online; accessed 29-July-2011.

[50] Diversity Workbench. Diversity Workbench – a virtual research environment for building and accessing biodiversity and environmental data. http://www.diversityworkbench.net, 2013. Online; accessed 16-May-2013.

[51] DiversityMobile. DiversityMobile Documentation. http://www.diversitymobile.net/wiki/media/Documentation_DiversityMobile_under_WindowsPhone_Version1.pdf, 2012. Online; accessed 29-March-2013.

[52] DiversityMobile. DiversityMobile Wiki. http://www.diversitymobile.net, 2013. Online; accessed 29-March-2013.

[53] H. Do and E. Rahm. COMA: a system for flexible combination of schema matching approaches. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 610–621, 2002.

[54] M. Donoghue. A critique of the biological species concept and recommendations for a phylogenetic alternative. *Bryologist*, pages 172–181, 1985.

[55] C. Drumm, M. Schmitt, H. Do, and E. Rahm. Quickmig: automatic schema matching for data migration projects. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 107–116. ACM, 2007.

[56] Duden. Duden Online. http://www.duden.de/rechtschreibung, 2011. Online; accessed 06-July-2012.

[57] EDIT. Biodiversity service & application tracker. `http://www.bdtracker.net`, 2011. Online; accessed 26-July-2011.

[58] EDIT. Common data model v.2.2. `http://wp5.e-taxonomy.eu/cdm/v22`, 2011. Online; accessed 29-July-2011.

[59] EDIT. European Distributed Institute of Taxonomy. `http://www.e-taxonomy.eu`, 2011. Online; accessed 23-August-2011.

[60] EDIT. Common data model. `http://dev.e-taxonomy.eu/trac/wiki/CommonDataModel`, 2012. Online; accessed 07-August-2012.

[61] EDIT. Common data model v.3.1. `http://wp5.e-taxonomy.eu/cdm/v31`, 2012. Online; accessed 07-August-2012.

[62] D. Ehrenfeld. Why put a value on biodiversity. *Biodiversity*, pages 212–216, 1988.

[63] P. Ehrlich. Has the biological species concept outlived its usefulness? *Systematic Biology*, 10(4):167, 1961.

[64] P. Ehrlich and A. Ehrlich. *Extinction: the causes and consequences of the disappearance of species*. Random House New York, 1981.

[65] D. W. Embley, S. W. Liddle, and R. Al-Kamha. Enterprise modeling with conceptual XML. In *ER*, pages 150–165, 2004.

[66] D. W. Embley and B. Thalheim. *Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges*. Springer Publishing Company, Incorporated, 1st edition, 2011.

[67] D. Endresen and et al. Darwincore-germplasm. `http://code.google.com/p/darwincore-germplasm/`, 2010. Online; accessed 29-July-2011.

[68] M. Faerber, S. Jablonski, and T. Schneider. A comprehensive modeling language for clinical processes. In *2nd European Conference on eHealth (ECEH 2007)*, 2007.

[69] X. Feng, J. Shen, and Y. Fan. REST: An alternative to RPC for web services architecture. In *Future Information Networks, 2009. ICFIN 2009. First International Conference on*, pages 7–10. IEEE, 2009.

[70] A. Ferrara and M. MacDonald. *.NET Web-Services*. O'Reilly Media, 2002.

[71] R. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, California, USA, 2000. http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm, Online; accessed 10-October-2012.

[72] K. Gaston and J. Spicer. *Biodiversity: an introduction*. Wiley-Blackwell, 2004.

[73] GBIF. A blueprint of the GBIF decentralisation strategy. http://www.gbif.org/communications/resources/posters/. Online; accessed 16-October-2012.

[74] GBIF. Towards establishing a functional GBIF participant node. http://www.gbif.org/orc/?doc_id=2745, 2009. Online; accessed 16-October-2012.

[75] GBIF. GBIF memorandum of understanding. http://www.gbif.org/orc/?doc_id=2955, 2010. Online; accessed 16-October-2012.

[76] GBIF. The GBIF harvesting and indexing toolkit (HIT). http://code.google.com/p/gbif-indexingtoolkit/, 2012. Online; accessed 16-October-2012.

[77] GBIF. http://code.google.com/p/gbif-npt/. http://code.google.com/p/gbif-npt/, 2012. Online; accessed 16-October-2012.

[78] GBIF. The integrated publishing toolkit. http://www.gbif.org/informatics/infrastructure/publishing/, 2012. Online; accessed 16-October-2012.

[79] GBIF. Resource disovery. http://www.gbif.org/informatics/standards-and-tools/integrating-data/resource-discovery/, 2012. Online; accessed 23-October-2012.

[80] GBIF. GBIF data portal. http://data.gbif.org/, 2013. Online; accessed 16-January-2013.

[81] GBIF. GBIF infrastructure. http://www.gbif.org/informatics/infrastructure/, 2013. Online; accessed 26-June-2013.

[82] GBIF. Global biodiversity information facility. http://www.gbif.org, 2013. Online; accessed 26-June-2013.

[83] GBIF. Multimedia resources in biodiversity. http://www.gbif.org/informatics/primary-data/types-of-primary-biodiversity-data/multimedia-resources-data/, 2013. Online; accessed 26-March-2013.

[84] GBIF Deutschland. Homepage GBIF Deutschland. `http://http://www.gbif.de/`, 2012. Online; accessed 16-November-2012.

[85] M. Genero, G. Poels, and M. Piattini. Defining and validating metrics for assessing the understandability of entity-relationship diagrams. *Data & Knowledge Engineering*, 64(3):534 – 557, 2008.

[86] C. Gonzalez-Perez and B. Henderson-Sellers. *Metamodelling for software engineering*. Wiley, 2008.

[87] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.*, 43(5-6):907–928, Dec. 1995.

[88] G. Guerrini and M. Mesiti. X-evolution: A comprehensive approach for xml schema evolution. In *DEXA Workshops*, pages 251–255, 2008.

[89] H.-P. Gumm and M. Sommer. *Einführung in die Informatik*. Oldenbourg Wissenschaftsverlag, 2012.

[90] G. Hagedorn. *Structuring Descriptive Data of Organisms - Requirement Analysis and Information Models*. PhD thesis, University of Bayreuth, Bayreuth, Germany, 2007.

[91] G. Hagedorn, K. Thiele, R. Morris, and P. Heidorn. The structured descriptive data (sdd) w3c-xml-schema, version 1.0. `http://www.tdwg.org/standards/116/`, 2005. Online; accessed 29-July-2012.

[92] G. Hagedorn, K. Thiele, R. Morris, and P. Heidorn. The structured descriptive data (SDD) w3c-xml-schema, version 1.1. `http://rs.tdwg.org/UBIF/2006/rddl.html`, 2006. Online; accessed 29-July-2012.

[93] A. Halevy, A. Doan, and Z. G. I. (Autor). *Principles of data integration*. Morgan Kaufmann, Waltham, 2012.

[94] A. Halevy, A. Rajaraman, and J. Ordille. Data integration: the teenage years. In *Proceedings of the 32nd international conference on Very large data bases*, VLDB '06, pages 9–16. VLDB Endowment, 2006.

[95] A. Y. Halevy. Data integration: A status report. In *BTW*, pages 24–29, 2003.

[96] M. Hammer and D. Mc Leod. Database description with SDM: a semantic database model. *ACM Transactions on Database Systems (TODS)*, 6(3):351–386, 1981.

[97] J. Harper and D. Hawksworth. Biodiversity: measurement and estimation. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 345(1311):5, 1994.

[98] D. Hay. *Requirements analysis: from business views to architecture*, chapter 3. Prentice Hall, 2003.

[99] L. Heinrich. *J.; Heinzl, A.; Roithmayr, F.: Wirtschaftsinformatik-Lexikon. 7. Auflage*. Oldenbourg Wissenschaftsverlag, Miinchen/Wien, 2004.

[100] L. Heinrich and D. Stelzer. *Informationsmanagement. Grundlagen, Aufgaben, Methoden. 9. Auflage*. München: Oldenbourg, 2009.

[101] M. Henning. The rise and fall of CORBA. *Queue*, 4(5):28–34, June 2006.

[102] V. Heywood. *Introduction*, pages 5–19. Cambridge University Press, 1995.

[103] V. H. Heywood et al. *Global biodiversity assessment*. Cambridge University Press, 1995.

[104] D. Hill. *Handbook of biodiversity methods: survey, evaluation and monitoring*. Cambridge Univ Pr, 2005.

[105] J. Hobbie, S. Carpenter, N. Grimm, J. Gosz, and T. Seastedt. The us long term ecological research program. *BioScience*, 53(1):21–32, 2003.

[106] IBF Project. IBF homepage. `http://www.diversitymobile.net/wiki/IBF_Project`, 2009. Online; accessed 29-March-2013.

[107] IBM. Information integration. `http://www-01.ibm.com/software/data/integration/products.html`. Online; accessed 21-October-2012.

[108] M. Igler. *ESProNa – Eine Constraintsprache zur multimodalen Prozessmodellierung und navigationsgestützten Ausführung*. PhD thesis, University of Bayreuth, Bayreuth, Germany, 2012.

[109] Informatica. Informatica homepage. `http://www.informatica.com/de/vision/a-platform-approach/`. Online; accessed 30-May-2013.

[110] INSDC. The DDBJ/EMBL/GenBank feature table definition. `http://www.insdc.org/documents/feature_table.html#5.1`, 2012. Online; accessed 24-November-2012.

[111] INSDC. INSDC homepage. `http://www.insdc.org/`, 2012. Online; accessed 24-November-2012.

[112] ISO. ISO 9126. `http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=22749`, 2012. Online; accessed 14-July-2012.

[113] S. Jablonski. Functional and behavioral aspects of process modeling in workflow management systems. In *Proceedings of the ninth Austrian-informatics conference on Workflow management : challenges, paradigms and products: challenges, paradigms and products*, pages 113–133, Munich, Germany, Germany, 1994. R. Oldenbourg Verlag GmbH.

[114] S. Jablonski and C. Bussler. *Workflow management - modeling concepts, architecture and implementation*. International Thomson, 1996.

[115] S. Jablonski, O. Cure, M. Rehman, and B. Volz. Architecture of the dalton data integration system for scientific applications. In *Cluster Computing and the Grid, 2008. CCGRID '08. 8th IEEE International Symposium on*, page 701, may 2008.

[116] S. Jablonski, A. Kehl, D. Neubacher, P. Poschlod, G. Rambold, T. Schneider, D. Triebel, B. Volz, and M. Weiss. Diversitymobile-mobile data retrieval platform for biodiversity research projects. In *Proceedings of the 39nd annual meeting of the GI*, GI '09, pages 610–624. Köllen Druck + Verlag GmbH, 2009.

[117] S. Jablonski, B. Volz, and M. A. Rehman. A conceptual modeling and execution framework for process based scientific applications. In *CIMS*, pages 23–30, 2007.

[118] S. Jablonski, B. Volz, and M. A. Rehman. Process and ontology based data integration. In *Proceedings of the 22th Conference on Environmental Informatics and Industrial Ecologyy*, EnviroInfo '08, pages 567–575, 2008.

[119] P. Janisch. *Überblick zu methodischen Grundproblemen der Biodiversität*, volume 10, chapter 1. Springer, 2002.

[120] N. Johnson. Biodiversity informatics. *Annu. Rev. Entomol.*, 52:421–438, 2007.

[121] A. Kehl, S. Dötterl, G. Aas, and G. Rambold. Is flower scent influencing host plant selection of leaf-galling sawflies (hymenoptera, tenthredinidae) on willows? *Chemoecology*, 20(3):215–221, 2010.

[122] A. Kemper and A. Eickler. *Datenbanksysteme: Eine Einführung*. Oldenbourg Wissenschaftsverlag, 2011.

[123] S. Kesh. Evaluating the quality of entity relationship models. *Information and Software Technology*, 37(12):681 – 689, 1995.

[124] G. King, C. Bauer, M. R. Andersen, E. Bernard, and S. Ebersole. Hibernate reference. `http://docs.jboss.org/hibernate/orm/3.3/reference/en/html/`, 2009. Online; accessed 212-February-2013.

[125] M. Klettke, L. Schneider, and A. Heuer. Metrics for xml document collections. In A. Chaudhri, R. Unland, C. Djeraba, and W. Lindner, editors, *XML-Based Data Management and Multimedia Engineering – EDBT 2002 Workshops*, volume 2490 of *Lecture Notes in Computer Science*, pages 519–523. Springer Berlin / Heidelberg, 2002.

[126] J. Klímek and J. Malý. exolutio. `http://exolutio.com/`, 2012. Online; accessed 14-July-2012.

[127] J. Klímek, J. Malý, I. Mlynkova, and M. Necasky. exolutio: Tool for xml schema and data management. In J. Pokorný, V. Snásel, and K. Richta, editors, *DATESO*, volume 837 of *CEUR Workshop Proceedings*, pages 69–80. CEUR-WS.org, 2012.

[128] KNB. Ecological metadata language v.2.1.0. `http://knb.ecoinformatics.org/software/eml/eml-2.1.0/index.html`, 2011. Online; accessed 29-July-2011.

[129] L. Koh, R. Dunn, N. Sodhi, R. Colwell, H. Proctor, and V. Smith. Species coextinctions and the biodiversity crisis. *Science*, 305(5690):1632, 2004.

[130] J. Krogstie, O. Lindland, and G. Sindre. Towards a deeper understanding of quality in requirements engineering. In J. Iivari, K. Lyytinen, and M. Rossi, editors, *Advanced Information Systems Engineering*, volume 932 of *Lecture Notes in Computer Science*, pages 82–95. Springer Berlin / Heidelberg, 1995.

[131] J. Krogstie, G. Sindre, and H. Jørgensen. Process models representing knowledge for action: a revised quality framework. *Eur. J. Inf. Syst.*, 15(1):91–102, Feb. 2006.

[132] J. Krogstie and A. Sølvberg. *Information systems engineering: Conceptual modeling in a quality perspective.* Kompendiumforlaget, Trondheim, Norway, 2003.

[133] M. Kuussaari, R. Bommarco, R. Heikkinen, A. Helm, J. Krauss, R. Lindborg, E. Ockinger, M. Partel, J. Pino, F. Rodà, et al. Extinction debt: a challenge for biodiversity conservation. *Trends in ecology & evolution*, 24(10):564–571, 2009.

[134] B. Lahres and G. Raỳman. Objektorientierte programmierung. *Galileo Computing*, 2, 2009.

[135] M. Lane and J. Edwards. *The Global Biodiverity Information Facility (GBIF)*, volume 73, chapter 1. CRC, 2007.

[136] R. C. Lathrop, S. R. Carpenter, and L. G. Rudstam. Water clarity in lake mendota since 1900: responses to differing levels of nutrients and herbivory. *Canadian Journal of Fisheries and Aquatic Sciences*, 53:2250–2261, 1996.

[137] R. Leinonen, H. Sugawara, and M. Shumway. The sequence read archive. *Nucleic acids research*, 39(suppl 1):D19–D21, 2011.

[138] M. Lenzerini. Data integration: a theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '02, pages 233–246, New York, NY, USA, 2002. ACM.

[139] U. Leser and F. Naumann. *Informationsintegration - Architekturen und Methoden zur Integration verteilter und heterogener Datenquellen.* dpunkt.verlag, 2007.

[140] LIAS. LIAS – a global information system for lichenized and non-lichenized ascomycetes. `http://www.lias.net/`, 2013. Online; accessed 24-January-2013.

[141] P. Liggesmeyer. *Software-Qualität: Testen, Analysieren und Verifizieren von Software.* Spektrum Akademischer Verlag, 2009.

[142] O. Lindland, G. Sindre, and A. Solvberg. Understanding quality in conceptual modeling. *Software, IEEE*, 11(2):42–49, 1994.

[143] R. Lämmel, S. Kitsis, and D. Remy. Analysis of XML schema usage. In *Proceedings of XML 2005*, 2005.

[144] B. F. Lósio, A. C. Salgado, and L. do Rêgo GalvĐo. Conceptual modeling of XML schemas. In *Proceedings of the 5th ACM international workshop on Web information and data management*, WIDM '03, pages 102–105, New York, NY, USA, 2003. ACM.

[145] LTER. LTER metadata systems. `http://im.lternet.edu/siteprofiles/Site_metadata_systems`, 2008. Online; accessed 26-November-2012.

[146] LTER. Long term ecological research network. `http://www.lternet.edu`, 2012. Online; accessed 26-November-2012.

[147] LTER-Europe. European long-term ecosystem research network. `http://www.lter-europe.net/`, 2012. Online; accessed 26-November-2012.

[148] J. Madhavan, P. Bernstein, and E. Rahm. Generic schema matching with cupid. In *Proceedings of the International Conference on Very Large Data Bases*, pages 49–58, 2001.

[149] J. Madin, S. Bowers, M. Schildhauer, S. Krivov, D. Pennington, and F. Villa. An ontology for describing and synthesizing ecological observation data. *Ecological Informatics*, 2(3):279 – 296, 2007. Meta-information systems and ontologies. A Special Feature from the 5th International Conference on Ecological Informatics ISEI5, Santa Barbara, CA, Dec. 4-7, 2006 - Novel Concepts of Ecological Data Management S.I.

[150] E. Maler. Schema design rules for UBL...and maybe for you. In *Proceedings of IDEAlliance XML 2002 Conference*, 2002.

[151] J. Malý, I. Mlýnková, and M. Necaský. Xml data transformations as schema evolves. In *ADBIS*, pages 375–388, 2011.

[152] R. Marggraf. *Ökonomische Aspekte*, volume 10, chapter 1. Springer, 2002.

[153] M. Markussen, R. Buse, H. Garrelts, M. Manez Costa, S. Menzel, and R. Marggraf. *General introduction*, chapter 1. Springer, 2005.

[154] B. Marnette, G. Mecca, P. Papotti, S. Raunich, and D. Santoro. ++ spicy: an open-source tool for second-generation schema mapping and data exchange. *Clio*, 19:21, 2011.

[155] E. Mayr. *Systematics and the origin of species*. Columbia University Press, 1942.

[156] E. Mayr. Animal species and evolution. *Animal species and their evolution*, 1963.

[157] P. McDaniel. Data provenance and security. *Security & Privacy, IEEE*, 9(2):83–85, 2011.

[158] A. McDowell, C. Schmidt, and K. bun Yue. Analysis and Metrics of XML Schema. In H. R. Arabnia and H. Reza, editors, *Software Engineering Research and Practice*, pages 538–544. CSREA Press, 2004.

[159] J. McNeill, F. Barrie, H. Burdet, V. Demoulin, D. Hawksworth, K. Marshold, D. Nicolson, J. Prado, C. Silva, J. Skog, J. Wiersema, and N. Turland, editors. *International Code of Botanical Nomenclature (VIENNA CODE)*. A.R.G. Gantner Verlag KG., 2006. adopted by the Seventeenth International Botanical Congress Vienna, Austria, July 2005.

[160] G. Mecca, P. Papotti, and D. Santoro. A short history of schema mapping systems. In *SEBD*, pages 99–106, 2012.

[161] S. Melnik, E. Rahm, and P. Bernstein. Rondo: A programming platform for generic model management. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 193–204. ACM, 2003.

[162] W. Michener, J. Porter, M. Servilla, and K. Vanderbilt. Long term ecological research and information management. *Ecological Informatics*, 6(1):13–24, 2011.

[163] W. Michener, D. Vieglais, T. Vision, J. Kunze, P. Cruse, and G. Janée. Dataone: Data observation network for earth-preserving data and enabling innovation in the biological and environmental sciences. *D-Lib Magazine*, 17(1):3, 2011. `http://dlib.org/dlib/january11/michener/01michener.print.html`; Online; accessed 21-November-2012.

[164] Microsoft. Iis homepage. `www.iis.net`, 2012. Online; accessed 15-October-2012.

[165] Microsoft. Overview of the .net framework. `http://msdn.microsoft.com/en-us/library/zw4w595w.aspx/`, 2012. Online; accessed 15-October-2012.

[166] Microsoft. What is windows communication foundation. `http://msdn.microsoft.com/de-DE/library/ms731082.aspx`, 2012. Online; accessed 15-October-2012.

[167] P. Missier, B. Ludascher, S. Bowers, M. K. Anand, I. Altintas, S. Dey, A. Sarkar, B. Shrestha, and C. Goble. Linking multiple workflow provenance traces for interoperable collaborative science. In *Proc.s 5th Workshop on Workflows in Support of Large-Scale Science (WORKS)*, 2010.

[168] D. L. Moody. Metrics for evaluating the quality of entity relationship models. In *Proceedings of the 17th International Conference on Conceptual Modeling*, ER '98, pages 211–225, London, UK, 1998. Springer-Verlag.

[169] D. L. Moody. Strategies for improving the quality of entity relationship models: a “toolkit” for practitioners. In *Proceedings of the 2000 information resources management association international conference on Challenges of information technology management in the 21st century*, pages 1043–1045, Hershey, PA, USA, 2000. IGI Publishing.

[170] D. L. Moody. Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data Knowl. Eng.*, 55:243–276, December 2005.

[171] D. L. Moody and G. G. Shanks. What makes a good data model? evaluating the quality of entity relationship models. In *Proceedings of the13th International Conference on the Entity-Relationship Approach*, ER '94, pages 94–111, London, UK, 1994. Springer-Verlag.

[172] D. L. Moody, G. G. Shanks, and P. Darke. Improving the quality of entity relationship models - experience in research and practice. In *Proceedings of the 17th International Conference on Conceptual Modeling*, ER '98, pages 255–276, London, UK, 1998. Springer-Verlag.

[173] D. L. Moody, G. Sindre, T. Brasethvik, and A. Sølvberg. Evaluating the quality of information models: empirical testing of a conceptual model quality framework. In *Proceedings of the 25th International Conference on Software Engineering*, ICSE '03, pages 295–305, Washington, DC, USA, 2003. IEEE Computer Society.

[174] M. Morrison, D. Brownell, and F. Boumphrey. *XML Unleashed*. Sams, Indianapolis, IN, USA, 1999.

[175] N. Myers, R. Mittermeier, C. Mittermeier, G. da Fonseca, and J. Kent. Biodiversity hotspots for conservation priorities. *Nature*, 403(6772):853–858, 2000.

[176] NBN. National biodiversity network exchange format. http://www.nbn.org.uk/Share-Data/Providing-Data/NBN-Data-Exchange-format.aspx, 2012. Online; accessed 10-August-2012.

[177] NBN. Homepage national biodiversity network. http://www.nbn.org.uk, 2013. Online; accessed 22-June-2013.

[178] NCEAS. Semantic tools for data management. https://semtools.ecoinformatics.org/, 2012. Online; accessed 11-August-2012.

[179] M. Necasky. Conceptual modeling for XML: A survey. Technical report, In Proceedings of the DATESO 2006 Annual International Workshop on Databases, Texts, Specifications and Objects (DATESO 2006, 2006.

[180] M. Necasky and J. Pokorny. Extending e-r for modelling xml keys. In *Digital Information Management, 2007. ICDIM '07. 2nd International Conference on*, volume 1, pages 236 –241, oct. 2007.

[181] H. J. Nelson, G. Poels, M. Genero, and M. Piattini. A conceptual modeling quality framework. *Software Quality Control*, 20(1):201–228, Mar. 2012.

[182] N. F. Noy. Semantic integration: A survey of ontology-based approaches. *SIGMOD Record*, 33:2004, 2004.

[183] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat, and P. Li. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–3054, 2004.

[184] OMG. Life sciences identifiers final adopted specification. http://www.omg.org/cgi-bin/doc?dtc/04-05-01. Online; accessed 23-August-2012.

[185] OMG. Life sciences identifiers (LIS). http://www.omg.org/spec/LIS/. Online; accessed 23-August-2012.

[186] OMG. OMG life sciences identifiers specification (LSID). Technical report, OMG, 2005. Online; accessed 23-August-2012.

[187] OMG. *Common Object Request Broker Architecture (CORBA)*, 2011. Online; accessed 10-October-2012.

[188] OMG. *Meta Object Facility (MOF) Core Specification Version 2.4.1*, 2011. Online; accessed 10-August-2012.

[189] OMG. UML v. 2.4.1. http://www.omg.org/spec/UML/2.4.1/, 2011. Online; accessed 18-August-2012.

[190] Oracle. Oracle data integrator. http://www.oracle.com/technetwork/middleware/data-integrator/overview/index.html. Online; accessed 21-October-2012.

[191] Oracle. Java EE at a glance. http://www.oracle.com/technetwork/java/javaee/overview/index.html, 2013. Online; accessed 15-June-2013.

[192] S. Ou. *Rice diseases.* CABI, 1985.

[193] M. T. Özsu and P. Valduriez. *Principles of distributed database systems.* Springer Science+ Business Media, 2011.

[194] S. Patig. IT Infrastrukturen. In *Enzyklopädie der Wirtschaftsinformatik – Online–Lexikon.* Oldenbourg, 4 edition, 2008. Online; accessed 22-August-2011.

[195] C. Pautasso, O. Zimmermann, and F. Leymann. Restful web services vs. "big" web services: making the right architectural decision. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 805–814, New York, NY, USA, 2008. ACM.

[196] D. Pearce. *Blueprint 4: capturing global environmental value.* Earthscan Publications Ltd, 1995.

[197] H. Pereira and H. David Cooper. Towards the global monitoring of biodiversity change. *Trends in Ecology & Evolution*, 21(3):123–129, 2006.

[198] G. Psaila. Erx: A conceptual model for xml documents. In *SAC (2)*, pages 898–903, 2000.

[199] M. Rands, W. Adams, L. Bennun, S. Butchart, A. Clements, D. Coomes, A. Entwistle, I. Hodge, V. Kapos, J. Scharlemann, et al. Biodiversity conservation: challenges beyond 2010. *Science*, 329(5997):1298, 2010.

[200] Regents of the University of California. Metacat: Metadata and data management server. http://knb.ecoinformatics.org/knb/docs/, 2012. Online; accessed 26-November-2012.

[201] A. Rehman. *Process and Data Management Support for Scientific Applications - Theoretical and Practical Issues.* PhD thesis, University of Bayreuth, 2010.

[202] M. A. Rehman, S. Jablonski, and B. Volz. An ontology based approach to automating data integration in scientific workflows. In *Proceedings of the 7th International Conference on Frontiers of Information Technology*, FIT '09, pages 44:1–44:6, New York, NY, USA, 2009. ACM.

[203] D. Remsen and M. Döring. The GBIF global names architecture. In *Proceedings of TDWG*. TDWG, 2009.

[204] L. Richardson and S. Ruby. *RESTful Web Services*. O'Reilly Media, 2007.

[205] B. Roth, B. Volz, and R. Hecht. Data integration systems for scientific applications. In *OTM Workshops*, pages 110–118, 2010.

[206] G. Saake, K. Sattler, and A. Heuer. *Datenbanken: Implementierungstechniken*. mitp, 2011.

[207] P. Sadalage and M. Fowler. *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Addison-Wesley Professional, 2012.

[208] J. Sandoval. *RESTful Java Web Services*. Packt Publishing, 2009.

[209] SAP. SAP data integrator. `http://www.sap.com/solutions/technology/enterprise-information-management/data-integrator/index.epx`. Online; accessed 21-October-2012.

[210] SAS. SAS enterprise data integration server. `http://www.sas.com/software/data-management/entdiserver/`. Online; accessed 30-May-2013.

[211] T. Schneider, M. Weiss, and D. Triebel. DiversityMobile information model (version 1.2). `http://diversityworkbench.net//Portal/MobileModel_v1.2`, 2011. Online; accessed 16-November-2012.

[212] R. Scholes, G. Mace, W. Turner, G. Geller, N. Jürgens, A. Larigauderie, D. Muchoney, B. Walther, and H. Mooney. Toward a global biodiversity observing system. *Science*, 321(5892):1044–1045, 2008.

[213] S. Schönig, C. Günther, M. Zeising, and S. Jablonski. Discovering cross-perspective semantic definitions from process execution logs. In *BUSTECH 2012, The Second International Conference on Business Intelligence and Technology*, pages 1–7, 2012.

[214] E. Seidewitz. What models mean. *IEEE Software*, 20(5):26–32, 2003.

[215] L. Seligman, P. Mork, A. Halevy, K. Smith, M. Carey, K. Chen, C. Wolf, J. Madhavan, A. Kannan, and D. Burdick. Openii: an open source information integration toolkit. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 1057–1060. ACM, 2010.

[216] M. Servilla, J. Brunt, I. San Gil, and D. Costa. PASTA: A network-level architecture design for generating synthetic data products in the lter network. *LTER DataBits Fall*, 2006. Online; accessed 21-November-2012.

[217] P. Shvaiko and J. Euzenat. Ontology matching: State of the art and future challenges. *Knowledge and Data Engineering, IEEE Transactions on*, PP(99):1, 2012.

[218] J. Simmel. IBF Fungi – großpilz-kartierung mit smartphone und gps. *Denkschriften der Regensburgischen Botanischen Gesellschaft*, 72:139–170, 2011.

[219] Y. Simmhan, B. Plale, and D. Gannon. A survey of data provenance techniques. *Computer Science Department, Indiana University, Bloomington IN*, 47405, 2005.

[220] SNSB-IT-Center. IT-Center of the staatliche naturwissenschaftliche sammlungen bayerns. `http://www.snsb.info`, 2013. Online; accessed 22-March-2013.

[221] J. Soberón and T. Peterson. Biodiversity informatics: managing and applying primary biodiversity data. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 359(1444):689, 2004.

[222] S. A. Socher, D. Prati, S. Boch, J. Müller, H. Baumbach, S. Gockel, A. Hemp, I. Schöning, K. Wells, F. Buscot, et al. Interacting effects of fertilization, mowing and grazing on plant species diversity of 1500 grasslands in germany differ between regions. *Basic and Applied Ecology*, 2013.

[223] R. Sokal and T. Crovello. The biological species concept: a critical evaluation. *American Naturalist*, pages 127–153, 1970.

[224] Species2000. Species2000. `http://www.species2000.org`, 2011. Online; accessed 24-August-2011.

[225] H. Stachowiak. *Allgemeine Modelltheorie*. Springer-Verlag, Wien, 1973.

[226] B. Streit. *Was ist Biodiversität?: Erforschung, Schutz und Wert biologischer Vielfalt*, volume 2417. CH Beck, 2007.

[227] D. Takacs. *The idea of biodiversity: philosophies of paradise.* Johns Hopkins University Press Baltimore, MD, 1996.

[228] TAPIR Task Group. TAPIR - TDWG access protocol for information retrieval (cover page). http://www.tdwg.org/standards/449/, 2009. Online; accessed 23-August-2012.

[229] TDWG. Access to Biological Collection Data - version 2.06. http://www.tdwg.org/standards/115/, 2006. Online; accessed 29-July-2012.

[230] TDWG. TDWG ontology. http://wiki.tdwg.org/twiki/bin/view/TAG/TDWGOntology/, 2006. Online; accessed 26-August-2012.

[231] TDWG. Darwin Core. http://www.tdwg.org/standards/450/, 2009. Online; accessed 29-July-2012.

[232] TDWG. Biodiversity information standards - TDWG. http://www.tdwg.org, 2013. Online; accessed 29-June-2013.

[233] TDWG Globally Unique Identifiers Task Group. GUID and life sciences identifiers applicability statements. http://www.tdwg.org/standards/150/, 2011. Online; accessed 29-July-2012.

[234] TDWG Wiki: ABCD. Access to Biological Collection Data - Extensions. http://wiki.tdwg.org/twiki/bin/view/ABCD/DesignAbcdExtensions, 2010. Online; accessed 29-July-2012.

[235] The BRAHMS Project. BRAHMS-botanical research and herbarium management system. http://dps.plants.ox.ac.uk/bol/BRAHMS/Home/Default, 2011. Online; accessed 29-July-2011.

[236] K. Thiele and D. Sharp. Sdd part 0: Introduction and primer to the sdd standard. http://wiki.tdwg.org/twiki/bin/view/SDD/Primer/WebHome, 2006. Online; accessed 29-July-2012.

[237] N. Thomson, M. Döring, R. De Giovanni, J. De la Torre, W. Berendsohn, W. Addink, and W. Ulate. Access to Biological Collection Data - Primer. http://wiki.tdwg.org/twiki/bin/view/ABCD/AbcdPrimer, 2007. Online; accessed 29-July-2012.

[238] E. Thoo, T. Friedman, and M. A. Beyer. Magic quadrant for data integration tools. http://www.gartner.com/technology/reprints.do?id=1-17QG4XL&ct=111020&st=sb. Online; accessed 14-October-2012.

[239] E. Thoo, T. Friedman, and M. A. Beyer. Magic quadrant for data integration tools. http://www.gartner.com/technology/reprints.do?id=1-1CYG9N1&ct=121127&st=sb. Online; accessed 30-May-2013.

[240] D. Tilman, R. M. May, C. L. Lehman, and M. A. Nowak. Habitat destruction and the extinction debt. *Nature*, 1994.

[241] D. Triebel, W. Ahlmer, A. Bresinsky, O. Dürhammer, S. Jablonski, A. Kehl, D. Neubacher, P. Poschlod, G. Rambold, T. Schneider, B. Volz, and M. Weiss. Developing a sustainable working platform for gathering biological data in the field. In *Proceedings of the e-Biosphere 2009*, e-Biosphere 09, page 142. Anonymous, 2009.

[242] D. Triebel, G. Hagedorn, and G. Rambold. An appraisal of megascience platforms for biodiversity information. *MycoKeys*, 5:45–63, 2012.

[243] M. Türkay and G. Back. *Quantifizierungsmöglichkeiten*, volume 10, chapter 7. Springer, 2002.

[244] J. Ullman. Information integration using logical views. In F. Afrati and P. Kolaitis, editors, *Database Theory – ICDT '97*, volume 1186 of *Lecture Notes in Computer Science*, pages 19–40. Springer Berlin / Heidelberg, 1997.

[245] United Nations. Resolution 61/203: International year of biodiversity, 2010. http://daccess-dds-ny.un.org/doc/UNDOC/GEN/N06/506/55/PDF/N0650655.pdf?OpenElement, 2006. Online; accessed 16-August-2011.

[246] United Nations. Resolution 65/161: Convention on biological diversity. http://www.un.org/Depts/dhl/resguide/r65.shtml, 2010. Online; accessed 16-August-2011.

[247] D. Vieglais. Digir provider manual. http://digir.net/prov/prov_manual.html, 2003. Online; accessed 23-August-2012.

[248] J. Visser. Metrics for XML schema. In *XATA*, Portalegre, 2006.

[249] B. Volz. *Einstieg in Visual C# 2008*. Galileo-Computing, 2008.

[250] B. Volz. *Werkzeugunterstützung für methodenneutrale Metamodellierung*. PhD thesis, University of Bayreuth, Bayreuth, Germany, 2011.

[251] W3C. XML schema: Formal description. http://www.w3.org/TR/2001/WD-xmlschema-formal-20010925//, 2001. Online; accessed 11-July-2012.

[252] W3C. Web services architecture. http://www.w3.org/TR/2003/WD-ws-arch-20030514/\#id2608426, 2003. Online; accessed 10-October-2012.

[253] W3C. RDF/XML syntax specification (revised). http://www.w3.org/TR/REC-rdf-syntax/, 2004. Online; accessed 23-August-2012.

[254] W3C. Web services architecture. http://www.w3.org/TR/ws-arch/\#whatis, 2004. Online; accessed 10-October-2012.

[255] W3C. Web ontology language. http://www.w3.org/TR/owl2-overview/, 2009. Online; accessed 26-August-2012.

[256] W3C. W3C XML schema definition language (XSD). http://www.w3.org/TR/xmlschema11-1/, 2012. Online; accessed 11-July-2012.

[257] Y. Wand and R. Y. Wang. Anchoring data quality dimensions in ontological foundations. *Commun. ACM*, 39(11):86–95, Nov. 1996.

[258] Y. Wand and R. Weber. An ontological model of an information system. *Software Engineering, IEEE Transactions on*, 16(11):1282 –1292, nov 1990.

[259] Y. Wand and R. Weber. Research commentary: information systems and conceptual modeling-a research agenda. *Information Systems Research*, 13(4):363–376, 2002.

[260] T. Weibulat, D. Triebel, T. Schneider, R. W., B. Volz, M. Weiss, and G. Rambold. DiversityMobile: Recording and processing data in the field via smartphone. In *Proceedings of the BioSyst.EU 2013 Global systematics!*, BioSyst.EU '13, page 229. NOBIS Austria, 2013.

[261] M. Weiss, G. Hagedorn, and D. Triebel. DiversityCollection information model (version 2.05.17). http://diversityworkbench.net//Portal/CollectionModel_v2.05.17, 2012. Online; accessed 16-November-2012.

[262] J. Wieczorek. The gbif integrated publishing toolkit user manual, version 2.0. http://code.google.com/p/gbif-providertoolkit/wiki/IPT2ManualNotes, 2011. Online; accessed 16-October-2012.

[263] J. Wieczorek, M. Döring, R. De Giovanni, T. Robertson, and D. Vieglais. Darwin Core Terms: A quick reference guide. http://rs.tdwg.org/dwc/terms/, 2011. Online; accessed 11-March-2013.

[264] Wieczorek, J. and Döring, M. and De Giovanni, R. and Robertson, T. and Vieglais, D. Simple Darwin Core. http://rs.tdwg.org/dwc/terms/simple/index.htm, 2009. Online; accessed 29-July-2012.

[265] J. Wilkins. How many species concepts are there? http://www.guardian.co.uk/science/punctuated-equilibrium/2010/oct/20/3, 2010. Online; accessed 16-August-2011.

[266] C. Woese, O. Kandler, and M. Wheelis. Towards a natural system of organisms: proposal for the domains archaea, bacteria, and eucarya. *Proceedings of the National Academy of Sciences*, 87(12):4576, 1990.

[267] P. Ziegler and K. Dittrich. Three decades of data intecration - all problems solved? In R. Jacquart, editor, *Building the Information Society*, volume 156 of *IFIP International Federation for Information Processing*, pages 3–12. Springer Boston, 2004. 10.1007/978-1-4020-8157-6_1.

Lebenslauf

Tobias Schneider
Bessererstr.14
97422 Schweinfurt

Geburtstag: 06.03.1977
Geburtsort: Würzburg
Nationalität: deutsch

Akademische Ausbildung

05/2006: Diplom in Wirtschaftsmathematik an der Universität Bayreuth
09/2000: Ärztliche Vorprüfung an der Universität Göttingen
07/1996: Abitur am Celtis-Gymnasium in Schweinfurt

Erklärung zur Selbständigkeit

Hiermit erkläre ich eidesstattlich Folgendes:

- Ich habe die vorliegende Arbeit selbständig erstellt und keine außer den angegebenen Quellen und Hilfsmittel verwendet.
- Ich habe bisher keinen Promotionsversuch unternommen.
- Ich habe niemals die Dienstleistungen von gewerblichen Promotionsberatern oder ähnlichen Dienstleistern in Anspruch genommen und werde diese auch in Zukunft nicht in Anspruch nehmen.

Bayreuth, den 28.06.2013 Tobias Schneider