Universität Leipzig
Fakultät für Mathematik und Informatik
(Institut für Informatik)

Adaptierung von Ontologie-Mappings in den Lebenswissenschaften

Masterarbeit

Leipzig, 15. April 2014

vorgelegt von: Gassner, Michael
Studiengang: Informatik (Master)

Betreuer: **Prof. Dr. Erhard Rahm**
Dr. Anika Groß
Universität Leipzig, Institut für Informatik, Abt. Datenbanken

Inhalt

1. Einleitung

1.1. Motivation

Der Einsatz von Ontologien hat in den letzten Jahren, gerade in den Lebenswissenschaften, stark an Bedeutung zugenommen [1][2]. Durch die parallele Entwicklung von Ontologien mit verschiedenen Schwerpunkten gibt es teilweise mehrere Ontologien zum gleichen Themengebiet. So beinhaltet die Open Biomedical Ontologies (OBO) Foundry mehr als 30 Ontologien, mit dem Schwerpunkt Anatomie [3]. Diese Ontologien sind teilweise sehr groß, zum Beispiel Foundational Model of Anatomy (FMA), National Cancer Institute Thesaurus (NCIt) oder SNOMED Clinical Terms (SNOMED-CT). Um bestimmte Datenintegrationsaufgaben zu vereinfachen werden Mappings zwischen verschiedenen Ontologien benötigt. So sind Ontologiemappings zum Beispiel für das Zusammenführen (Mergen) mehrerer Ontologien in eine integrierte Ontologie/Datenquelle wie das Unified Medical Language System (UMLS) [4] notwendig. Der naive Ansatz zur Erstellung eines solchen Mappings ist die manuelle Zuordnung der einzelnen Konzepte

einer Ontologie zu den Konzepten der zweiten Ontologie. Dafür wird Expertenwissen auf dem jeweiligen Themengebiet benötigt. Zudem ist eine manuelle Erstellung sehr aufwendig und für besonders große Ontologien eventuell nicht realisierbar. Manuell erstellte Mappings haben zwar eine sehr hohe Qualität, verursachen aber mit zunehmendem Wachstum der Ontologien einen stetig wachsenden Aufwand an qualifiziertem Personal und Zeit. Daher steigt die Notwendigkeit von (semi)automatischen Matchingverfahren zur Erstellung von Mappings zwischen Ontologien [5][6][7].

Die Lebenswissenschaften sind ein wichtiges Forschungsgebiet. Daher gibt es regelmäßig neue Entdeckungen, Erkenntnisse und Forschungsergebnisse, wodurch das bestehende Wissen in Ontologien erweitert und angepasst werden muss. Diese Änderungen betreffen auch die Ontologien in den Lebenswissenschaften. Typischerweise erscheinen dort in regelmäßigen Abständen neue Versionen, um die Ontologien auf dem neusten Stand zu halten [8]. Diese Updates können in schneller Folge auftreten, so wird zum Beispiel Gene Ontology (GO) [9], eine der bekanntesten Ontologien der Biowissenschaften, jeden Tag aktualisiert. Die Änderungen bestehen hauptsächlich aus dem Hinzufügungen (add), Löschungen (del) oder Änderungen (update) der bestehenden Konzepte, deren Attribute und Relationen. Die Häufigkeit dieser Änderungen variiert dabei abhängig von der jeweiligen Ontologie und den verschiedenen Regionen innerhalb einer Ontologie [10][11]. Ontologieevolution kann dabei Einfluss auf verschiedene ontologiebasierte Mappings und Anwendungen haben. Dazu gehören die Mappings zwischen Ontologien [12][13], Annotationsmappings [14][15] oder ontologiebasierte Anfragen [16][17]. Ontologiemappings können ihre Aktualität und Gültigkeit infolge der Veröffentlichung neuer Ontologieversionen verlieren und müssen angepasst werden. Wenn zum Beispiel eine neue Version einer Ontologie in Bioportal [18] oder UMLS hinzugefügt wird, dann müssen die betroffenen Mappings adaptiert werden, damit die Anwender und Anwendungen auf der aktuell gültigen Version arbeiten können.

Da in die bestehenden, meist manuell erstellten, Mappings sehr viel Expertenwissen und Arbeitszeit investiert wurde, ist es von zunehmendem Interesse, dass die Ergebnisse der alten Mappings wiederverwendet werden können, damit diese nicht mit jeder neuen Ontologieversion komplett obsolet werden und neu bestimmt werden müssen. An dieser Stelle setzen (semi-)automatische Mapping-Adaptierungsverfahren an, die auf Basis des alten Mappings, der alten und neuen Ontologieversion ein aktuelles Mapping erstellen.

Das Ergebnis ist im Optimalfall ein vollständiges aktuelles Mapping zwischen den beiden aktuellen Ontologieversionen. Expertenwissen wird in diesen Fällen nur noch bei der Entscheidung komplizierter Fälle benötigt, die während der automatischen Adaptierung gesammelt und dem Anwender anschließend präsentiert werden. Alternativ kann der Nutzer bei einem semi-automatischen Verfahren direkt gefragt und das Wissen einbezogen werden.

Um solche (semi-)automatischen Verfahren (z.Bsp. Matchalgorithmen) zu testen werden Benchmarks benötigt. Insbesondere wichtig ist die Qualität der erzeugten Mappings zu testen. Ein Benchmark zur Untersuchung der Mappingqualität besteht aus mindestens 2 Ontologien und einem manuell erstellten Referenzmapping zwischen diesen Ontologien, dem sogenannten Goldstandard. Ziel ist es, dass das z.Bsp. von einem Matchingalgorithmus erstellte Mapping möglichst genau (hohe Genauigkeit, Precision) und möglichst vollständig (hohe Abdeckung, Recall) ist. Die Ontology Alignment Evaluation Initiative (OAEI)[1] bietet zum Beispiel Benchmarks für Ontologie-Matching-Probleme aus unterschiedlichen Domänen. Um eine Evaluierung von Mapping-Adaptierungsverfahren infolge von Ontologieevolution zu ermöglichen, werden jedoch mehrere (mindestens 2) Versionen eines Mappings benötigt.

Bisher gibt es noch relativ wenig Forschungsergebnisse auf dem Gebiet der (semi-)automatischen Mapping-Adaptierung infolge von Ontologieevolution und wie diese optimal umgesetzt werden kann. Da bisher auch ein entsprechender Benchmark fehlte konnten in der Vergangenheit nur schwer Angaben zur Qualität der verschiedenen Adaptierungsverfahren gemacht werden.

1 http://oaei.ontologymatching.org/

1.2. Ziele der Arbeit

Ziel dieser Arbeit ist die Untersuchung der Adaptierung von Ontologiemappings infolge von Ontologieevolution in den Lebenswissenschaften. Dafür sollen folgende Kernziele umgesetzt werden:

- Erstellung eines Benchmarks unter Verwendung der drei sehr großen Ontologien der Lebenswissenschaften (NCI Thesaurus, Foundational Model of Anatomy, SNOMED Clinical Terms)
- Anpassung von bestehenden und Erstellung neuer Adaptierungsstrategien
- Evaluierung der Adaptierungsstrategien anhand des erstellten Benchmarks

Um den Benchmark zu erstellen sollen die sehr großen Ontologien NCI Thesaurus, Foundational Model of Anatomy und SNOMED Clinical Terms aus UMLS extrahiert und in ein Repository übertragen werden. Anhand der Datenstruktur von UMLS sollen die vorhandenen Mappings extrahiert und zusammen mit den Ontologien in ein GOMMA-Repository übertragen werden. Ziel ist es, erstmals einen Benchmark für verschiedene Mappingversionen unter Verwendung unterschiedlicher Ontologieversionen zur Verfügung zu stellen.

Anschließend sollen bestehende Adaptierungsstrategien vorgestellt, an bestimmte Szenarien angepasst und eine neue Strategie zur Behandlung verschiedener Änderungsoperationen vorgestellt werden. Dabei sollen insbesondere die komplexen Änderungsoperationen *split* zum Aufspalten eines Konzepts in mehrere Konzepte sowie Änderungen der Konzeptattribute betrachtet werden. Abschließend soll eine Evaluierung erfolgen, in der die Auswirkungen und Effektivität der vorgestellten Adaptierungsstrategien anhand des erstellten Benchmarks untersucht werden.

1.3. Gliederung der Arbeit

Der Rest der Arbeit gliedert sich wie folgt:
Kapitel 2 gibt einen Überblick und Hintergrundinformationen über die Grundlagen, die für diese Arbeit wichtig sind. Es wird der Begriff Ontologie erklärt, Modelle für Ontologie, Ontologieevolution und Mappingevolution eingeführt, sowie verwandte Arbeiten vorgestellt.

Kapitel 3 stellt die einzelnen Schritte zur Erstellung des Benchmarks vor. Es werden das Unified Medical Language System (UMLS) und die für den Benchmark verwendeten Ontologien vorgestellt. Dabei wird näher darauf eingegangen, wie die Informationen in UMLS gespeichert werden und wie die benötigten Informationen extrahiert wurden. In Kapitel 4 werden verschiedene Adaptierungsstrategien vorgestellt. In Kapitel 5 erfolgt eine Evaluierung der vorgestellten Adaptierungsstrategien anhand des erstellten Benchmarks.

2. Grundlagen und verwandte Arbeiten

Dieses Kapitel gibt einige wichtige Informationen zu den Grundlagen dieser Arbeit. Es wird der Begriff Ontologie definiert und ein Modell für die Beschreibung von Ontologien, Ontologieevolution und zur Adaptierung der Mappings eingeführt. Danach folgt eine Übersicht über den aktuellen Stand der Forschung auf diesem Gebiet indem verwandte Arbeiten vorgestellt werden.

2.1. Ontologien

“An ontology is an explicit, formal specification of a shared conceptualization. The term is borrowed from philosophy, where an ontology is a systematic account of Existence. For knowledge-based systems, what “exists” is exactly that which can be represented.” *(Thomas R. Gruber, 1993) [19]*

Eine Ontologie ist nach dieser Definition eine explizite, formale Spezifikation einer gemeinsamen Konzeptualisierung. Es handelt sich um ein abstraktes Modell (Konzeptualisierung) einer Domäne mit identifizierten relevanten Begriffen (Konzepte) und Beziehungen zwischen diesen Begriffen. Die Bedeutungen aller Begriffe sind definiert (explizit), das Modell ist für Maschinen lesbar (formal) und es herrscht Konsens bezüglich der Ontologie (gemeinsam). Konzepte und Beziehungen können durch verschiedene Attribute näher definiert oder beschrieben werden, z.Bsp. Namen, Synonyme oder Definitionen. Ontologien sind Werkzeuge zur Organisation von Informationen und sie werden genutzt, um das Wissen über bestimmte Fachgebiete zu repräsentieren. Diese können dabei breit gefächert oder sehr stark spezialisiert sein. Die Art der Wissensrepräsentation ermöglicht die Interpretation und Analyse der Informationen durch Mensch und Maschine gleichermaßen. Die Hauptanwendung von Ontologien ist daher die Annotation von Realweltobjekten (z.Bsp. Patientenakten, Gene, Literaturquellen) und somit Unterstützung der semantischen Suche auf diesen Objekten.

In dieser Arbeit wird das folgende Modell für eine Ontologie verwendet. Eine Ontologie $O=(C, A, R)$ besteht aus den Konzepten $c \in C$, denen Attribute $a=(a_{concept}, a_{name}, a_{value}) \in A$ zugeordnet sind und die über Relationen $r=(r_{source}, r_{type}, r_{target}) \in R$ miteinander in Beziehung stehen. Ein Attribut ist einem Konzept $a_{concept}$ zugeordnet und hat eine Bezeichnung a_{name} sowie einen Wert a_{value}. Jedes Konzept wird eindeutig über eine *accession number* identifiziert. Attribute beschreiben das Konzept detaillierter, zum Beispiel durch Synonyme und eine Definition. Ein domänenspezifisches Attribute ist *to obsolete*. Es gibt an, ob ein Konzept aktuell oder veraltet ist und deshalb nicht mehr genutzt werden sollte. Beziehungen haben immer eine Quelle, ein Ziel und einen Typ, der die Relation näher definiert, z.Bsp. ermöglichen es is-a-Beziehungen Vererbungshierarchien auszudrücken. Dabei werden alle Eigenschaften an das erbende Element weitergegeben. Part-of-Beziehungen verdeutlichen, dass das

Quellkonzept ein Teil des Zielkonzepts ist.

Abbildung 1 zeigt ein Beispiel für eine kleine Ontologie O. Die blauen Rechtecke sind die Konzepte, die durchgezogenen Pfeile sind part-of-Beziehungen und die gestrichelten Pfeile sind is-a-Beziehungen. So sind im Beispiel unter anderem *Menschlicher Körper*, *Kopf*, *Rumpf* und *Gliedmaßen* Konzepte der Ontologie.

$$(Menschlicher\ Körper\ ,\ Kopf\ ,\ Rumpf\ ,\ Gliedmaßen) \in C$$

Dabei sind *Kopf*, *Rumpf* und *Gliedmaßen* Teile von *Menschlicher Körper*, was durch die Relationen part-of zwischen den Konzepten dargestellt wird.

$$((Kopf\ ,\ part-of\ ,\ Menschlicher\ Körper),(Rumpf\ ,\ part-of\ ,\ Menschlicher\ Körper),\\ (Gliedmaßen,\ part-of\ ,\ Menschlicher\ Körper)) \in R$$

Bei *Arm* und *Bein* handelt es sich um *Gliedmaßen*, d.h. diese sind durch eine is-a-Beziehung mit Gliedmaßen verbunden.

$$((Arm\ ,\ is-a\ ,\ Gliedmaßen),(Bein\ ,\ is-a\ ,\ Gliedmaßen)) \in R$$

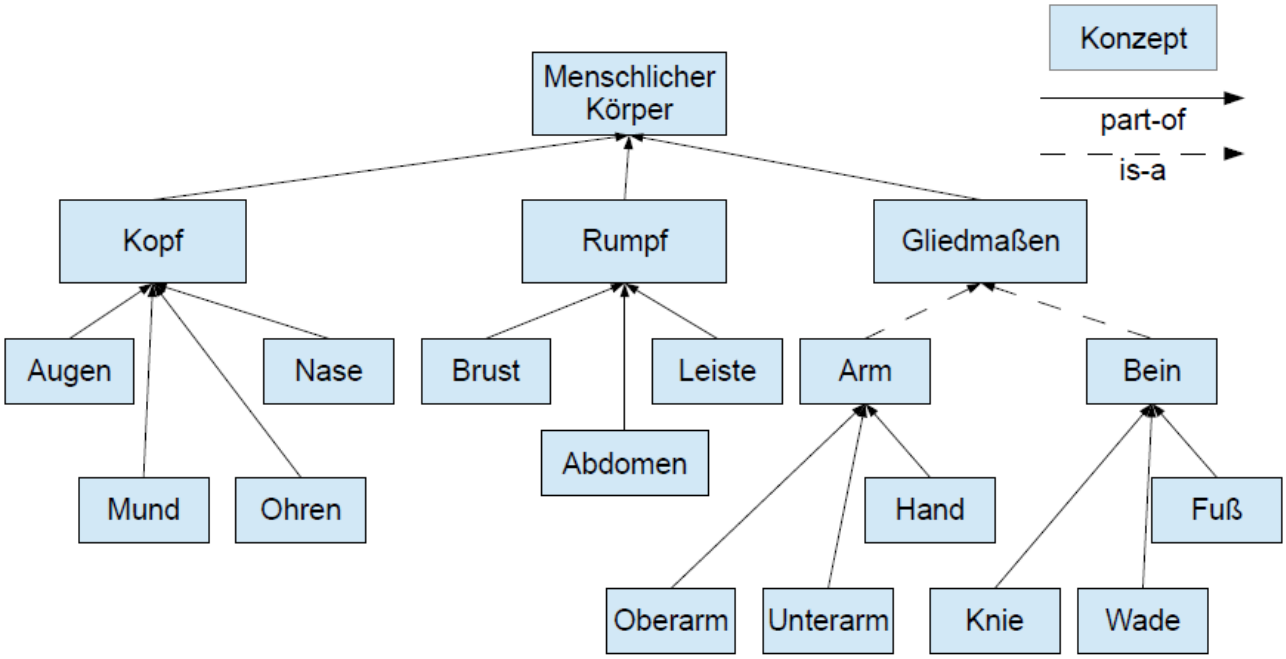

Abbildung 1: Beispiel für eine Ontologie

Eine Ontologieversion $O_v = (C_v, R_v, A_v)$ ist eine Veröffentlichung einer Ontologie O. Diese ist so lange gültig, bis eine aktuellere Version der Ontologie $O_{v'}$ veröffentlicht wird. In dieser Arbeit wird die alte Version einer Ontologie mit O angegeben und die neuere

Version mit O'.

2.2. Ontologiemapping

Das hier beschriebene Modell orientiert sich an dem in [20] vorgestellten Model. Ein Ontologiemapping M_{O_1,O_2} verbindet die Konzepte zweier verschiedener Ontologien O_1/O_2 durch sogenannte Korrespondenzen:

$$M_{O_1,O_2}=\{(c_1,c_2,\mathrm{sim},semType,status)\,|\,c_1\in O_1,c_2\in O_2,\mathrm{sim}\in[0,1],semType\in\{=,\leq,\geq,\approx\},\\ status\in\{'handled','verifiy','unhandled'\}$$

Eine Korrespondenz $(c_1,c_2,\mathrm{sim},semType,status)$ verknüpft 2 Konzepte $c_1\in O_1$ und $c_2\in O_2$. Mit den Attributen *sim*, *semType* und *status* kann die Korrespondenz detaillierter beschrieben werden. Der Wert *sim* ist ein Maß für die Ähnlichkeit zwischen zwei Konzepten c_1 und c_2. Je höher der Wert ist, desto ähnlicher sind sich beide Konzepte. Bei einer manuell erstellten Korrespondenz wird der Wert auf 1 festgelegt. Mit *semType* werden verschiedene semantische Beziehungen zwischen Konzepten unterschieden. So können zwei Konzepte gleich (*equal*, z.Bsp. Hirn = Gehirn), allgemeiner oder spezialisierter sein (*more or less general,* z.Bsp. Arm ≤ Gliedmaßen) oder irgendwie mit einander verbunden sein(≈). Eine Korrespondenz kann erfolgreich abgearbeitet (*handled*) sein, muss noch durch einen Experten verifiziert werden (*to verify*) oder kann unbehandelt (*unhandled*) bleiben.

2.3. Ontologieevolution

Ontologien werden regelmäßig aktualisiert und angepasst. Gründe für die ständige Weiterentwicklung (Evolution) von Ontologien sind unter anderem neue Erkenntnisse, zum Beispiel durch Forschung, in einer Domäne. Sehr häufig werden die Ontologien dafür um neue Konzepte und Beziehungen erweitert. Ebenso werden veraltete Konzepte und Beziehungen gelöscht. Durch diese Operationen entstehen verschiedene Versionen der Ontologien, die sich von Version zu Version weiterentwickeln. Deshalb spricht man dabei von Ontologieevolution.

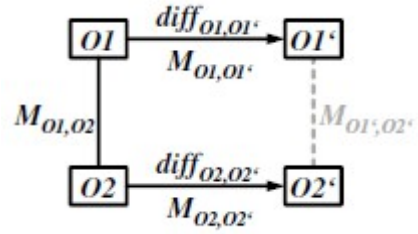

Abbildung 2: Ontologie- und Mappingevolution (aus [20])

Die Ausgangslage der in dieser Arbeit untersuchten Szenarien ist in *Abbildung 2* dargestellt. Es liegen 2 Ontologien jeweils in einer alten $(O1, O2)$ und einer neuen $(O1', O2')$ Version vor. Es existiert ein Mapping $M_{O1,O2}$ zwischen den beiden alten Versionen der beiden Ontologien. Die Aufgabe ist es das Mapping $M_{O1',O2'}$ zu bestimmen, das die beiden neuen Versionen der Ontologien verbindet. Für diese Aufgabe werden weitere Mappings zwischen den Ontologieversionen benötigt. Je nach Herangehensweise werden diese Mappings unterschiedlich generiert. [20] schlägt dazu zwei mögliche Ansätze vor:

- kompositionsbasierte Mappingadaption und
- diff-basierte Mappingadaption.

Bei der kompositionsbasierten Mappingadaption werden die Mappings $M_{O1,O1'}$ und $M_{O2,O2'}$ erstellt indem die beiden Ontologieversionen miteinander gematcht werden. Diese Mappings beinhalten Informationen über die Beziehungen zwischen den Konzepten in der alten und der neuen Version. Bei einem kompositionsbasierten Ansatz werden die Mappings $M_{O1,O2}$, $M_{O1,O1'}$, $M_{O2,O2'}$ genutzt um das Mapping $M_{O1',O2'}$ zu erstellen.

Änderungsoperation	Beschreibung
$addC(c), delC(c)$	Hinzufügen/Löschen eines Konzepts c
$toObsolete(c), revolveObsolete(c)$	Setzen/Aufheben des Status "veraltet" eines Konzepts c
$substitute(c, c')$	Ersetzen eines Konzepts c durch ein anderes Konzept c'
$split(s, T)$	Aufspalten eines Quellkonzepts s in mehrere Zielkonzepte T
$merge(S, t)$	Zusammenfassen mehrerer Quellkonzepte S zu einem Zielkonzept t
$addSubGraph(c_{root}, C_{\text{Sub}})$	Einfügen eines Subgraphen mit Wurzel c_root und den Konzepten C_Sub
$delSubGraph(c_{root}, C_{\text{Sub}})$	Löschen eines Subgraphen mit Wurzel c_root und den Konzepten C_Sub
$addR(r), delR(r)$	Hinzufügen/Löschen einer Beziehung r
$move(c, P, P')$	Verschieben eines Konzepts c von seinen Elternkonzepten P zu anderen Elternkonzepten P'
$addA(a), delA(a)$	Hinzufügen/Löschen eines Attributs a
$chgAttValue(c, att, v_{1,} v_2)$	Ändern des Attributwertes v1 von Attribut att des Konzepts c zu Wert v2

Tabelle 1: Änderungsoperationen in Conto-Diff (nach [20])

Bei einer diff-basierten Mappingadaption werden sogenannte Evolutionsmappings $diff_{O1,O1'}$, $diff_{O2,O2'}$ zwischen den Ontologieversionen erstellt. Diese beinhalten alle Änderungen, die bei der Evolution der Ontologien $(O1\,zu\,O1'\,und\,O2\,zu\,O2')$ stattgefunden haben. Ein solches Evolutionsmapping kann mit einem Diff-Werkzeug wie PromptDiff [21] oder COnto-Diff [22] erstellt werden und dokumentiert verschiedene Arten von Änderungen. *Tabelle 1* beinhaltet einige Änderungsoperationen, die von COnto-Diff erkannt werden können, welches in dieser Arbeit zum Einsatz kam. So gibt $split(s, T)$ die Aufspaltung eines Quellkonzepts *s* in mehrere Zielkonzepte *T* in der neuen Version an. Das Ändern eines Attributs wird durch $chgAttValue(c, att, v_{1,} v_2)$ angegeben. Der Attributwert *v1* eines Attributs *att* des Konzepts *c* wird in der neuen Version durch den Attributwert *v2* ersetzt.

2.4. Vewandte Arbeiten

Während die Evolution von Ontologien regelmäßig im Fokus der Forschung steht und entsprechend viele Arbeiten dazu veröffentlicht wurden (siehe [23] für einen Überblick) ist die Evolution der abhängigen Mappings bisher wenig erforscht. Ein kompositionsbasierter Ansatz für Schema Mappings wurde in [24] untersucht und vorgestellt um die komplette Neuberechnung der existierenden Mappings zu vermeiden. In dieser Arbeit wird ein diff-basierter Ansatz untersucht, der mit Hilfe von Evolutionsmappings eine komplette Neuberechnung vermeidet, indem nur Korrespondenzen neu berechnet werden, deren Konzepte von der Ontologieevolution betroffen sind. Die Abteilung Datenbanken der Universität Leipzig hat in [13] untersucht welche Änderungen an den Ontologien zum Löschen und Hinzufügen von Korrespondenzen führen. Dos Reis et al. [12] haben ein erstes Framework vorgestellt, dass den Einfluss verschiedener Ontologieänderungen auf Mappingadaptierungen untersucht. Darüber hinaus wurde gezeigt, dass die Verwendung verschiedener semantischer Typen für die Korrespondenzen wichtig für die Mappingadaptierung ist. Diese beiden Arbeiten bilden mit ihren Ergebnissen eine Grundlage für den diff-basierten Ansatz, der in dieser Arbeit zum Einsatz kommt. Dieser diff-basierte Ansatz wurde in [20] vorgestellt und es wurden bereits verschiedene Mappingstrategien evaluiert. Allerdings wurden keine Attributänderungen betrachtet, nur ein naiver Ansatz genannt. Für Split wurden bereits 2 Strategien untersucht, aber es ist sinnvoll noch andere Strategien oder Erweiterungen der bestehenden Strategien zu untersuchen.

Die Idee UMLS als integrierte Datenquelle zu nutzen, um verschiedene Ontologien zusammen mit den Mappings zwischen diesen Ontologien zu extrahieren wurde im Kontext des Ontologie-Matchings von Jiménez-Ruiz et al. [25] vorgeschlagen. Dabei wurden Ontologien aus der Version 2009AA extrahiert und die durch UMLS erstellten Mappings auf ihre Qualität hin untersucht. Bisher existiert kein Benchmark für verschiedene Mappingversionen um Adaptierungsstrategien zu evaluieren. Ein solcher Benchmark soll im Rahmen dieser Arbeit erstellt werden.

3. Erstellung eines Benchmarks für Versionen von Ontologiemappings

Dieses Kapitel behandelt die Erstellung des Benchmarks für mehrere Versionen von Ontologiemappings. Es folgen Informationen und Hintergrundwissen zu der integrierten Datenquelle UMLS und den extrahierten Ontologien. Es wird beschrieben wie mehrere Versionen der ausgewählten Ontologien extrahiert und in ein GOMMA-Repository importiert wurden. Der letzte Schritt war die Extraktion der Mappings zwischen den einzelnen Ontologien der gleichen Version.

3.1. Unified Medical Language System

Das Unified Medical Language System (UMLS) [4] wurde 1989 von der US National Library of Medicine geschaffen und stellt eine Sammlung ausgewählter Vokabulare der biomedizinischen Wissenschaften dar. Aktuell umfasst UMLS über 100 Ontologien der Biomedizin und bietet somit eine mächtige integrierte Datenquelle. Dazu zählen unter anderen Medical Subject Headings (MeSH), International Statistical Classification of Diseases and Related Health Problems (ICD-10), Logical Observation Identifiers Names and Codes (LOINC), sowie die für diese Arbeit relevanten Foundational Model of Anatomy (FMA), NCI Thesaurus des National Cancer Institute (NCIt) und SNOMED – Clinical Terms (SNOMED-CT).

UMLS bietet die folgenden drei Werkzeuge an:

- Metathesaurus: Begriffe und Codes (ID in der Quelle) aus vielen verschiedenen Vokabularen
- Semantic Network: allgemeine Kategorien (semantische Typen) und ihre Beziehungen (semantische Beziehungen)
- SPECIALIST Lexicon and Lexical Tools: Werkzeuge zur Verarbeitung natürlicher Sprache

Der Metathesaurus wird mit Hilfe des Semantic Networks und der Lexical Tools erstellt. Dies geschieht in den folgenden Schritten:

- Verarbeitung der Begriffe und Codes mit Lexical Tools
- Synonyme werden zu Konzepten zusammengeführt
- Kategorisierung der Konzepte nach semantischen Typen mit Hilfe des Semantic Networks
- Einfügen der Beziehungen und Attribute, die aus den Quellvokabularen stammen
- Veröffentlichung der Informationen in einem gut zu verarbeitendem Format

Die für die Erstellung des Metathesaurus genutzten Werkzeuge sind frei zugänglich und können in Kombination oder unabhängig voneinander für eigene Arbeiten genutzt werden.

Der Zugriff auf UMLS ist dabei auf 3 verschiedenen Wegen möglich und wird über die

UMLS Technology Services (UTS) bereitgestellt. Die erste Möglichkeit ist mit einem einfachen Webbrowser. UTS bieten dafür einen Metathesaurus Browser und einen Semantic Network Browser an. Mit dem Metathesaurus Browser ist es möglich Informationen, einschließlich einer eindeutigen internen ID, semantischem Typ und Synonymen, zu den Konzepten in UMLS abzurufen. Der Semantic Network Browser zeigt Namen, Definitionen und hierarchische Strukturen des Semantic Network an. Die zweite Möglichkeit ist eine lokale Installation von UMLS. Dafür werden Dateien zum Download bereitgestellt. Aus diesen können die gewünschten Daten mit Hilfe eines Tools (MetamorphoSys) extrahiert und in einer eigenen Datenbank gespeichert werden. Alternativ ist der Zugriff direkt auf die Dateien mit dem MetamorphoSys RFF-Browser möglich. Die dritte Möglichkeit ist der direkte Zugriff mit einer eigenen Applikation via Web Services Application Programming Interfaces (APIs) auf den Servern von UMLS. Voraussetzung für die Nutzung von UMLS ist eine Registrierung auf der Webseite. Die Lizenz ist für akademische Nutzer kostenfrei.

3.2. Extraktion der Ontologien

Um die Ziele dieser Arbeit umzusetzen, wurden die gewünschten Ontologien aus der integrierten Datenquelle UMLS extrahiert und in einer relationalen Datenbank auf einem Server der Abteilung Datenbanken der Universität Leipzig gespeichert. Die dafür benötigten Dateien im nlm-Format wurden aus dem UMLS-Portal herunter geladen und auf der lokalen Festplatte gespeichert. Der Zugriff auf die Daten ermöglicht das ebenfalls bereitgestellte Programm MetamorphoSys. Nach dem Ausführen des Programms hat der Anwender die Wahl ob es als Browser für bereits extrahierte (in MetamorphoSys „installierte“ genannt) Ontologien genutzt werden soll oder Ontologien ausgewählt und extrahiert werden sollen. Um eine oder mehrere Ontologien aus dem UMLS-Datensatz zu extrahieren muss der Nutzer „Install UMLS“ auswählen. Als nächster Schritt erfolgt die Auswahl der gewünschten Werkzeuge, die für die Ontologien extrahiert werden sollen (Metathesaurus, Sematic Network, SPECIALIST Lexicon and Lexical Tools), ob ein Script für den Import in eine MySQL – oder Oracledatenbank erstellt werden soll und ein Quell- sowie Zielverzeichnis werden abgefragt. Als nächstes muss für die Extraktion eine sogenannte Konfiguration für MetamorphoSys angelegt werden. In dieser erfolgt die Auswahl der gewünschten Ontologien und es sind verschiedene Einstellungen für das

Ausgabeformat verfügbar. Im Rahmen dieser Arbeit wurden die Standardeinstellungen verwendet um die 3 großen Ontologien der Lebenswissenschaften NCIt, FMA und SNOMED-CT zu extrahieren.

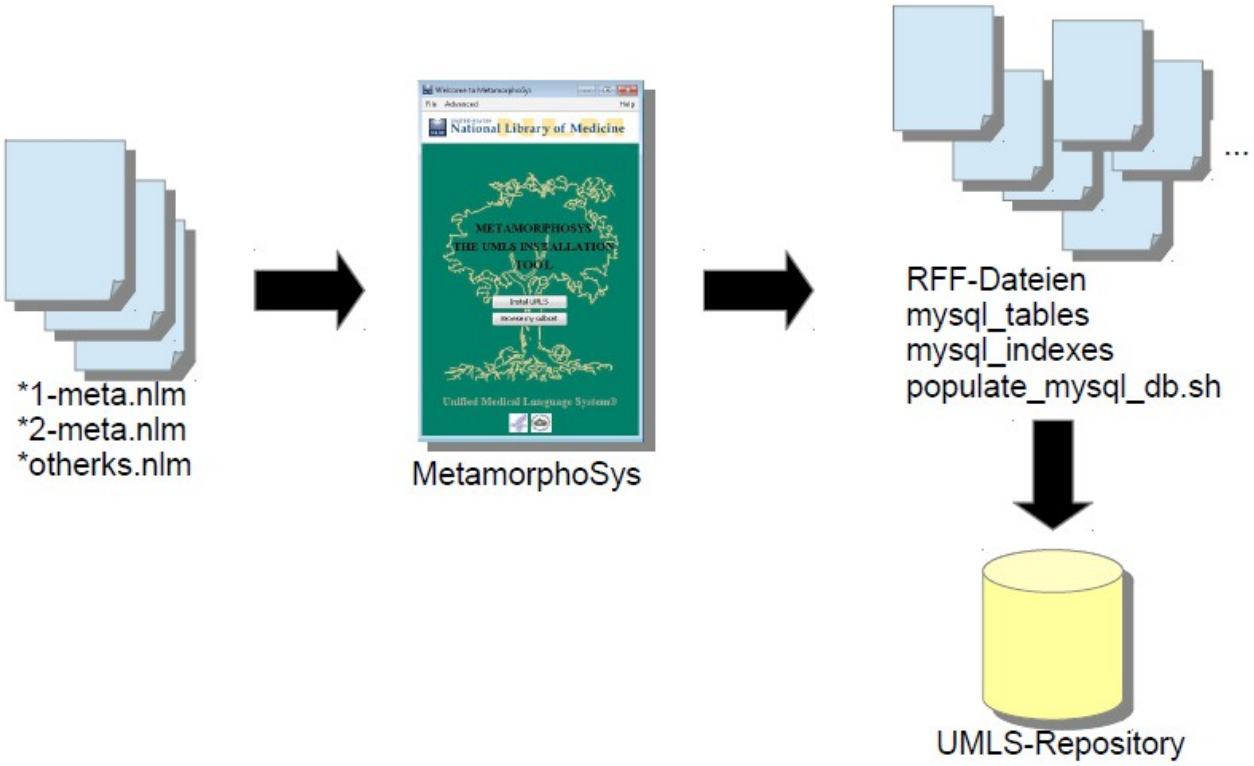

Abbildung 3: Überblick über den Ablauf der Extraktion der Ontologien

Das Ergebnis des Programmdurchlaufs sind RFF-Dateien, in denen die Informationen gespeichert sind, die später in die Tabellen der Datenbank gespeichert werden bzw. auf die MetamorphoSys in der Funktion als Ontologiebrowser Zugriff ermöglicht. Alternativ dazu werden, bei entsprechender Auswahl, Dateien zur Verfügung gestellt um die extrahierten Ontologien in eine eigene Datenbank zu übertragen. Die Dateien mysql_tables und mysql_indexes enthalten die SQL-Befehle zum Erstellen des UMLS-Repositorys. Mit der Datei populate_mysql_db.sh wird das Repository erstellt und die Daten werden aus den RFF-Dateien in die Datenbank importiert. Dafür muss diese Datei noch editiert werden um den Pfad der Datenbank, den Namen des Repositorys, Nutzernamen und Passwort anzugeben.

Für diese Arbeit sollte ein Benchmark mit mehreren Versionen erstellt werden. Deshalb musste neben verschiedenen Ontologien auch mehrere Versionen der Ontologien

extrahiert werden. Dabei ist der Veröffentlichungszyklus von UMLS sehr hilfreich gewesen. Ontologien haben im Allgemeinen alle einen unterschiedlichen Veröffentlichungszyklus und es ist schwer von mehreren Ontologien Versionen zu finden, die zeitlich genau zusammen passen. Von UMLS als Sammlung mehrerer integrierter Ontologien werden jährlich 2 Versionen veröffentlicht, jeweils eine im Frühjahr (AA) und eine im Herbst (AB). Den Versionen des Benchmarks liegt diese Versionierung zugrunde. Die folgenden Ontologien wurden jeweils in den Versionen 2009AA, 2010AA, 2011AA und 2012AA extrahiert.

a) Foundational Model of Anatomy

Das Foundational Model of Anatomy (FMA)[2] wird von der School of Medicine der University of Washington erstellt, gewartet und angeboten. FMA steht frei zur Verfügung und kann ohne Einschränkung kostenlos genutzt werden. Es beinhaltet derzeit ungefähr 75.000 Konzepte, 115.000 medizinische Begriffe und über 2,1 Millionen Beziehungen zwischen den Konzepten, bei 168 verschiedenen Beziehungsarten. FMA soll als Wissensquelle für die medizinische Informatik und Bioinformatik dienen. Die Ontologie repräsentiert die phänotypische Struktur des menschlichen Körpers, die für den Menschen verständlich ist. Aber auch maschinelle Systeme können auf der Ontologie navigieren, sie lesen und interpretieren. FMA ist zwar auf den menschlichen Körper spezialisiert, kann aber für andere Spezies angepasst und erweitert werden.

b) NCI Thesaurus

Der NCI Thesaurus (NCIt)[3] wird vom National Cancer Institute erstellt und publiziert. Er bietet ein Vokabular mit einer is-a Hierarchie für klinische Versorgung, translationale und Basisforschung und administrative Aufgaben in der Biomedizin. Der Schwerpunkt von NCIt liegt auf der Krebsforschung. Aktuell (Stand 2013) umfasst der NCI Thesaurus ca. 100.000 Konzepte und 200.000 Beziehungen zwischen den Konzepten. Ein Konzept besteht aus einer festen, eindeutigen ID, der bevorzugten Bezeichnung, Synonymen und der Definition. Es werden regelmäßige Aktualisierungen durch Experten auf den entsprechenden Gebieten vorgenommen. Der NCI Thesaurus wird kostenlos für

2 http://sig.biostr.washington.edu/projects/fm/index.html
3 http://ncit.nci.nih.gov/

kommerzielle und nicht-kommerzielle Anwendung zum Download zur Verfügung gestellt.

c) SNOMED – Clinical Terms

SNOMED – Clinical Terms (SNOMED-CT)[4] befindet sich im Besitz der International Health Terminology Standards Development Organisation (IHTSDO)[5] in Kopenhagen und wird von dieser regelmäßig aktualisiert und vertrieben. Der IHTSDO gehören 24 Länder an (Stand April 2014). Es ist eine Ontologie mit medizinischen Begriffen, die als Konzepte hinterlegt sind und Beziehungen untereinander besitzen. SNOMED-CT bietet Zugriff auf einheitliche ID's, Bezeichnung, Synonyme und Definitionen für diese an. Es werden Krankheiten, Diagnosen, Behandlungen, Mikroorganismen, Substanzen und andere für die Medizin relevanten Objekte abgedeckt. Ziel ist es ein für den internationalen Gebrauch einheitliches, möglichst vollständiges, Vokabular zu erstellen, dass als Standard in Krankenhäusern, Forschung und Schriftverkehr zum Einsatz kommt um die Kommunikation zwischen verschiedenen Einrichtungen zu verbessern. Jedes Mitglied der International Health Terminology Standards Development Organisation ist mitverantwortlich für die Verfügbarkeit, Aktualität und Qualität von SNOMED-CT.

Die Ontologie umfasst ca. 400.000 Konzepte und 600.000 Beziehungen zwischen den Konzepten (Stand 2012). Für die Nutzung wird eine Lizenz benötigt. Es ist möglich über UMLS als akademischer Nutzer Zugriff auf SNOMED CT zu erhalten.

3.3. Import in GOMMA-Repository

Im UMLS-Repository sind alle Informationen enthalten, die durch UMLS zur Verfügung gestellt werden. Diese Daten sind über 48 Tabellen verteilt und für diese Arbeit wird nur ein Teil davon benötigt. Von den Ontologien wurden die Konzepte mit ID, Name, ob das Konzept obsolet ist und, sofern vorhanden, Synonymen und Definitionen sowie die Beziehungen mit semantischem Typ extrahiert. Diese Informationen befinden sich in den Tabellen MRCONSO, MRDEF, MRREL und MRHIER des UMLS-Repositorys.

MRCONSO enthält alle Informationen zu den Konzepten in UMLS, nur die Definition muss zusätzlich aus MRDEF abgerufen werden. In den Tabellen MRREL und MRHIER sind die Beziehungen zwischen den Konzepten gespeichert. Um die Daten aus dem UMLS-

4 http://www.ihtsdo.org/snomed-ct/
5 http://www.ihtsdo.org/about-ihtsdo/

Repository in ein GOMMA-Repository zu übertragen, damit diese mit den Funktionen, die GOMMA bietet bearbeitet werden können, wurde ein Programm in Java realisiert. Die Eingabedaten sind der Name der gewünschten Ontologie (Dabei gilt zu beachten, dass FMA in UMLS als UWDA bezeichnet wird.) sowie die Version, die extrahiert werden soll. Die Ausgabe der extrahierten Information erfolgt in einer obo-Datei., die dann vom obo-Importer von GOMMA geparst wird, um die Daten in das GOMMA-Repository zu übertragen. *Abbildung 4* zeigt einen Überblick über diesen Vorgang.

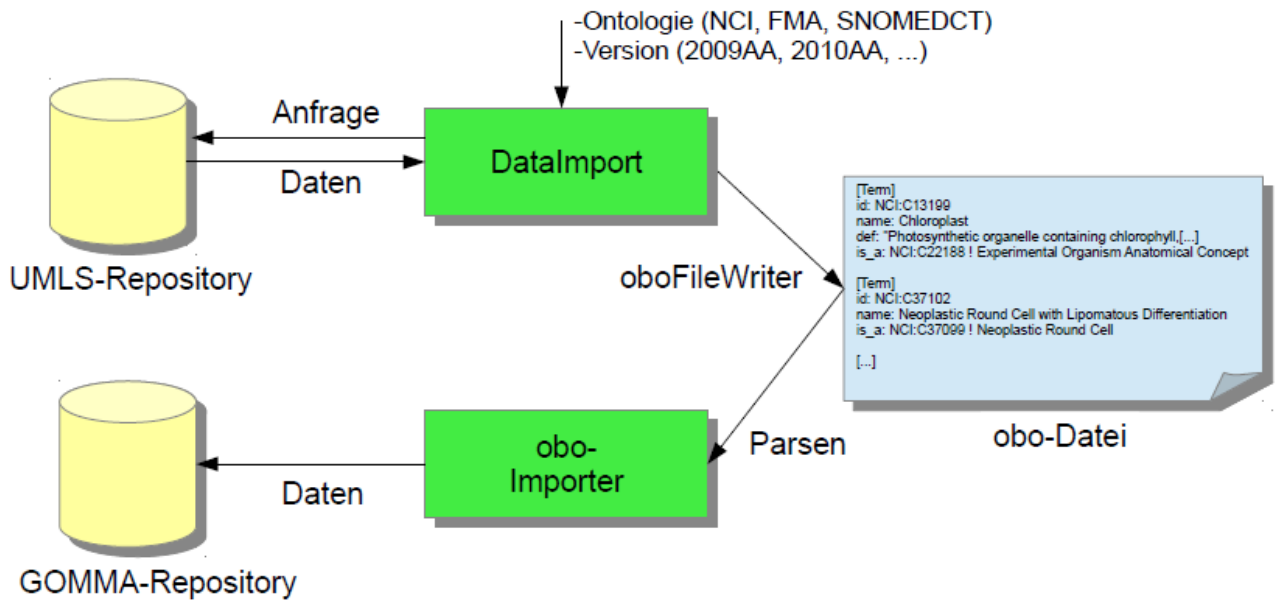

Abbildung 4: Import der Ontologiedaten in das GOMMA-Repository

Die Klasse *DataImport* greift über einen jdbc-Treiber auf das *UMLS-Repository*, das in einer MySQL-Datenbank realisiert wurde, zu. Mit Hilfe von prepared Statements werden erst alle benötigten Informationen (ID, Name, Synonyme, Definitionen) zu den Konzepten abgerufen und in einer Klasse *Data* hinterlegt, die als Datencontainer dient. Dabei wird für jedes Konzept ein einzelnes Objekt angelegt. Diese werden dann in einer Hashmap gespeichert, wobei die eindeutige ID als Schlüssel dient. Sind alle Konzepte abgerufen und lokal in der Hashmap verfügbar, wird über die einzelnen Elemente iteriert und die Beziehungen und Definition für jedes Konzept werden abgerufen und ebenfalls in der Klasse *Data* des Konzepts gespeichert. Als letzter Schritt wird ein zweites Mal über alle Elemente iteriert und mit Hilfe der Klasse *oboFileWriter* eine *obo-Datei* erstellt, die die extrahierte Ontologie enthält. Der *oboImporter* liest die Informationen aus der Datei aus und überträgt diese an das *GOMMA-Repository*.

3.4. Extraktion der Mappings

Das aus der Integration der verschiedenen Ontologien in die Datenquelle UMLS resultierende Datenformat vereinfacht die Extraktion der Mappings zwischen verschiedenen Ontologien. Für das Mapping sind die Informationen, die in der Tabelle MRCONSO gespeichert sind von Interesse. *Abbildung 5* zeigt wie die für das Mapping relevanten Daten in UMLS hinterlegt sind.

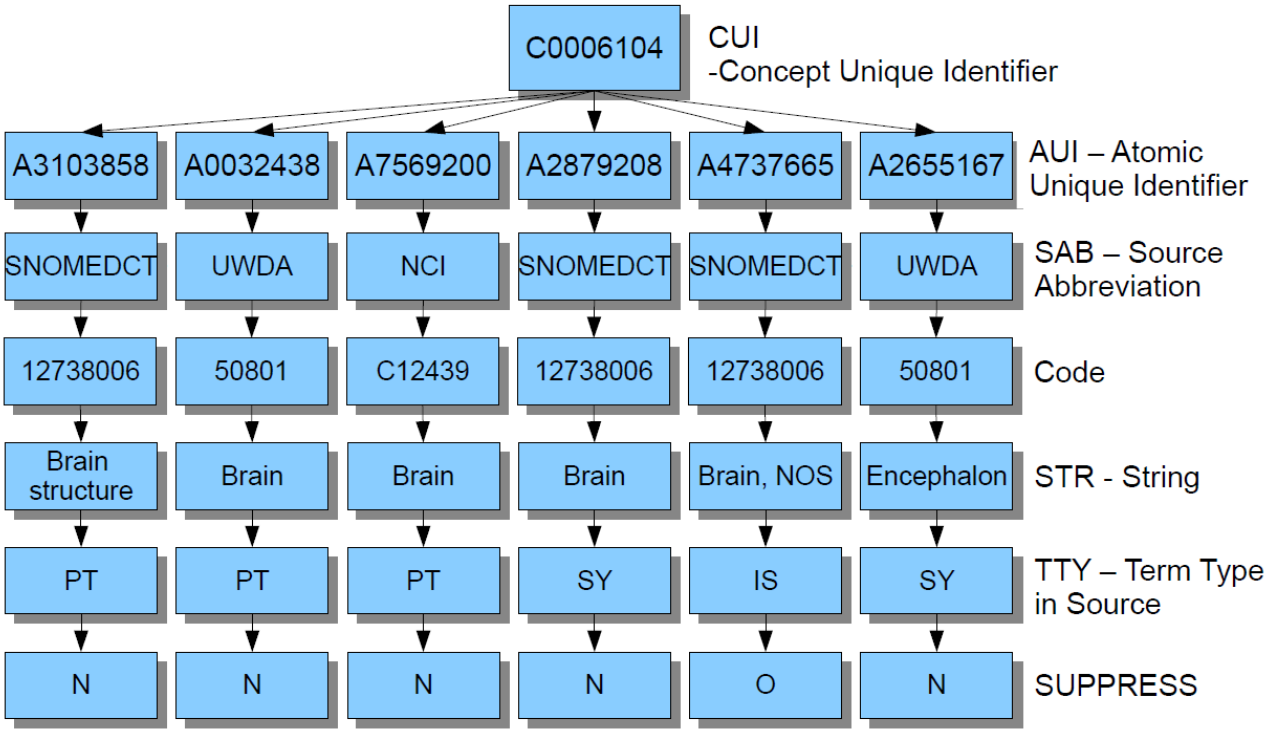

Abbildung 5: Beispielkonzept in UMLS

- *CUI* steht für Concept Unique Identifier, die UMLS-interne ID. Mit dieser kann das Konzept innerhalb der UMLS-Datenquelle eindeutig identifiziert werden.
- *AUI* steht für Atomic Unique Identifier, mit dem die einzelnen Instanzen eines Konzeptes in UMLS eindeutig identifiziert werden können.
- Hinter *SAB* verbirgt sich die Source Abbreviation, die Abkürzung der Quelle aus der die jeweilige Instanz des Konzeptes stammt.
- *Code* ist die interne ID der Konzepte in ihren Quellen.
- *STR* (String) in Verbindung mit *TTY* (Term Type in Source) gibt an ob es sich bei

dem Term um den Namen (PT, Prefered Term) oder ein Synonym (SY) handelt. Es gibt Sonderfälle, zum Beispiel für obsolete Terme.

- SUPPRESS gibt an ob eine Instanz des Konzeptes obsolet ist oder nicht. O steht dabei für obsolet und N für aktuell (nicht obsolet).

Bei dem Beispiel in Abbildung 4 sieht man einen Eintrag des Konzeptes „Brain“ in UMLS. Man kann erkennen, dass das Konzept in allen 3 ausgewählten Ontologien existiert, auch wenn es unterschiedliche Namen dafür gibt. In FMA und NCIt heißt das Konzept ebenfalls „Brain“, während es in SNOMED CT „Brain Structure“ heißt und „Brain“ als Synonym hat. Der veraltete Term „Brain, NOS“ wurde nicht aus UMLS entfernt, sondern auf obsolet gesetzt. Falls er wieder eingeführt werden sollte muss hier nur eine Änderung des Eintrags vorgenommen werden. Daraus ergibt sich folgende Darstellung der Konzepte in den Quellontologien:

	NCI	**FMA**	**SNOMEDCT**
ID	C12439	50801	12738006
Name	Brain	Brain	Brain structure
Synonym		Encephalon	Brain

Unter dem *Concept Unique Identifier* sind alle Instanzen des Konzeptes in UMLS abgelegt. Daher ist es möglich in Verbindung mit der *Source Abbreviation* und dem *Code* die Mappings zwischen den Ontologien direkt zu extrahieren (*Abbildung 6*).

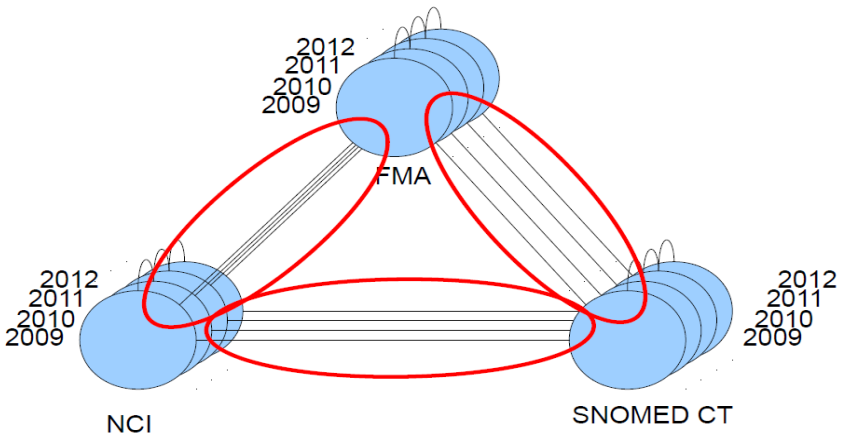

Abbildung 6: Mappings zwischen den Ontologien können direkt extrahiert werden

Die Extraktion der Mappings wurde ebenfalls mit einem Programm in Java realisiert. Der grobe Ablauf ist in *Abbildung 7* dargestellt.

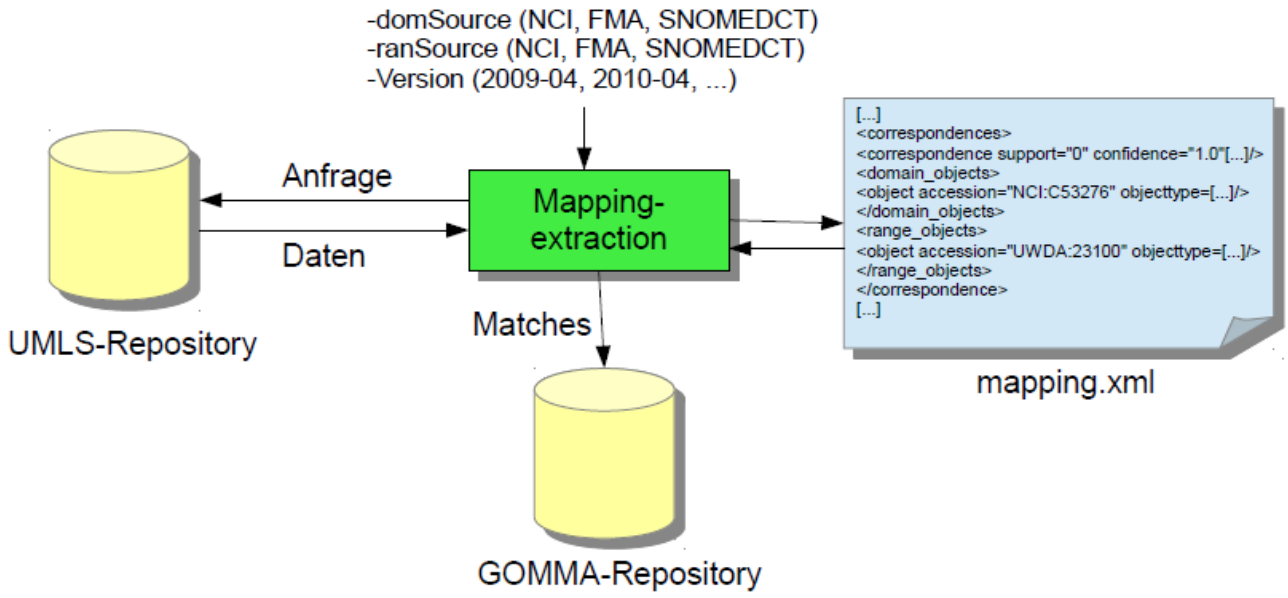

Abbildung 7: Extraktion der Mappings aus UMLS; Import der Mappings in GOMMA

Das Programm *Mappingextraction* bekommt als Eingabewerte die *domSource* (domain Source, die Ontologie, die auf eine andere gemappt werden soll), die *ranSource* (range Source, die Ontologie, auf die gemappt werden soll) und die *Version* der Ontologien. Mit diesen Daten wird über einen jdbc-Treiber mit prepared Statements auf das UMLS-Repository zugegriffen. Mit der ersten Anfrage werden alle CUI's mit dem dazugehörigen Code der ranSource abgerufen und ein einer Containerklasse hinterlegt, die in einer Liste gespeichert wird. Mit diesem Schritt erhält man alle Konzepte der ranSource, die in UMLS gespeichert sind und deren ID in der Quellontologie. Danach wird über die einzelnen Elemente der Liste iteriert und mit jedem *CUI* geprüft ob in dem Konzept in UMLS auch eine Instanz von Konzepten der ranSource existiert. Ist das der Fall, wird der Code für das Konzept der ranSource abgerufen und auf den Code der domSource gemappt. Die Mappinginformationen werden dann in einer XML-Datei gespeichert. Nachdem die Überprüfung für alle Elemente abgeschlossen ist, werden alle Daten aus der Datei *mapping.xml* über die bereitgestellte Schnittstelle in das GOMMA-Repository übertragen. Dieser Vorgang wird für alle 4 Versionen durchgeführt.

Mit den in diesem Kapitel aufgeführten Arbeitsschritten wurde ein Benchmark erstellt, der die 3 großen Ontologien der Lebenswissenschaften (Foundational Model of Anatomy, National Cancer Institute Thesaurus und SNOMED-Clinical Terms) in 4 verschiedenen Versionen (2009-2012) sowie die Mappings zwischen zwischen den Ontologien, ebenfalls in 4 Versionen (2009-2012). Die dafür genutzten und hier aufgeführten Werkzeuge zur Erstellung des Benchmarks können problemlos angepasst werden um weitere Ontologien, Mappings und Versionen zu extrahieren. Dafür müssen nur die Eingabeparameter bei Aufruf der Klassen *DataImport* und *Mappingextraction* entsprechend angepasst werden. Dieser Benchmark ermöglicht durch die verschiedenen Mappingversionen eine Evaluierung von Adaptierungsstrategien. Ein solcher Benchmark existierte bisher nicht. Allerdings kann man bei einem auf diese Weise erstellten Benchmark nicht von einem Goldstandard für die Referenzmappings ausgehen, weil die einzelnen Versionen von UMLS zum Teil durch Expertenwissen optimiert werden. Dadurch können in einer neuen Mappingversion Korrespondenzen existieren, die der Adaptierungsalgorithmus nicht finden kann, da keine Änderung der betroffenen Konzepte stattgefunden hat. Deshalb entsprechen die Referenzmappings dieses Benchmarks eher einem „Silberstandard“.

Der in dieser Arbeit erstellte Benchmark kam bereits für einzelne Publikationen erfolgreich zum Einsatz und wurde auch in Kapitel 5 zur Evaluierung der Mappingstrategien genutzt. In [26] wurde anhand der im Rahmen dieser Arbeit extrahierten Ontologien und Mappings Ontology Change Operations (OCO) definiert und welche Adaptierungsstrategien die entsprechenden OCO's nach sich ziehen. In dem Fall wurden die Daten als Datenquelle für die Untersuchungen genutzt. Die Publikation [20] verwendet den in dieser Arbeit erstellten Benchmark bereits erfolgreich zur Evaluierung der Ergebnisse, der darin vorgestellten Adaptierungsstrategien.

4. Mappingstrategien

Im folgenden Kapitel werden die für diese Arbeit verwendeten Werkzeuge GOMMA und COnto-Diff vorgestellt, sowie eine grobe Übersicht über deren Arbeitsweise gegeben. Es werden bestehende Strategien zur Mappingadaption vorgestellt. Die bestehende Split-Strategie wird angepasst und eine neue Strategie für den Umgang mit Attributänderungen wird eingeführt.

4.1. GOMMA

GOMMA (Generic Ontology Matching and Mapping Management)[27] bietet eine umfassende Infrastruktur zur Verwaltung und Analyse von Ontologien und Mappings sowie deren Evolution in den Lebenswissenschaften. GOMMA benutzt ein generisches Repository, um Ontologieversionen und verschiedene Mappings einheitlich und effizient zu verwalten. Darüber hinaus werden verschiedene Funktionalitäten zum Matching von Ontologien und zur Bestimmung von Änderungen zwischen verschiedenen Versionen angeboten. *Abbildung 8* bietet einen Überblick über die verschiedenen Komponenten von GOMMA. Das System besteht aus 3 verschiedenen Ebenen:

- einem Repository
- funktionellen Komponenten und
- Applikationen.

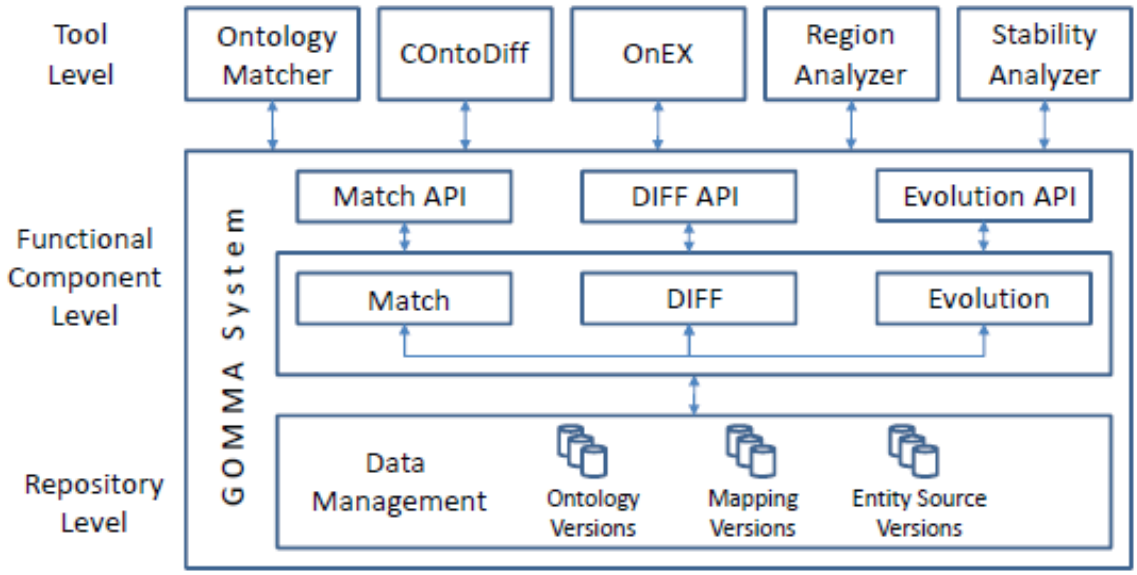

Abbildung 8: Überblick über die komponentenbasierte Infrastruktur von GOMMA (entnommen aus [27])

Das Repository ermöglicht die einheitliche und effiziente Verwaltung verschiedener Versionen von Ontologien, Mappings und Instanzquellen. Für den Import von Ontologien und Mappings in GOMMA werden verschiedene Funktionalitäten bereitgestellt. So werden beispielsweise typische Ontologieformate wie OBO und OWL (Web Ontology Language) [28] unterstützt. Mappings können aus Dateien im CSV-, XML- oder RDF-Format importiert

oder in diese Formate exportiert werden. Dabei werden Elemente einer Datenquelle (Konzepte einer Ontologie, deren Attribute und Beziehungen zueinander) einmalig in GOMMA abgespeichert, unabhängig davon, in wie vielen Versionen die Elemente auftreten. Dadurch wird verhindert, dass unveränderte Elemente redundant für jede Version gespeichert werden. Um dennoch eine vollständige Verfügbarkeit zu garantieren kommt ein Versionierungsmodell zum Einsatz. Jedes Element hat eine bestimmte Lebensdauer, die durch ein festes Start- (t_{start}) und Enddatum (t_{end}) festgelegt ist. Da jeder Ontologieversion ein festes Versionsdatum t zugeordnet ist, kann GOMMA anhand dieser Informationen feststellen ob ein Element in einer bestimmten Version vorkommt oder nicht. $(t_{start} \leq t \leq t_{end})$ Diese Funktionalität wurde im Rahmen dieser Arbeit genutzt um die Ontologien in verschiedenen Versionen einheitlich und effizient zu speichern und zu verwalten.

Die funktionelle Komponente setzt sich aus den drei Einzelkomponenten *Match*, *Diff* und *Evolution* zusammen. Die funktionellen Komponenten greifen über eine zentrale Schnittstelle auf die im GOMMA-Repository gespeicherten Ontologie- und Mappingversionen zu. Die *Match*-Komponente dient der automatischen Bestimmung von Ontologiemappings. Dafür wird die semantische Ähnlichkeit zwischen zwei Ontologiekonzepten bestimmt und wenn ein festgelegter Ähnlichkeitswert erreicht wird, dann werden beide Konzepte aufeinander gemappt. Mit der *Diff*-Komponente können Diff-Evolutionsmappings zwischen verschiedenen Versionen von Ontologien erstellt werden. Dabei kann zwischen einfachen und komplexen Änderungsoperationen unterschieden werden. (siehe Kapitel 4.2 Conto-Diff) Die mit *Match* und *Diff* berechneten Ontologie- und Evolutionsmappings können direkt im GOMMA-Repository gespeichert werden. Mit der Komponente *Evolution* kann eine Evolutionsanalyse von Ontologiequellen durchgeführt werden. Dafür wird die Änderungen in der Ontologie statistisch ausgewertet.

Auf der oberen Ebene befinden sich verschiedene Applikationen, die über eine komponentenspezifische Schnittstelle (API) auf die drei funktionellen Komponenten *Match*, *Diff* und *Evolution* zugreifen. Für diese Arbeit wurde die Applikation COntoDiff verwendet. Diese wird auf den folgenden Seiten näher betrachtet.

COnto-Diff

COnto-Diff[22] ist ein wesentlicher Bestandteil von der *Diff*-Komponente in GOMMA. Es ermöglicht das Erstellen eines semantisch ausdrucksstarken Diff-Evolutionsmappings zwischen verschiedenen Ontologieversionen. Der COnto-Diff-Algorithmus (Abbildung 9) erstellt als ersten Schritt ein Ontologiemapping $OM_{O_{old},O_{new}}$ zwischen der alten O_{old} und der neuen Ontologieversion O_{new}. Zusätzlich zu den beiden Ontologieversionen können für diesen Schritt Hintergrundinformationen, zum Beispiel lexikographisches Wissen, hinzugezogen werden. Für das Mapping kann beispielsweise die Komponente *Match* von GOMMA genutzt werden. Auf Basis des so erstellten Mappings und der beiden Ontologieversionen werden sogenannte Basisänderungsoperationen bestimmt. Diese beziehen sich immer auf ein einzelnes Konzept, eine Beziehung oder ein Attribut und decken dabei einfache Veränderungen ab. Zu diesen Basisänderungen gehören das Hinzufügen (*add*), das Löschen (*del*) und das Ändern bzw. Abbilden (*map*). Die bestimmten Basisänderungsoperationen bilden zusammen ein Diff-Evolutionsmapping $diff_{basic}(O_{old}, O_{new})$.

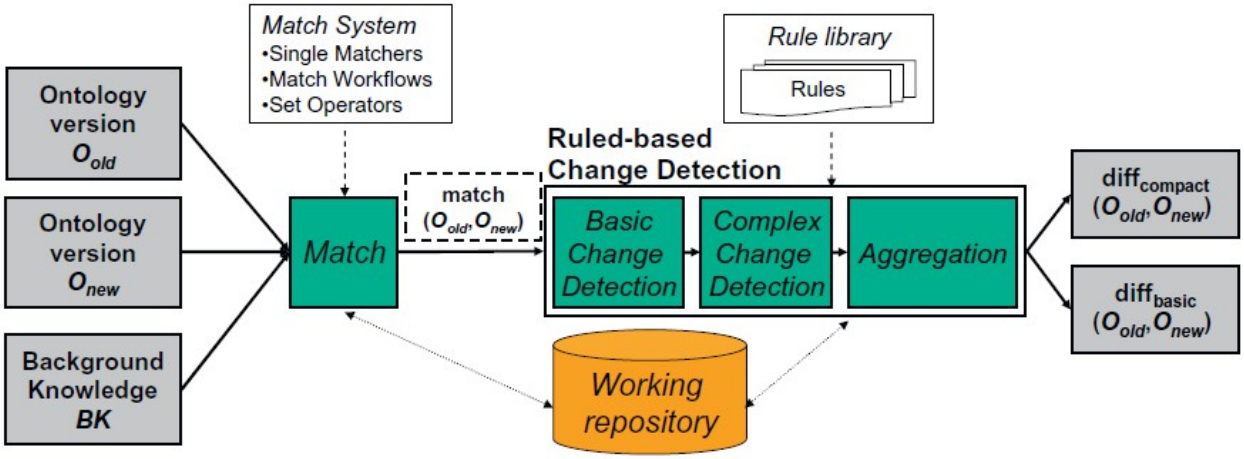

Abbildung 9: COnto-Diff-Algorithmus (entnommen aus [22])

Dieses einfache Diff-Evolutionsmapping wird im folgenden Schritt durch komplexe Änderungsoperationen erweitert, um es semantisch anzureichern. Beispiele für komplexe Änderungsoperationen sind das Zusammenfassen mehrerer Konzepte zu einem Konzept (*merge)* oder die Aufspaltung einzelner Konzepte in eine Menge von neuen Konzepten (*split*). Diese komplexen Änderungsoperationen setzen sich aus wiederholter Anwendung der Basisänderungsoperationen zusammen und sind in bestimmten Regeln (Rule library)

beschrieben. *Abbildung 10* zeigt eine solche Regel für die komplexe Änderungsoperation *merge*. Das Ergebnis ist dieses Schrittes ist ein komplexes Diff-Evolutionsmapping $diff_{compact}(O_{old}, O_{new})$.

$$
\begin{aligned}
& a, b \in O_{old} \wedge c \in O_{new} \wedge mapC(a,c) \wedge mapC(b,c) \wedge a \neq b \\
& \wedge\ \nexists d(d \in O_{new} \wedge mapC(a,d) \wedge c \neq d) \\
& \wedge\ \nexists e(e \in O_{new} \wedge mapC(b,e) \wedge c \neq e) \\
& \rightarrow \mathbf{create}[merge(\{a\},c), merge(\{b\},c)], \\
& \quad \mathbf{eliminate}[mapC(a,c), mapC(b,c)]
\end{aligned}
$$

Abbildung 10: merge in Conto-Diff als Kombination von Basisänderungsoperationen (aus [22])

4.2. Bestehende Strategien zur Mappingadaptierung

Die in diesem Abschnitt behandelten Mappingadaptierungsstrategien wurden in der Publikation [20] vorgestellt. Diese Arbeit baut auf den diff-basierten Adaptierungsstrategien auf, die in dieser Publikation vorgestellt werden. Die Grundidee hinter der diff-basierten Herangehensweise ist es, so viele Teile des alten Mappings wie möglich zu verwenden und nur die Korrespondenzen neu zu bestimmen, deren Konzepte von der Ontologieevolution betroffen sind. Dafür werden über COnto-Diff alle Änderungen bestimmt und die betroffenen Korrespondenzen gesammelt. Die Korrespondenzen, auf deren Konzepte die Ontologieevolution keinen Einfluss hatte werden für das Evolutionsmapping übernommen. Abhängig von den ermittelten Veränderungen werden sogenannte Changehandler angewandt um das Mapping zu adaptieren. In diesen Changehandlern sind verschiedene Strategien hinterlegt, die bestimmen wie das Mapping bei den entsprechenden Änderungen zu adaptieren ist. Dieser Ansatz zeichnet sich durch eine hohe Modularität aus. Die Mappingadaptierung kann dadurch nur anhand von bestimmten Änderungen durchgeführt werden, indem nur ausgewählte Changehandler aufgerufen werden und eine schnelle Anpassung der bestehenden sowie das Hinzufügen neuer Strategien innerhalb der Changehandler ist jederzeit möglich. *Abbildung 11* zeigt die in [20] vorgestellten Changehandler.

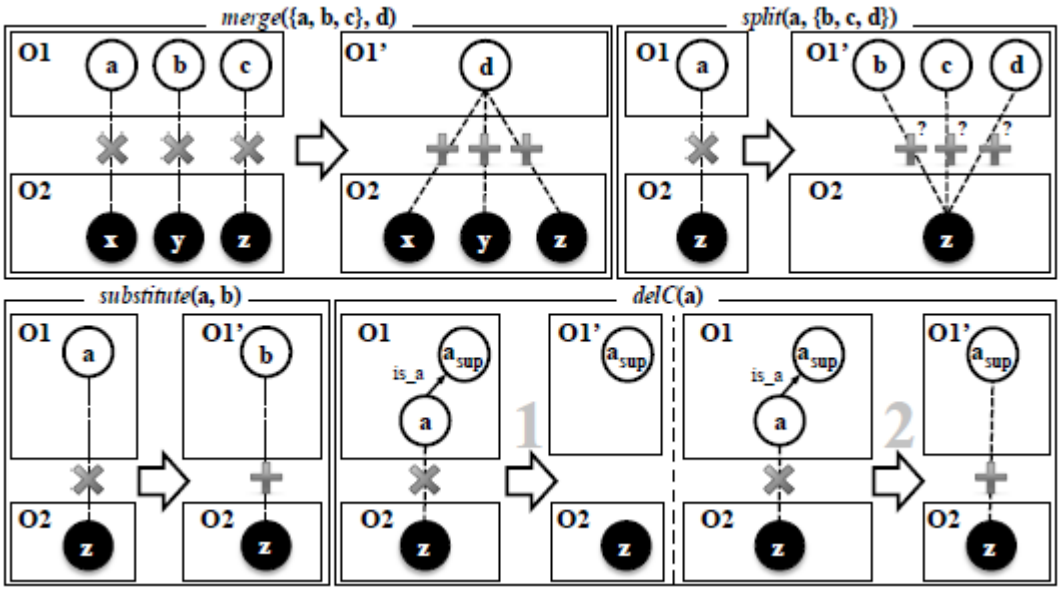

Abbildung 11: Changehandler (übernommen aus [20])

Im folgenden werden einzelne Changehandler sowie deren Strategie vorgestellt. Diese sind CH_{Merge}, $CH_{Substitute}$, CH_{delC}. Auf die Changehandler CH_{Split} und $CH_{attrChg}$ wird in den folgenden Abschnitten des Kapitels eingegangen, weil deren Strategien im Rahmen dieser Arbeit erweitert bzw. neue Strategien hinzugefügt wurden.

Bei einem *Merge* werden zwei oder mehr Konzepte der Ontologie *O1* zu einem Konzept in *O1'* zusammengefügt. Der Changehandler überträgt alle Korrespondenzen, die mit den betroffenen Konzepten in *O1* zusammen hängen und matcht diese mit den Zielkonzepten der jeweiligen Mergeoperationen. Danach werden die alten Korrespondenzen aus dem Mapping $M_{O1,O2}$ entfernt und die neuen hinzugefügt.

Bei *Substitute(a,b)* wird eine ähnliche Strategie wie bei *Merge* angewendet. In der *Abbildung 11* wird das Konzept a ∈ O1 durch das Konzept b ∈ O1' ersetzt. Da a mit dem Konzept z ∈ O2 über eine Korrespondenz verbunden war wird die Korrespondenz zwischen a und z entfernt und eine neue Korrespondenz zwischen b und z hinzugefügt.

Bei dem Löschen von Konzepten *delC* werden 2 Strategien verfolgt. Die erste sieht vor, dass alle betroffenen Korrespondenzen gelöscht werden (*1* bei *delC(a)* in *Abbildung 11*). Bei der zweiten Strategie werden die Korrespondenzen auf das Elternkonzept a_{sup} übertragen. Die Korrespondenzen mit *a* werden aus dem Mapping $M_{O1,O2}$ entfernt und die neuen Korrespondenzen hinzugefügt.

4.3. Anpassung der Split-Strategie

Die zwei bereits implementierten Split-Strategien (*TAKE BEST* und *TAKE ALL*) haben beide eine ähnliche Herangehensweise. Bei einem Split eines Konzepts a der Ontologie O1 $split(a,(b,c,d))|a \in O1; b,c,d \in O1'$ mit der dazugehörigen, vereinfachten Korrespondenz $corr(a,z)|a \in O1; z \in O2$ wird für die durch den Split erzeugten Konzepte b,c und d nur z als mögliches Ziel einer neuen Korrespondenz in Betracht gezogen. (*Abbildung 11*) Je nach verwendeter Strategie entstehen so unterschiedliche Korrespondenzen.

- TAKE BEST: nur das durch den Split erzeugte Konzept mit der größten Übereinstimmung mit z wird für eine Korrespondenz genutzt
- TAKE ALL: es werden für alle durch den Split erzeugten Konzepte werden Korrespondenzen mit z erstellt

Für das folgende Beispiel (*Abbildung 12*) zeigt die Evolution der Ontologie NCIt und wie das Mapping zwischen NCIt und FMA anhand der alten Split-Strategien adaptiert wird.

Die bestehenden Mappingstrategien erzeugten folgende vereinfachte Korrespondenzen aufgrund von

$$split(\text{Ear, (Ear,Left Ear,Right Ear)})|\text{Ear} \in NCI\ 2009; \text{Ear, Left Ear, Right Ear} \in NCI\ 2012 \quad :$$

TAKE LOCAL BEST:

- $corr(\text{Ear,Ear})|\text{Ear} \in NCI2012; \text{Ear} \in FMA\ 2012$

TAKE ALL:

- $corr(\text{Ear,Ear})|\text{Ear} \in NCI2012; \text{Ear} \in FMA\ 2012$
- $corr(\text{Left Ear, Ear})|\text{Left Ear} \in NCI2012; \text{Ear} \in FMA\ 2012$
- $corr(\text{Right Ear,Ear})|\text{Right Ear} \in NCI2012; \text{Ear} \in FMA\ 2012$

Beide Ergebnisse sind nicht optimal, da für *Right Ear* und *Left Ear* entsprechende Konzepte in FMA existieren, diese aber nicht vom Matchalgorithmus berücksichtigt werden. Selbst werden sie auch nicht neu behandelt, weil sie bereits in der Version von 2009 vorkamen und damit an dieser Stelle keine Änderung stattgefunden hat.

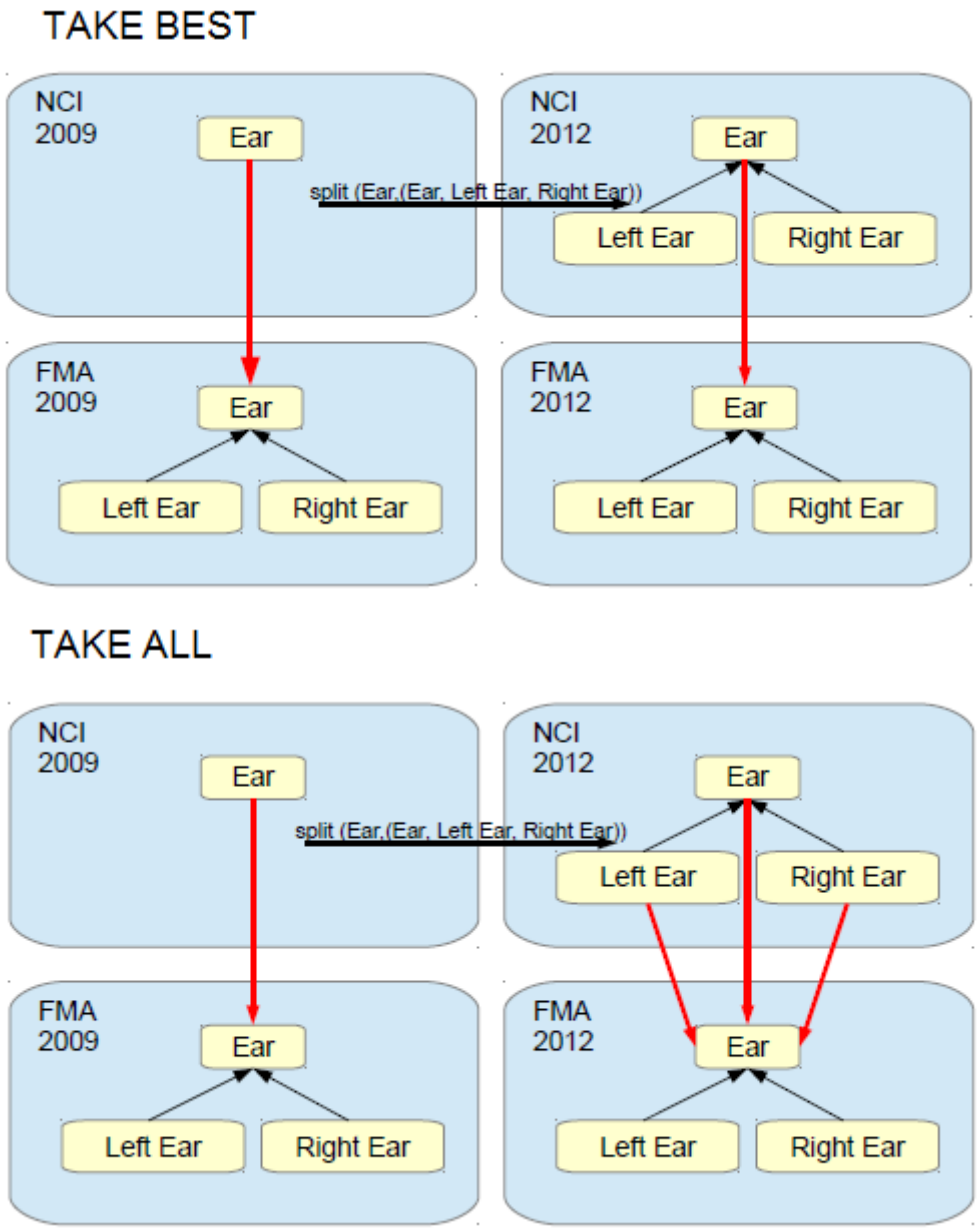

Abbildung 12: Beispiel für die Anwendung der bestehenden Splitstrategien

Die Strategie wurde deshalb dahingehend angepasst, dass bei einem *Split* $split(a,(b,c,d))|a \in O1; b,c,d \in O1'$ und einer bestehenden Korrespondenz $corr(a,z)|a \in O1; z \in O2$ nicht nur das Konzept ***z*** für potentielle neue Korrespondenzen berücksichtigt wird, sondern die Hierarchie der Ontologie genutzt wird um auch die Kinderkonzepte von ***z*** $z_{c1}, z_{c2}, \ldots, z_{cn}|(z_{c1}, \ldots, z_{cn}) \in O2'$ mit einzubeziehen.

```
foreach corr ∈ M do
    foreach split ∈ Split do
        s ← split.getSourceID();
        if s = corr.getDomainID() then
            T← split.getTargetIDs();
            newType ← getNewType(corr.getType());
            newStatus ← getNewStatus(corr.getStatus());
            foreach t∈ T do
                newCorr ← createCorr(tAcc, corr.getRangeID(),
                corr.getSim(),newType,newStatus);
                M.remove(corr).add(newCorr);
```

Abbildung 13: Algorithmus für alte Split-Strategie (TAKE ALL)

Abbildung 13 zeigt den Algorithmus für den $splitHandler(M, Split, O1, O1', O2)$ mit der Strategie TAKE ALL. *M* ist die Menge aller Korrespondenzen des existierenden Mappings zwischen *O1* und *O2*. *Split* beinhaltet alle Splitoperationen des Evolutionsmappings zwischen *O1* und *O1'*. Der Algorithmus prüft für alle Korrespondenzen *corr* und alle Elemente von *M*, ob das Domain-Konzept einer Korrespondenz mit einem Source-Konzept in *Split* übereinstimmt. Wenn dies zutrifft, dann werden für alle Target-Konzepte von Split und das Range-Konzept der Korrespondenz neue Korrespondenzen hinzugefügt und die alte Korrespondenz gelöscht.

Die angepasste Strategie (*Abbildung 14*) verläuft analog zur Strategie *TAKE ALL*, nur werden hier nicht alle Target-Konzepte von *Split* automatisch auf das Range-Konzept der Korrespondenz gemappt. In einem Zwischenschritt werden auch dessen Kinder *C* als potentielle Kandidaten für das Range-Konzept behandelt. Durch einen Matchingalgorithmus wird der Kandidat bestimmt, der die meiste Ähnlichkeit mit dem jeweiligen Target-Konzept hat und die neue Korrespondenz erstellt. Für die Ähnlichkeit kann ein Mindestwert *minSim* festgelegt werden, so dass mitunter keine neue Korrespondenz erstellt wird.

Die Abfolge des Algorithmus ist wie folgt:

- Prüfe ob Domain-Konzept der Korrespondenz mit Source-Konzept in *Split* übereinstimmt!
- Prüfe ob Target-Konzept der Korrespondenz Kinder hat und rufe diese ggf. ab!
- Prüfe welcher Kandidat die höchste Ähnlichkeit mit dem Source-Konzept hat und küre ihn zum „Sieger“!
- Erstelle neue Korrespondenz, setze dabei Source-Konzept = Domain-Konzept und Target-Konzept = „Sieger“!
- Füge neue Korrespondenz hinzu und lösche die alte!

Diese Strategie ermöglicht eine einfache Erweiterung der Kandidatenmenge um eventuell geeignetere Paare für das Erstellen einer Korrespondenz zu finden, ohne dass dabei die komplette Ontologie betrachtet werden muss.

```
foreach corr ∈ M do
    foreach split ∈ Split do
        s ← split.getSourceID();
        if s = corr.getDomainID() then
            T ← split.getTargetIDs();
            C ← corr.getRange().getChildren();
            newType ← getNewType(corr.getType());
            newStatus ← getNewStatus(corr.getStatus());
            foreach t∈ T do
                w ← getWinnerWithHighestSim(t, (C,corr.getRange()));
                sim ← getSim(t,w);
                if sim ≥ minSim then
                    newCorr ← createCorr(tAcc, wAcc,
                    corr.getSim(),newType,newStatus);
                    M.remove(corr).add(newCorr);
                else
                    M.remove(corr);
```

Abbildung 14: Algorithmus für angepasste Splitstrategie (HIERARCHIE)

4.4. Strategie bei Attributänderung

Die existierenden Strategien hatten eine recht einfache Herangehensweise. Entweder wurden alle Korrespondenzen, mit betroffenen Konzepten neu bestimmt (*MATCH ALL*) oder sie werden abhängig davon, ob die betroffene Korrespondenz noch von einer anderen Änderungsoperation betroffen war und ein Rematch schon stattgefunden hatte, ignoriert (*MATCH UNMATCHED*).

Im Rahmen dieser Arbeit wurde die Mappingadaptierungsstrategie *TRIGRAM* erstellt. Die Motivation dahinter ist die Vermutung, dass Attributänderungen in vielen Fällen keinen großen Einfluss auf das bestehende Mapping haben. In den meisten Fällen handelt es sich bei den Attributänderungen um das Hinzufügen, Löschen oder Änderungen in der Schreibweise von Namen und Synonymen. Von daher können eventuell bessere Werte im Benchmark erzielt werden, wenn man nicht alle von Attributänderungen betroffenen Korrespondenzen neu bestimmt, sondern nur diejenigen, die eine Veränderung über einem bestimmten Schwellwert (*treshhold*) erfahren haben.

Um dies umzusetzen wurden die Attributänderungen $chgAttValue(c, att, value_{old}, value_{new})$ näher untersucht. Bei jeder Änderung wurden $value_{old}$ und $value_{new}$ miteinander verglichen. Dafür werden sie in Trigramme zerlegt und der Dice-Koeffizient ermittelt. Nur wenn dieser unter einem festgelegten Schwellwert liegt wird die betroffene Korrespondenz neu bestimmt.

Beispiel: Der Name des Konzeptes *c* wird von *Hirn* auf *Gehirn* geändert. Dies wird von Conto-Diff erkannt und in das Evolutionsmapping aufgenommen. Dem Changehandler für Attributänderungen wird die folgende Information übergeben:

$$chAttValue(c, Name, Hirn, Gehirn)$$

Daraufhin wird die Ähnlichkeit zwischen den beiden Termen untersucht, indem diese erst in Trigramme zerlegt werden und dann der Dice-Koeffizient ermittelt wird. Damit Änderungen an den Rändern des Terms dabei nicht vernachlässigt werden, werden beim preprocessing an Anfang und Ende 2 Füllzeichen angehängt. Dies ist nötig, da die beiden äußeren Buchstaben sonst nur in einem der Trigramme vorkommen würden und nicht in drei Trigrammen, wie die Buchstaben in der Mitte des Terms. Dann wird bestimmt wie viele der Trigramme übereinstimmen. Der Dice-Koeffizient wird nach der Formel

$K_{Dice} = \frac{2 \cdot |Übereinstimmungen|}{|Trigramme\ Term\ 1| + |Trigramme\ Term\ 2|}$ bestimmt. Das Ergebnis sieht wie folgt aus:

```
Hirn zerlegt in {n$$=>1, rn$=>1, $$H=>1, $Hi=>1, irn=>1, Hir=>1}
Gehirn zerlegt in {n$$=>1, rn$=>1, $Ge=>1, $$G=>1, Geh=>1, ehi=>1,
irn=>1, hir=>1}
Übereinstimmungen: 3
Anzahl Trigramme: 14
sim(Hirn, Gehirn) = 0.42857143
```

Je nach festgelegtem Maximalwert für die Ähnlichkeit (*maxSim*) wird die betroffene Korrespondenz neu bestimmt oder die bestehende Korresponenz beibehalten.

$$minSim = 0{,}5 \Rightarrow bestimme\ Korrespondenz\ neu$$
$$minSim = 0{,}3 \Rightarrow behalte\ Korrespondenz\ bei$$

Bei mehreren Änderungen eines Attributes zum Beispiel dem Hinzufügen mehrerer Synonyme $chAttValue(c, Synonym, Ader, (Adern, Blutgefäß, Blutgefäße))$, wird der Dice-Koeffizient für alle Änderungen einzeln bestimmt und der höchste Wert wird für den Vergleich mit *maxSim* genutzt.

```
foreach corr ∈ M do
    foreach chg∈ AttrChg do
        c ← chg.getConceptID();
        if c = corr.getDomainID() then
            v1 ← chg.getValueOld();
            v2 ← chg.getValueNew();
            sim ← trigram(v1,v2);
            if sim ≤ maxSim then
                w ← getWinnerWithHighestSim(s, O2);
                newCorr ← createCorr(cAcc, wAcc,
                getSim(c,w),getNewType(c,w), getNewStatus(c,w));
                M.remove(corr).add(newCorr);
```

Abbildung 15: Algorithmus für Strategie bei Attributänderungen (Trigramm)

Der Ablauf des Algorithmus (*Abbildung 15*) ist wie folgt:

- Prüfe ob Domain-Konzept der Korrespondenz mit dem Konzept übereinstimmt dessen Attribut geändert wurde!
- Bestimme den Dice-Koeffizienten zwischen altem und neuem Attributwert!
- Liegt der Dice-Koeffizient unter einem festgelegten maximalen Ähnlichkeitsmaß (*maxSim*), dann bestimme die Korrespondenz neu! Füge die neue Korrespondenz hinzu und lösche die alte!
- Liegt der Dice-Koeffizient über einem festgelegten Wert, dann behalte die bestehende Korrespondenz.

Eine Variante dieser Adaptierungsstrategie ist *Trigramm Ignore Syn* (*Abbildung 16*). Bei dieser Strategie werden Änderungen, die nur Synonyme betreffen ignoriert und die Korrespondenzen werden nicht neu ermittelt. Der Ablauf ist analog zu Trigramm, nur dass noch eine Überprüfung stattfindet ob es sich bei dem geänderten Attribut um ein Synonym handelt. Ist dies der Fall, dann wird die bestehende Korrespondenz behalten.

```
foreach corr ∈ M do
    foreach chg∈ AttrChg do
        if chg.getAtt ≠ „synonym“ then
            c ← chg.getConceptID();
            if c = corr.getDomainID() then
                v1 ← chg.getValueOld();
                v2 ← chg.getValueNew();
                sim ← trigram(v1,v2);
                if sim ≤ maxSim then
                    w ← getWinnerWithHighestSim(s, O2);
                    newCorr ← createCorr(cAcc, wAcc,
                    getSim(c,w),getNewType(c,w), getNewStatus(c,w));
                    M.remove(corr).add(newCorr);
```

Abbildung 16: Algorithmus für Strategie bei Attributänderungen (Trigramm Ignore Syn)

5. Evaluierung

Im folgenden Kapitel werden die werden die Änderungen an den Strategien zur diff-basierten Mappingadaptierung anhand des erstellten Benchmarks evaluiert. Als erster Schritt erfolgt eine quantitative Untersuchung der drei extrahierten großen Ontologien der Lebenswissenschaften FMA, NCIt und SNOMED CT um die Aussagekraft der Ergebnisse des Benchmarks besser einschätzen zu können. Auf Basis dieser Ergebnisse werden die Ontologien und Versionen gewählt, mit denen der Benchmark durchgeführt wird, um die Auswirkungen und die Effektivität der vorgestellten Mappingstrategien überprüft werden.

5.1. Evolution der Ontologien im betrachteten Zeitraum

Ein Benchmark ist dann besonders aussagekräftig, wenn sich die verwendeten Ontologien entsprechend stark verändern, so dass das anhand der Ontologieevolution nötige Mappingadaptierungsverfahren in einem gesunden Maße gefordert wird. Dabei sind zu viele gleichzeitige Änderungen genauso kontraproduktiv wie sehr stabile Ontologien. Deshalb galt die erste Untersuchung der Frage wie stark sich die einzelnen Ontologieversionen im betrachteten Zeitraum von 2009 bis 2012 entwickelt haben. Dafür wurde mit GOMMA ermittelt wie viele Konzepte und Beziehungen die einzelnen Ontologieversionen besitzen.

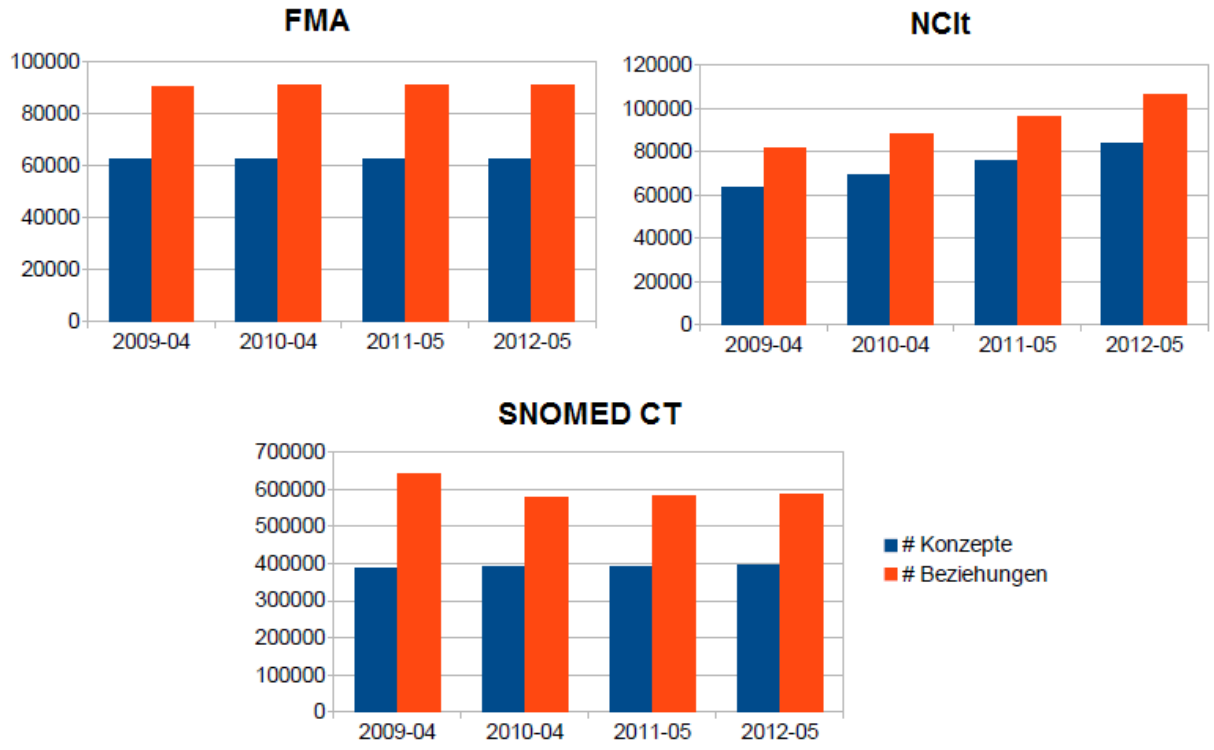

Abbildung 17: Entwicklung von FMA, NCIt und SNOMEDCT 2009-2012

Abbildung 17 zeigt die quantitative Entwicklung der Ontologien FMA, NCIt und SNOMED CT. Es ist erkennbar, dass FMA im betrachteten Zeitraum von 2009 – 2012 sehr stabil bleibt, was die Anzahl der Konzepte und Beziehungen betrifft. NCIt hingegen wächst von Version zu Version stetig stark an. SNOMED CT mag auf den ersten Blick recht stabil erscheinen. Nach einer starken Reduzierung der Beziehungen zwischen den Versionen 2009 und 2010 scheint sich die Ontologie nicht mehr stark zu verändern. Allerdings trügt

der erste Anschein, weil SNOMED CT sehr viel größer ist und deutlich mehr Konzepte und Beziehungen besitzt als FMA und NCIt (*Abbildung 18*). Daher ist die relative Anzahl der Änderungen bei SNOMED CT zwar niedriger als zum Beispiel bei NCIt, aber die absoluten Zahlen zeigen ebenfalls starke Änderung an SNOMED CT.

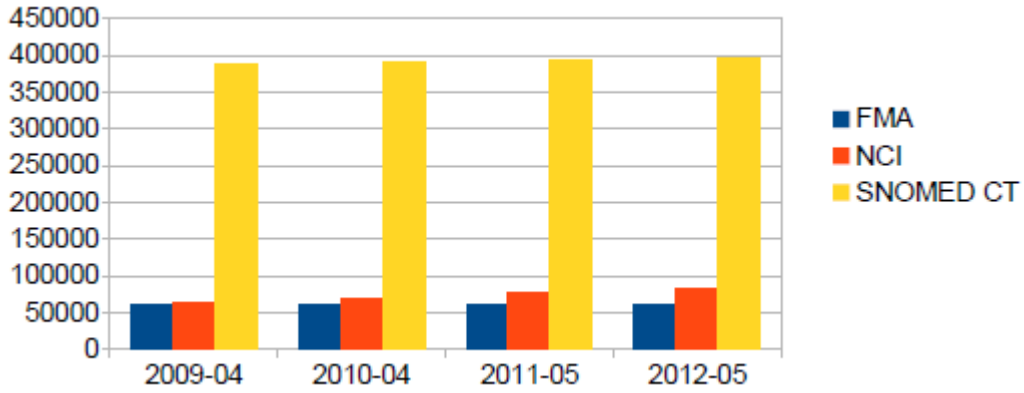

Abbildung 18: Anzahl der Konzepte in den Ontologien

Um dies zu verdeutlichen wurde im nächsten Schritt überprüft, welche Basisänderungen im betrachteten Zeitraum an den Ontologien aufgetreten sind. Dafür wurden die Werte mit COnto-Diff ermittelt, indem ein einfaches Diff-Evolutionsmapping erstellt und analysiert wurde. Die Ergebnisse sind in *Abbildung 19* dargestellt.

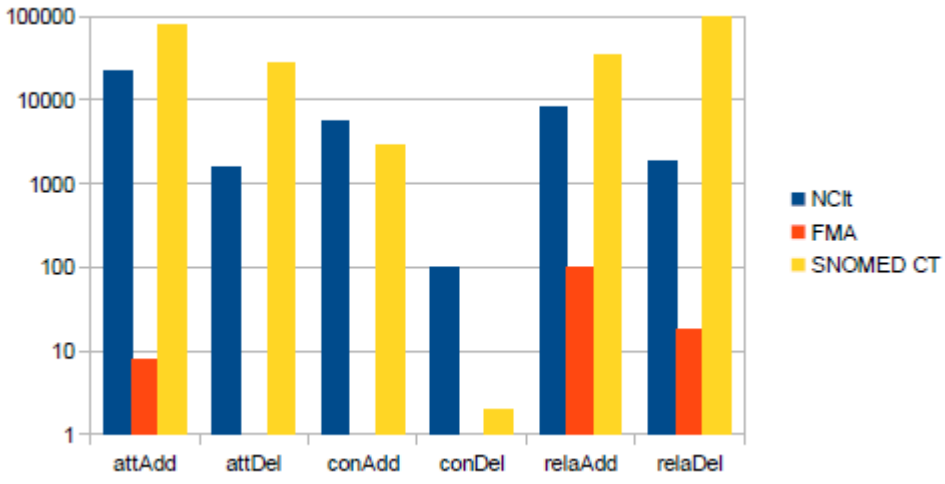

Abbildung 19: Anzahl der Änderungen zwischen den Versionen 2009 und 2010

Hier bestätigt sich die Annahme, dass FMA sehr stabil ist, während sich NCI und SNOMED CT stark verändern. Auffällig ist bei SNOMED CT, dass Konzepte nur sehr selten gelöscht werden (nur 2 Löschungen bei fast 400.000 Konzepten). Diese werden scheinbar innerhalb der Ontologie beibehalten und auf *obsolete* gesetzt. Das hat den Vorteil, dass man sie leicht wieder aktivieren kann, bläht den Datensatz aber sehr stark auf. Von fast 400.000 Konzepten sind nur knapp über 300.000 Konzepte auch aktiv (nicht obsolote). (Stand 2012)

Von diesen Werten her wären für eine Evaluierung ein Benchmark mit den Datensätzen NCIt und SNOMED CT am interessantesten. Aufgrund der extremen Größe und Komplexität der Ontologie SNOMED CT und der daraus schwer im Detail zu untersuchenden Ergebnisse (z.Bsp. Finden von aussagekräftigen Beispielen) wurden die nachfolgenden Evaluierungen der Adaptierungsstrategien vorerst auf den Datensätzen NCIt und FMA in den Versionen von 2009 und 2012 vorgenommen. Eine Evaluierung auf den Datensätzen von FMA und SNOMED CT ergibt keinen Sinn. Durch die geringen Änderungen an FMA und die enorme Größe von SNOMED CT würden die Ergebnisse stark verfälscht. Die Werte für Precision, Recall und F-Measure lagen bei Untersuchungen auf diesen Datensätzen unabhängig von der wirklichen Qualität der angewandten Strategien immer zwischen 98% und 100%, da der unverändert übernommene Teil des Mappings im Vergleich zu den Änderungen zu groß ausfällt. Die Aussagekraft hinter einer Evaluation auf diesen beiden Datensätzen ist daher viel zu gering bzw. nicht gegeben.

5.2. Evaluierung der neuen Mappingstrategien

Im folgenden wird ein kurzer Überblick über die in dieser Arbeit zur Evaluation der angewendeten Mappingstrategien verwendeten Maße und Kriterien gegeben. Bei automatischen Mappingadaptierungsverfahren wird die Qualität der Ergebnisse im Allgemeinen anhand der Maße Genauigkeit (Precision), Abdeckung (Recall) und F-Measure gemessen. Dafür wird ein korrektes Referenzmapping benötigt, der sogenannte Goldstandard, den es im Optimalfall zu erreichen gilt. Durch den direkten Vergleich des Referenzmappings mit dem automatisch erstellten Mapping können richtig positive (*true positive, TP*), falsch positive (*false positive, FP*) und falsch negative (*false negative, FN*) Korrespondenzen bestimmt werden. TP sind Korrespondenzen, die sowohl im Referenzmapping als auch im automatisch erstellten Mapping vorkommen, FP kommen im

Referenzmapping nicht vor und FN sind Korrespondenzen, die im Referenzmapping existieren vom automatischen Adaptierungsverfahren aber nicht identifiziert wurden.

Die *Precision* gibt den Anteil der korrekt identifizierten Korrespondenzen gegenüber allen identifizierten Korrespondenzen an:

$$Precision = \frac{|TP|}{|TP| + |FP|}$$

Der Recall gibt den Anteil der korrekt identifizierten Korrespondenzen gegenüber allen Korrespondenzen des Referenzmappings an:

$$Recall = \frac{|TP|}{|TP| + |FN|}$$

F-Measure ist das harmonische Mittel aus Precision und Recall:

$$F - Measure = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Darüber hinaus können weitere Aspekte wie Speicherplatz und Laufzeit der Adaptierungsverfahren untersucht werden. Gerade bei Verfahren, die einen Nutzer bei der Entscheidungsfindung beim Erstellen manueller Mappings unterstützen sollen und eine Interaktion mit dem Anwender vorsehen, ist die Laufzeit ein entscheidender Faktor. In dieser Arbeit spielt die Laufzeit eine untergeordnete Rolle, weil die durchgeführten Änderung der Mappingstrategien nur einen sehr geringen Einfluss auf die Laufzeit haben.

a) Split-Strategie

Ziel dieser Strategie war es die Qualität des Mappings $M_{O1',O2'}$ zu steigern, indem die hierarchischen Eigenschaften der Ontologie *O2* in das Adaptierungsverfahren einbezogen wurden. Realisiert wurde es dadurch, dass bei einem Split $split(a,(b,c,d))|a \in O1; b,c,d \in O1'$ mit der dazugehörigen, vereinfachten Korrespondenz $corr(a,z)|a \in O1; z \in O2$ auch mögliche Konzepte eine Ebene unter $z \in O2$ im Adaptierungsalgorithmus berücksichtigt wurden. Die Motivation dazu gab das Ergebnis des Splits

$$split(\text{Ear}, (\text{Ear}, \text{Left Ear}, \text{Right Ear}))|\text{Ear} \in NCI\ 2009; \text{Ear}, \text{Left Ear}, \text{Right Ear} \in NCI\ 2012.$$

Das Ergebnis des des angepassten Algorithmus *Hierarchie* ist in *Abbildung 20* zu sehen.

Zum Vergleich der alten Strategien siehe *Abbildung 12*.

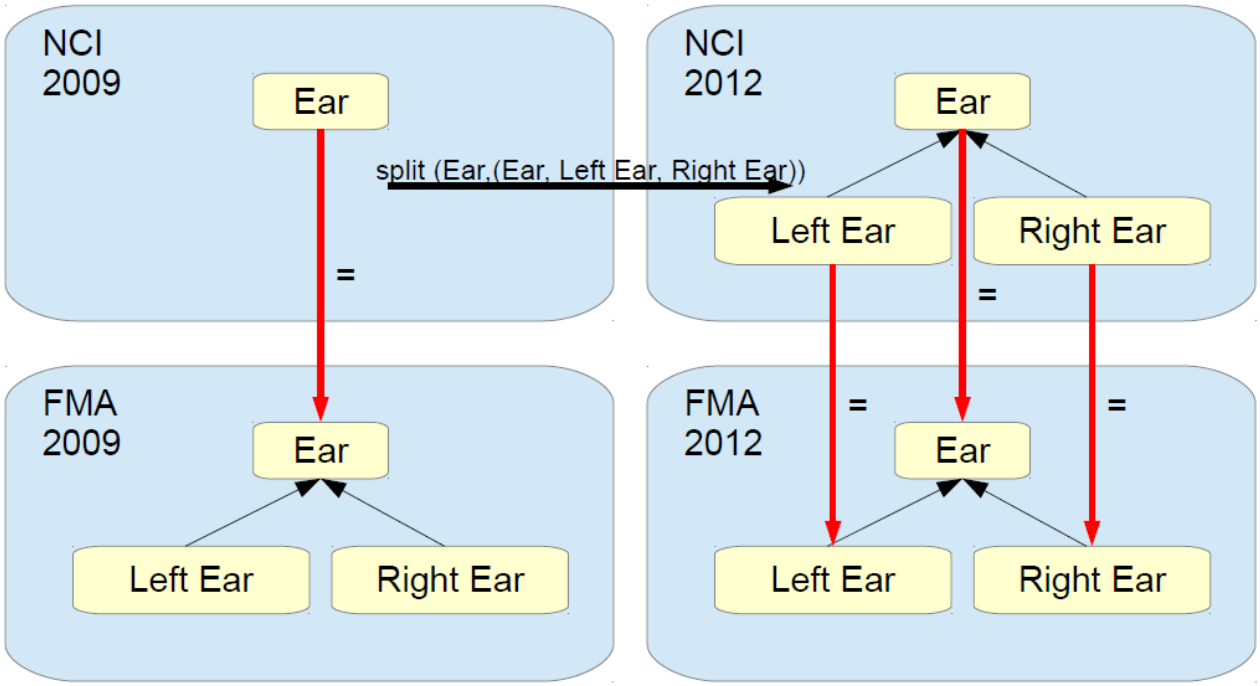

Abbildung 20: Beispiel für das Ergebnis der neuen Strategie Hierarchie

Mit diesem Ergebnis war das Hauptziel erreicht. Es folgt eine Evaluierung der Adaptierungsstrategie anhand des im Rahmen dieser Arbeit erstellten Benchmarks mit den Ontologien FMA und NCI in den Versionen 2009 und 2012. Dabei wurden für das mindeste Ähnlichkeitsmaß (*minSim*) verschiedene Werte (0.0, 0.5, 0.8 und 1.0) festgelegt. Die Ergebnisse sind in *Abbildung 21* dargestellt.

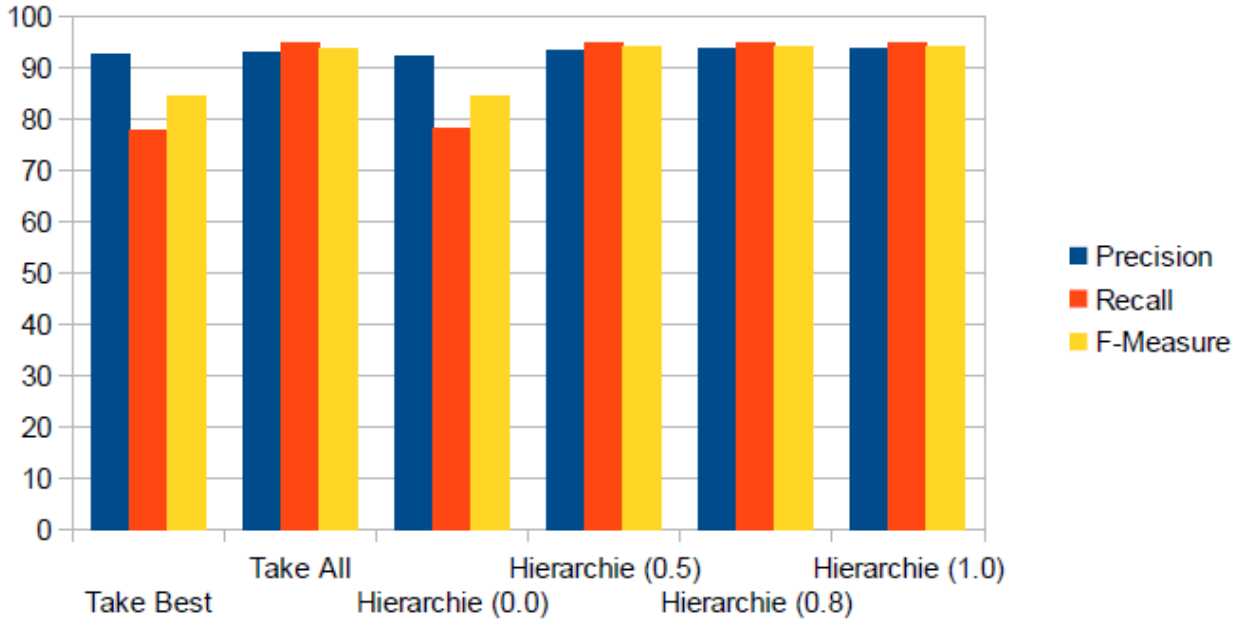

Abbildung 21: Evaluierung des Adaptierungsalgoritmus Hierarchie

Die erste Erkenntnis der Evaluierung ist, dass von den beiden bestehenden Strategien *Take All* der Strategie *Take Best* überlegen ist. Die neue Strategie liefert für *minSim* ≥ 0.5 leicht bessere Ergebnisse als *Take All* und für den Recall deutlich bessere Ergebnisse als *Take Best.* Dies wird bei der genaueren Betrachtung der Werte deutlich:

	Take Best	**Take All**	**Hierarchie (0.0)**	**Hierarchie (0.5)**	**Hierarchie (0.8)**	**Hierarchie (1.0)**
#TP	1896	2307	1900	2310	2310	2310
#FP	154	181	158	162	159	159
#FN	542	131	538	128	128	128
Precision	92.49	92.73	92.32	93.45	93.56	93.56
Recall	77.77	94.63	77.93	94.75	94.75	94.75
F-Measure	84.49	93.67	84.52	94.09	94.15	94.15

Tabelle 2: Ergebnisse der Evaluation der Adaptierungsstrategien bei Split

Von den 3 true positive Matchings, die Hierarchie ab *minSim* ≥ 0.5 mehr hat, sind zwei Left Ear und Right Ear aus dem Beispiel, dass die Motivation für diese Strategie gegeben hat. Die höchste Qualität wird mit der neuen Strategie *Hierarchie* und *minSim* = 0.8

erreicht. Eine weitere Anhebung von *minSim* führte auf den Testdaten zu keiner weiteren Verbesserung der Qualität.

Die Evaluierung hat gezeigt, dass der in dieser Arbeit vorgestellte und umgesetzte Algorithmus *Hierarchie* brauchbare Ergebnisse liefert und den bisherigen Strategien *Take All* und *Take Best* überlegen ist.

b) Strategie bei Änderung von Attributen

Die Überlegung hinter den vorgestellten Strategien *Trigramm* und *Trigramm Ignore Syn* war ein eventuell unnötiges Rematchen von Konzepten zu verhindern und dadurch die Qualität des Mappings $M_{O1',O2'}$ zu steigern, indem die bestehenden, als korrekt angenommenen, Korrespondenzen übernommen werden. Dadurch sollte eine möglichst hohe Wiederverwertung des alten Mappings zu ermöglicht werden. Dies geschah unter der Annahme, dass Attributänderungen, insbesondere von Synonymen, keinen großen Einfluss auf das Mapping zwischen den Ontologien haben. Die Evaluierung der beiden Adaptierungsstrategien erfolgte anhand des im Rahmen dieser Arbeit erstellten Benchmarks mit den Ontologien FMA und NCI in den Versionen 2009 und 2012. Dabei wurden für das maximale Ähnlichkeitsmaß (*maxSim*) verschiedene Werte (0.3, 0.5 und 0.8) festgelegt. Auch wurden die bereits bestehenden Strategien *Match All* und *Match Unmatched* das erste Mal evaluiert. Die Ergebnisse sind in *Abbildung 22* dargestellt und in *Tabelle 3* aufgeführt.

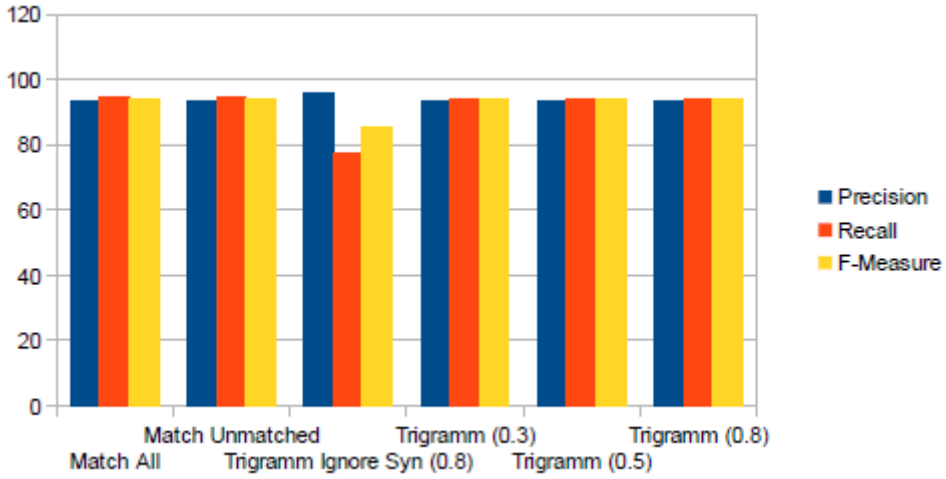

Abbildung 22: Evaluierung der Adaptierungsalgoritmen Match All, Match Unmatched, Trigramm Ignore Syn und Trigramm

	Match All	Match Unmatched	Trigramm Ignore Syn	Trigramm (0.3)	Trigramm (0.5)	Trigramm (0.8)
#TP	2307	2307	1880	2288	2291	2295
#FP	159	159	85	157	159	159
#FN	131	131	558	150	147	143
Precision	93,55	93,55	95,67	93.58	93,51	93,52
Recall	94,63	94,63	77,11	93,85	93,97	94,13
F-Measure	94,09	94,09	85,4	93,71	93,74	93,83

Tabelle 3: Ergebnisse der Evaluation der Adaptierungsstrategien bei Attributänderungen

Die Ergebnisse der Evaluierung zeigen, dass die Qualität der Ergebnisse der untersuchten Adaptierungsstrategien recht ähnlich ist. Einzig *Trigramm Ignore Syn* bietet eine höhere Genauigkeit, aber erkauft sich diese teuer mit einem deutlich niedrigerem Recall. Die Werte für von *Trigramm* nähern sich mit steigendem *maxSim* immer weiter denen von *Match All* an. Die beiden naiven Adaptierungsstrategien *Match All* und *Match Unmatched* liefern auf den Daten des Benchmarks die besten Ergebnisse bezüglich der Qualität der generierten Mappings. Interessanterweise gibt es keine Unterschiede in der Qualität der durch die beiden verschiedenen Strategien erstellten neuen Mappings.

Damit haben sich die beiden neuen Strategien *Trigramm* und *Trigramm Ignore Syn* nicht gegen die naiven Ansätze durchsetzen können. Zwar konnte mit *Trigramm Ignore Syn* eine Verbesserung der Genauigkeit (fast 50% weniger FP) erreicht werden, diese fällt aber im Vergleich zur geringeren Vollständigkeit deutlich kleiner aus.

6. Zusammenfassung und Ausblick

Im Rahmen der Arbeit wurde ein Benchmark mit verschiedenen Mappingversionen erstellt, der es ermöglicht Adaptierungsstrategien zu evaluieren. Dafür wurden 3 große Ontologien der Lebenswissenschaften (FMA, NCIt, SNOMED CT), inklusive der Mappings zwischen den Ontologien in jeweils 4 Versionen (für die Jahre 2009-2012) aus UMLS extrahiert und in ein GOMMA-Repository übertragen. Der so erstellte Benchmark ist der erste über mehrere Versionen und somit der einzige, der zur Evaluierung von Adaptierungsstrategien verwendet werden kann. Mit den dafür genutzten Werkzeugen können problemlos andere Ontologien, Mappings und Versionen aus UMLS extrahiert werden. Im zweiten Teil der Arbeit erfolgte die Konzeption und Implementierung der Strategien *Hierarchie*, *Trigramm* und *Trigramm Ignore Syn* zur evolutionsbedingten Adaptierung von Mappings zwischen Ontologien. Im dritten Teil wurden diese Strategien evaluiert und die Qualität der erstellten Mappings anhand des Benchmarks untersucht.

Der erstellte Benchmark wurde bereits in einzelnen Publikationen erfolgreich eingesetzt. Auf der einen Seite um die Auswirkungen verschiedener Änderungsoperationen auf die Mappingadaptierung zu untersuchen und auch zur Evaluierung von Mappingadaptierungsstrategien.

In fortführenden Arbeiten könnte der Benchmark um zusätzliche Ontologien, Mappings und Versionen erweitert werden. So gibt es zum Beispiel inzwischen UMLS-Versionen für 2013 und 2014. Die hier vorgestellten Adaptierungsstrategien könnten ein weiteres Mal mit den Ontologien NCIt und SNOMED CT aus dem Benchmark evaluiert werden um zu prüfen wie sie sich die deutlich größere Menge an Konzepten, Korrespondenzen und Änderungen auf die Qualität der erzeugten Mappings auswirkt. Interessant wäre dabei die Untersuchung des Kosten/Nutzen-Verhältnisses der Strategie *Trigramm Ignore Syn*. Darüber hinaus können in den Changehandlern komplett neue Strategien angelegt werden.

Die in dieser Arbeit vorgestellten Algorithmen unterstützen zwar die Berücksichtigung semantisch reichhaltigerer Mappings, aber bei den bisher erstellten Ontologiemappings handelt es sich um reine Äquivalenzmappings. Auch bei der Erstellung der Diff-Evolutionsmappings durch COnto-Diff werden die Beziehungstypen zwischen den

Konzepten unterschiedlicher Versionen nicht explizit mit einbezogen. Dabei spiegeln gerade Änderungsoperationen wie *Split* oder *Merge* eine komplexe Semantik wider. Um die Ausdrucksstärke und Korrektheit von Mappings zu verbessern, ist es deshalb sinnvoll diese semantisch anzureichern. Dafür benötigt es erweiterte Verfahren, die es ermöglichen verschiedene Beziehungsarten (*is-a* oder *part-of*, *Abbildung 1*) zu berücksichtigen. Darüber hinaus müssen semantisch reichhaltige Ausgangsmappings für COnto-Diff und Adaptierung zur Verfügung gestellt werden.

Die Adaptierung von Ontologie-Mappings in den Lebenswissenschaften bietet auch in Zukunft noch genug Ansätze für weitere Forschung.

7. Literaturverzeichnis

1. Bodenreider, O., Stevens, R.: Bio-ontologies: current trends and future directions, Briefings in bioinformatics 7 (3). (2006)

2. Lambrix, P., Tan, H., Jakoniene, V., Strömbäck, L.: Biologogical Ontologies, Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences. (2007)

3. Smith, B., et al.: The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration, Nature biotechnology 25(11). (2007)

4. Bodenreider, O.: The unified medical language system (UMLS): integrating biomedical terminology, Nucleic acids research 32(suppl 1). (2004)

5. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching, The VLDB Journal 10(4). (2001)

6. Euzenat, J., Shvaiko, P.: Ontology Matching, Springer Verlag New York (2007)

7. Rahm, E.: Towards Large Scale Schema and Ontology Matching, Schema Matching and Mapping. Springer (2011)

8. Hartung, M., Kirsten, T., Rahm, E.: Analyzing the evolution of life science ontologies and mappings, Proc. DILS. (2008)

9. Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H.,Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T.,et al.: Gene Ontology: tool for the unication of biology, Nature Genetics 25 (25-29). (2000)

10. Hartung, M., Groß, A., Kirsten, T., Rahm, E.: Discovering Evolving Regions in Life Science Ontologies, Proc. DILS. (2010)

11. Malone, J., Stevens, R.: Measuring the level of activity in community built bioontologies, J Biomed Inform 46(1). (2013)

12. Dos Reis, J., Pruski, C., Da Silveira, M., Reynaud, C.: Analyzing and Supporting the Mapping Maintenance Problem in Biomedical Knowledge Organization Systems, Proc. SIMI Workshop at ESWC. (2012)

13. Groß, A., Hartung, M., Thor, A., Rahm, E.: How do computed ontology mappings evolve? - A case study for life science ontologies, Joint Workshop on Knowledge Evolution and Ontology Dynamics. (2012)

14. Groß, A., Hartung, M., Kirsten, T., Rahm, E.: Estimating the quality of ontology-based annotations by considering evolutionary changes, Proc. DILS. (2009)

15. Groß, A., Hartung, M., Prüfer, K., Kelso, J., Rahm, E.: Impact of Ontology Evolution on

Functional Analyses, Bioinformatics 28(20). (2012)
16. Kondylakis, H., Plexousakis, D.: Ontology Evolution: Assisting Query Migration, Proc. ER. (2012)
17. Liang, Y., Alani, H., Shadbolt, N.: Changing ontology breaks queries, Proc. ISWC. (2006)
18. Noy, N., et al.: Bioportal: ontologies and integrated data resources at the click of a mouse, Nucleic acids res. 37(suppl 2). (2009)
19. Thomas R. Gruber: Toward Principles for the Design of Ontologies Used for Knowledge Sharing, International Journal Human-Computer Studies Vol. 43, Issues 5-6, p.907-928. (1995)
20. Groß, A., Dos Reis, J.C., Hartung, M., Pruski, C., Rahm, E.: Semi-Automatic Adaptation of Mappings between Life Science Ontologies, Proc. 9th Intl. Conference on Data Integration in the Life Sciences (DILS). (2013)
21. Noy, N.F., Musen, M.A.: Promptdiff: A fixed-point algorithm for comparing ontology versions, Proc. of Nat. Conf. on Artificial Intelligence. (2002)
22. Hartung, M., Gross, A., Rahm, E.: COnto-Diff: generation of complex evolution mappings for life science ontologies, Journal of Biomedical Informatics 46, (15-22). (2013)
23. Hartung, M., Terwilliger, J.F., Rahm, E.: Recent Advances in Schema and Ontology Evolution, Schema Matching and Mapping. Springer (2011)
24. Yu, C., Popa, L.: Semantic Adaptation of Schema Mapping when Schmeas Evolve, Proc. VLDB. (2005)
25. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga, R.: Logic-based assessment of the compatibility of UMLS ontology sources, Journal of Beiomedical Semantics 2(suppl 1). (2011)
26. Dos Reis, J.C., Dinh, D., Prusky, C., Da Silveira, M, Reynaud-Delaître, C.: Mapping adaptation actions for the automatic reconciliation of dynamic ontologies, CIKM '13 Proceedings of the 22nd ACM international conference on Conference on information & knowledge management (599-608). (2013)
27. Kirsten, T., Gross, A., Hartung, M., Rahm, E.: GOMMA: a component-based infrastructure for managing and analyzing life science ontologies and their evolution, Journal of Biomedical Semantics 2. (2011)

28. McGuinness, D. L., Van Harmelen, F.: OWL Web Ontology Language overview, W3C recommendation 10. (2004)

8. Abbildungs- und Tabellenverzeichnis

Abbildungen

Tabellen

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig und unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann.

Ort: Leipzig Datum: 15.04.14 Unterschrift: