

Automatische Wissensintegration mit Ontologien

York Sure, Marc Ehrig, Rudi Studer
Institut AIFB, Universität Karlsruhe (TH)
Englerstr. 11, 76128 Karlsruhe
{sure,ehrig,studer}@aifb.uni-karlsruhe.de

Abstract: In der Informatik sind Ontologien formale Modelle eines Anwendungsbereiches, die die Kommunikation zwischen menschlichen und/oder maschinellen Akteuren unterstützen und damit den Austausch und das Teilen von Wissen in Unternehmen erleichtern. Ontologien zur strukturierten Darstellung von Wissen zu nutzen hat deshalb in den letzten Jahren zunehmende Verbreitung gefunden. Schon heute existieren weltweit tausende Ontologien. Um Interoperabilität zwischen darauf aufbauenden Softwareagenten oder Webservices zu ermöglichen, ist die semantische Integration der Ontologien eine zwingendnotwendige Voraussetzung. Wie man sich leicht verdeutlichen kann, ist die rein manuelle Erstellung der Abbildungen ab einer bestimmten Größe, Komplexität und Veränderungsrate der Ontologien nicht mehr ohne weiteres möglich. Automatische oder semiautomatische Technologien müssen den Nutzer darin unterstützen. Das Integrationsproblem beschäftigt Forschung und Industrie schon seit vielen Jahren z.B. im Bereich der Datenbankintegration. Neu ist jedoch die Möglichkeit komplexe semantische Informationen, wie sie in Ontologien vorhanden sind, einzubeziehen. Zur Ontologieintegration wird in diesem Kapitel ein sechsstufiger genereller Prozess basierend auf den semantischen Strukturen eingeführt. Erweiterungen beschäftigen sich mit der Effizienz oder der optimalen Nutzereinbindung in diesen Prozess. Außerdem werden zwei Anwendungen vorgestellt, in denen dieser Prozess erfolgreich umgesetzt wurde. In einem abschließenden Fazit werden neue aktuelle Trends angesprochen. Da die Ansätze prinzipiell auf jedes Schema übertragbar sind, das eine semantische Basis enthält, geht der Einsatzbereich dieser Forschung weit über reine Ontologieanwendungen hinaus.

1 Motivation

In der Informationstechnologie wird heutzutage Wissen in einer Vielzahl von Repräsentationen dargestellt. Datenbanken, Dokumentsammlungen, Schemata wie Link- oder Ordnerstrukturen, sowie Prozessbeschreibungen. Nutzer werden mit einem hohen Maß an Heterogenität konfrontiert.

Ein Ansatz zur Überwindung dieser Heterogenität ist die Integration durch Verwendung von ausdrucksächtigen Schemata. Die Kernidee von Ontologien ist es, eine explizite Konzeptualisierung darzustellen welche von Nutzergruppen geteilt wird. Häufig wird diese Definition erweitert um formale Repräsentation sowie die Einschränkung auf eine bestimmte Domäne. Eine standardisierte Syntax wie die W3C Recommendation *OWL Web*

*Ontology Language*¹ ist ein wichtiger Schritt zur Überwindung der Heterogenität.

Nichtdestotrotz ist mit einer syntaktischen Integration das semantische Integrationsproblem noch nicht gelöst. Gleiche Sachverhalte können mit unterschiedlichen Begriffen in verschiedenen Ontologien modelliert sein. Die semantische Integration ist jedoch Grundvoraussetzung um schließlich Interoperabilität zwischen Softwareagenten oder Webservices zu gewährleisten.

Bestehende Ansätze [AS01, NM01, DDH03] nutzen semantischen Strukturen nur sehr begrenzt. Die dieser Arbeit zugrunde liegende Hypothese ist, dass durch die Ausnutzung semantischer Strukturen innerhalb der Ontologien bessere Ergebnisse erzielt werden können als mit rein syntaktischen Ansätzen. Hierzu stellen wir ein Verfahren vor, welches in praktischen Anwendungen bereits sehr vorteilhaft evaluiert wurde. Dieser Beitrag baut auf bereits existierende Arbeiten auf und fasst Sie in neuer Art zusammen. Ziel ist es in einem anschaulichen und inhaltlich abgeschlossenen Beitrag zu zeigen wie automatische Wissensintegration mit Ontologien derzeit realisiert werden kann.

Dieser Beitrag hat fünf Abschnitte. Die Einleitung legt die Probleme dar, die durch Wissensrepräsentation in mehreren Ontologien entstehen können. Im nächsten Abschnitt 2 wird zunächst erläutert wie Wissen in Form von Ontologien modelliert werden kann. In Abschnitt 3 wird ein Prozess zur (semi-) automatischen Wissensintegration mit Ontologien eingeführt. Daraufhin, in Abschnitt 4, wird in zwei Anwendungen, OntoMap und Bibster, die praktische Anwendbarkeit in unterschiedlichen Szenarien gezeigt. Schließlich wird ein Fazit zu Wissensintegration mit Ontologien gezogen und ein Ausblick auf zukünftige Trends gegeben.

2 Wissensmodellierung mit Ontologien: Ein Beispiel

Von ihrer Grundstruktur her gesehen setzen sich Ontologien i.a. aus vier Bestandteilen zusammen (vgl. auch Abb. 1, erstellt mit dem Open Source Ontologie Editor KAON²):

Das **Lexikon** enthält eine Menge von Worten (lexikalischen Einträgen, Symbolen), mit denen Begriffe und semantische Relationen bezeichnet werden. Beispiele hierfür sind `Angestellter` oder `arbeitet in`. Das Lexikon ist im Beispiel nicht separat visualisiert.

Die in einer Ontologie enthaltenen **Begriffe** charakterisieren, welche Begrifflichkeiten für einen Anwendungsbereich als relevant erachtet werden. Dabei finden sich nur die Begriffe in einer Ontologie, auf die sich eine Gruppe von Personen geeinigt haben (*“shared conceptualization”*). Beispiele für Begriffe sind `Person`, `Angestellter` und `Projekt`. Dabei können verschiedene Worte auf denselben Begriff verweisen. In unserem Beispiel bezeichnen die Worte `Angestellter` und `Mitarbeiter` beide denselben Begriff `Angestellter`.

Begriffe einer Ontologie werden durch **semantische Relationen** zueinander in Beziehung

¹<http://www.w3.org/TR/owl-guide/>

²<http://kaon.semanticweb.org>

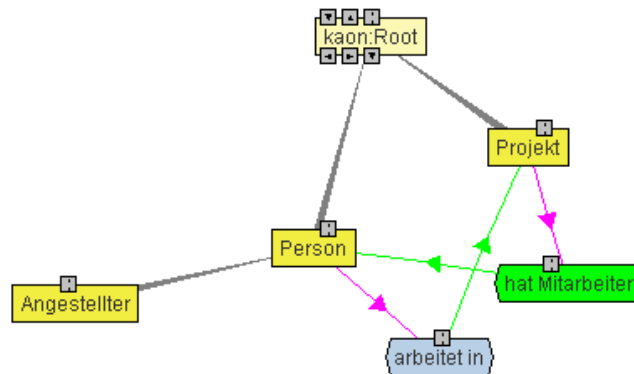


Abbildung 1: Ontologie Beispiel

gesetzt. Eine in Ontologien vordefinierte Relation ist die *is-a*-Beziehung, die einen spezielleren Begriff, den sogenannten Unterbegriff, mit einem allgemeineren Begriff, dem sogenannten Oberbegriff, in Beziehung setzt. So ist in Abb. 1 beispielsweise eine *is-a*-Beziehung zwischen dem Begriff *Angestellter* und dem Begriff *Person* definiert. Dabei ist *Angestellter* der Unterbegriff und *Person* der Oberbegriff.

Darüber hinaus können in Ontologien anwendungsspezifische Relationen zwischen Begriffen definiert werden, um weitere Bedeutungsinhalte der Begriffe zu erfassen. In unserer Beispiel-Ontologie ist z.B. eine Relation *arbeitet in* zwischen den Begriffen *Angestellter* und *Projekt* spezifiziert, die zum Ausdruck bringt, dass Angestellte in Projekten arbeiten. Umgekehrt bringt die Relation *hat Mitarbeiter* zum Ausdruck, dass Projekte Angestellte als Mitarbeiter haben.

Mit der *is-a*-Beziehung verbunden ist das sogenannte Vererbungskonzept. Vererbung besagt, dass Eigenschaften von Oberbegriffen, die durch Relationen definiert werden, auf die zugehörigen Unterbegriffe vererbt werden. So ist die Relation *hat Mitarbeiter* nicht nur für den Begriff *Projekt* sondern vielmehr auch für seine Unterbegriffe *Forschungsprojekt* und *Entwicklungsprojekt* definiert, ohne dass diese Relation bei diesen Begriffen explizit angegeben ist. Dieses Vererbungskonzept macht Ontologien kompakter, da anwendungsspezifische Relationen nur genau einmal in einer Ontologie explizit für einen Begriff spezifiziert werden und dann implizit für alle seine Unterbegriffe mitdefiniert sind.

Zusätzliche Bedeutungsinhalte von Begriffen und Relationen können durch **regelmäßige Zusammenhänge** erfasst werden. In Abb. 1 ist z.B. zu sehen, dass zwischen den Begriffen *Angestellter* und *Projekt* die semantischen Relationen *arbeitet in* sowie *hat Mitarbeiter* definiert sind. Jedem menschlichen Betrachter dieser Ontologie ist dabei klar, dass diese beiden Relationen zueinander invers sind, d.h. wenn eine *arbeitet in*-Relation besteht zwischen einer Angestellten Müller und einem Projekt Skill-Management, dann muss auch eine Relation *hat Mitarbeiter* zwischen dem Projekt Skill-Management und der Angestellten Müller existieren.

Damit derartige Zusammenhänge auch maschinell erkannt und verarbeitet werden können, beinhalten Ontologien solche regelhaften Zusammenhänge. In unserem Beispiel könnte diese Regel wie folgt spezifiziert sein: “Wenn ein Angestellter A1 in einem Projekt P1 arbeitet, dann hat das Projekt P1 den Angestellten A1 als Mitarbeiter.”

Ein anderer in unserem Anwendungsbeispiel sinnvoller regelhafter Zusammenhang könnte durch folgende weitere Regel definiert sein (vgl. Abb. 1): “Wenn ein Angestellter A1 in einem Projekt P1 arbeitet und das Projekt P1 hat das Unternehmen U1 als Kunde, dann hat der Angestellte A1 Erfahrung mit dem Kunden U1.” Die Bereitstellung solcher regelhaften Zusammenhänge hat den Vorteil, dass ein auf solchen Ontologien aufbauende Wissensmanagementanwendung Antworten bereitstellen kann, ohne dass bestimmte Sachverhalte der Wissensmanagementanwendung explizit bekannt sind. So kann die Frage nach Angestellten, die Erfahrung mit dem Unternehmen SAP hat, mit Namen von Angestellten beantwortet werden, von denen nur bekannt ist, dass sie in einem Projekt mit dem Kunden SAP mitarbeiten.

Wissensmodelle in Form von Ontologien spielen für die zukünftige Entwicklung von IT-basierten Wissensmanagement-Lösungen eine immer wichtigere Rolle, da durch die Verwendung von Ontologien Wissensinhalte aus Dokumenten erfasst und miteinander verknüpft werden können.

3 Wissensintegration

In diesem Abschnitt stellen wir nun unseren Prozess zur Wissensintegration vor. Insbesondere fokussiert sich unser Prozess auf die Abbildung zweier Ontologien aufeinander. Sobald klar ist wie die Objekte aufeinander abgebildet werden, können diese zusammengeführt und das Wissen integriert werden.

3.1 Abbildungs-Definition

Wir definieren den Abbildungsbegriff in Anlehnung an [Kle01]: Gegeben seien zwei Ontologien. Sie aufeinander abzubilden bedeutet, dass für jede Entität (Begriff, Relation oder Instanz) der ersten Ontologie eine Entsprechung mit derselben Bedeutung in der zweiten Ontologie gesucht wird.

Definition 1 (Abbildung von Ontologien) Wir definieren eine Funktion *align* auf einem Vokabular, E , aller Entitäten $e \in E$, d.h. Begriffe, Relationen und Instanzen, und der Menge möglicher Ontologien, O , als partielle Funktion:

$$\text{align} : E \times O \times O \rightarrow E,$$

$$\text{mit } \forall e \in E_{O_1} \exists f \in E_{O_2}, O_1, O_2 \in O :$$

$$\text{align}(e, O_1, O_2) = f \vee \text{align}(e, O_1, O_2) = \perp.$$

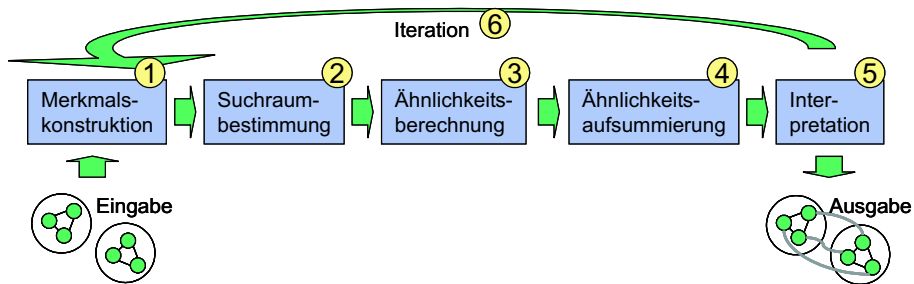


Abbildung 2: Integrationsprozess

Wir lassen O_1 und O_2 weg wo diese offensichtlich sind und schreiben stattdessen $align(e) = f$. '⊥' zeigt an, dass keine Entsprechung gefunden werden konnte. Ein Entitätenpaar, welches noch auf Identität untersucht werden muss, bezeichnen wir als Kandidaten.

3.2 Prozess

Der vorgestellte Prozess ist ursprünglich definiert in [ES04a], ein ähnlicher Ansatz wird verfolgt in [vEK04]. Viele bekannte Integrationsansätze (wie z.B. PROMPT [NM00], GLUE [DDH03] oder QOM [ES04b]) lassen sich unserem Prozess unterordnen. Eine Übersicht existierender Ansätze kann in [SE05] gefunden werden.

Abb. 2 veranschaulicht die sechs Hauptschritte des Prozesses. Als Eingabe dienen lediglich zwei beliebige Ontologien, die aufeinander abgebildet werden sollen.

1. Merkmalskonstruktion In diesem Schritt wählt man die Merkmale einer Entität innerhalb der Ontologie aus, durch die diese treffend beschrieben wird. Der Integrationsprozess könnte z.B. auf RDF(S)-Merkmalen aufgebaut sein. Eine solche Eigenschaft könnte der lexikalische Eintrag einer Entität sein. Es können aber auch intensionale Struktureigenschaften wie die *is-a*-Beziehung genutzt werden bzw. Ausgangs- und Zielmengen von Relationen. Instanzmerkmale umfassen auch die Attributwerte. Des Weiteren nutzen wir extensionale Beschreibungen. In den Beispielen 1 und 2 haben wir Ausschnitte zweier verschiedener Ontologien. Eine davon beschreibt den Begriff *Angestellter*, die andere *Mitarbeiter*. Sowohl *o1:Angestellter* und *o2:Mitarbeiter* besitzen ein generelles Merkmal `rdfs:subClass`. Die Werte hiervon sind `Person` und `Unternehmenskapital` bzw. nur `Person`.

2. Suchraumbestimmung Zur Ableitung von Abbildungen in Ontologien wird ein Suchraum von Kandidaten aufgestellt. Dieser Suchraum kann weiter eingeschränkt werden, so dass nur manche (sinnvolle) Kandidaten für einen Vergleich zugelassen und andere aussortiert werden. Für unser Beispiel nehmen wir jedoch erst einmal alle

```

<rdf:Description rdf:about=''o1:Angestellter''>
  <rdfs:subClass rdf:resource=''arbeit:Person''>
  <rdfs:subClass rdf:resource=''arbeit:Unternehmenskapital''>
</rdf:Description>

```

Beispiel 1. Ausschnitt der Ersten Beispielontologie.

```

<rdf:Description rdf:about=''o2:Mitarbeiter''>
  <rdfs:subClass rdf:resource=''arbeit:Person''>
</rdf:Description>

```

Beispiel 2. Ausschnitt der Zweiten Beispielontologie.

möglichen Entitätenpaare des gleichen Typs. Dies bedeutet, dass wir mit dem Vergleich von `o1:Angestellter` und `o2:Mitarbeiter` fortfahren werden.

3. Ähnlichkeitsberechnung Für die Ähnlichkeitsberechnung der Kandidaten nutzen wir heuristische Maße. Dazu benötigen wir Ähnlichkeitsfunktionen für Strings [Lev65], für Einzelobjekte oder Objektmengen [EHHS05], sowie Untersuchungen auf Unähnlichkeit oder Einschluss anstatt logischer Gleichheit. Dies ist notwendig, weil Objekte selten exakt identisch, sondern vielmehr ähnlich sind. In unserem Beispiel nutzen wir Ähnlichkeiten von Instanzen. Im Detail untersuchen wir, ob die zwei Begriffsmengen, die Oberbegriffe von `o1:Angestellter` (`Person` und `Unternehmenskapital`) und die Oberbegriffe von `o2:Mitarbeiter` (nur `Person`) gleich sind. Für das Beispiel stimmt dies nur partiell, wodurch die Ähnlichkeit auf 0,5 gesetzt wird. Die zugehörige Eigenschaft/Ähnlichkeits-Regel (FS3) wird in Tabelle 1 dargestellt: Wenn die Oberbegriffe gleich sind, sind auch die Instanzen ähnlich. Zusätzlich sind noch weitere Regeln angegeben, so z.B. Regel FS1, die die Ähnlichkeit der Bezeichner untersucht.

Tabelle 1: Eigenschaften und Ähnlichkeiten für verschiedene Entitätentypen.

Typ	Nr.	Eigenschaft	Ähnlichkeit
Entitäten	FS1	(Bezeichner, X_1)	Stringähn. (X_1, X_2)
Begriffe	FS2	(Relationen, Y_1)	Mengenähn. (Y_1, Y_2)
	FS3	(Oberbegriffe, Y_1)	Mengenähn. (Y_1, Y_2)
	FS4	(Unterbegriffe, Y_1)	Mengenähn. (Y_1, Y_2)
	FS5	(Instanzen, Y_1)	Mengenähn. (Y_1, Y_2)
Relationen	FS6	(Domain, X_{d1}) and (Range, X_{r1})	Objektähn. (X_{d1}, X_{d2}), (X_{r1}, X_{r2})
	FS7	(Relationeninstanzen, Y_1)	Mengenähn. (Y_1, Y_2)
Instanzen	FS8	(Elternbegriffe, Y_1)	Mengenähn. (Y_1, Y_2)
	FS9	(Relationeninstanzen, Y_1)	Mengenähn. (Y_1, Y_2)
...

4. Ähnlichkeitsaufsummierung Normalerweise gibt es mehrere Ähnlichkeitswerte für ein Entitätenpaar, beispielsweise basierend auf den lexikalischen Einträgen

oder der Ähnlichkeit der Relationen zu anderen Entitäten. Diese individuellen Ähnlichkeiten müssen aggregiert werden. Dies kann durch eine simple Durchschnittsbildung erreicht werden, allerdings auch durch komplexe Aggregierungsfunktionen mit Gewichten für jede einzelne Ähnlichkeit. In unserem Beispiel haben wir nur $\text{sim}(o1:\text{Angestellter}, o2:\text{Mitarbeiter})=0,5$.

5. Interpretation Auf Basis der aufsummierten Ähnlichkeitswerte werden die Entitäten nun aufeinander abgebildet. Mögliche Vorgehensweisen sind ein Schwellwert [NM01], sogenanntes Relaxation Labelling [DDH03] oder eine Mischung aus Strukturen und Ähnlichkeitswerten. $\text{sim}(o1:\text{Angestellter}, o2:\text{Mitarbeiter})=0,5 \geq 0,5$ führt somit zu $\text{align}(o1:\text{Angestellter})=o2:\text{Mitarbeiter}$. Semiautomatische Methoden präsentieren den Nutzern die Entitäten mit ihren Ähnlichkeiten und überlassen diesem die Interpretation. Diese Eingaben können nachfolgend für bessere Berechnungen hinzugezogen werden.

6. Iteration Einige Algorithmen wiederholen die bisher beschriebenen Schritte um die Struktur der Ontologien besser zu nutzen (siehe hierzu auch [MGMR02]). Ähnlichkeiten von benachbarten Entitäten beeinflussen die Ähnlichkeit der eigentlichen Objekte. Dieser Wiederholungsprozess wird abgebrochen sobald keine weiteren Abbildungen gefunden werden, oder wenn alternativ ein Maximalwert an Runden erreicht wird. Man sollte beachten, dass nicht alle fünf vorhergehenden Schritte notwendigerweise in dem Wiederholungszyklus enthalten sein müssen, sei es weil die zu untersuchenden Merkmale nur einmal festgelegt werden müssen oder weil manche Ähnlichkeitsmaße nur einmal berechnet werden müssen. Die Ergebnisse einer Iteration, d.h. die gefundenen Identitäten, werden dem System zurückübergeben und können somit für eine bessere Berechnung in nachfolgenden Iterationen genutzt werden.

Schließlich wird das Gesamtergebnis gespeichert. Dieses Ergebnis repräsentiert die Relation align_{O_1, O_2} .

3.3 Erweiterungen

Zahlreiche Erweiterungen können diesen Standardprozess verbessern. Sowohl der Standardprozess als auch dessen Erweiterungen sind im Rahmen des Tools FOAM³ umgesetzt worden.

QOM – Quick Ontology Mapping In dem Ansatz von [ES04b] wird auf das Effizienzproblem eingegangen, das bei größeren Ontologien auftritt. Dabei werden die Besonderheiten von Ontologiestrukturen ausgenutzt. Insbesondere wird die Anzahl der zu vergleichenden Paare deutlich eingeschränkt, indem nur solche zugelassen werden, die entweder

³<http://www.aifb.uni-karlsruhe.de/WBS/meh/foam>

sehr ähnliche Bezeichner haben oder in unmittelbarer Nähe schon bestehender Abbildungen liegen. Auch werden als Merkmale nur solche erlaubt, die nicht eine komplette Traversierung der Ontologie benötigen, z.B. werden nur direkte Instanzen eines Begriffes untersucht, anstatt der Aufsummierung von allen Instanzen aller Unterbegriffe. Sowohl auf theoretischer als auch auf praktischer Ebene lässt sich der Prozess somit deutlich beschleunigen.

APFEL Schon die Auswahl der zu untersuchenden Merkmale und der zugehörigen Ähnlichkeitsmaße ist sehr schwierig. Das zusätzliche optimale Setzen der Gewichte ist selbst für Ontologieexperten nahezu unmöglich. APFEL [ESS05] beschreibt deshalb einen Ansatz, bei dem diese Aufgaben durch maschinelles Lernen gelöst werden. Der Nutzer muss lediglich Ontologien und die einige korrekte Abbildungen vorgeben. Der gelernte Entscheidungsbaum wird nachher zur Aufsummierung und Interpretation ausgewählter Merkmale genutzt.

Interaktive Integration Die beschriebenen Prozesse sind auf ein vollautomatisches Verfahren ausgerichtet. Gegebenenfalls ergibt es jedoch sehr wohl Sinn den Nutzer gegebenenfalls miteinzubeziehen. Durch geschickt gestellte Fragen an diesen wird sein Aufwand minimiert bei gleichzeitigem maximalen Nutzen für die Anwendung. Hierzu werden dem Nutzer nur die Kandidaten präsentiert, bei denen das System den größten Mehrwert durch die Nutzereingabe erwartet. Diese sind diejenigen Paare, die eine Ähnlichkeitswert nahe dem Schwellwert besitzen und sehr stark in der Ontologie vernetzt sind. Der Nutzer ordnet diese manuell in Abzubildende und Nicht-Abzubildende ein. Die Qualität der identifizierten Abbildungen lässt sich hierdurch nochmals deutlich verbessern.

Adaptive Integration Die Untersuchung mehrerer Anwendungsszenarien [dBF05] hat gezeigt, dass diese teilweise sehr unterschiedliche Anforderungen an den Abbildungsalgorithmus haben. Nach Erfassung der Eigenheiten wurde der Algorithmus so angepasst, dass nach Eingabe des Szenarios (Abbildungen Erstellen, Integration, Zusammenführen, Evolution, usw.) die Parameter des Prozesses automatisch gewählt werden. Auch die Eigenschaften der Ontologien werden in Betracht gezogen, so dass sehr große Ontologien eine wesentlich effizientere Implementierung nutzen als kleine Ontologien, die eine sehr detaillierte Suche erlauben. Der Prozess kann somit für viele Anwendungen eingesetzt werden.

Dieser Abschnitt hat gezeigt wie Wissensintegration mit Ontologien anhand eines ausgearbeiteten Prozesses durchgeführt werden kann.

4 Anwendungen

In diesem Abschnitt wird beispielhaft anhand zweier praktischer Anwendungen der Einsatz von Techniken zur Wissensintegration aufgezeigt.

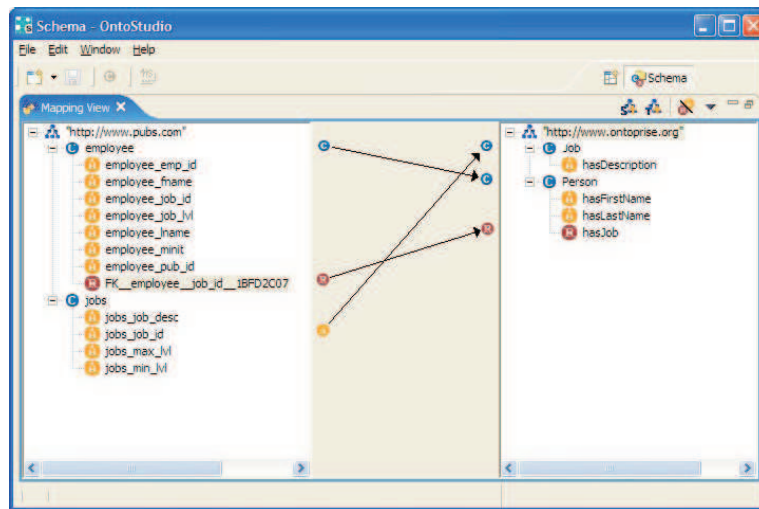


Abbildung 3: OntoMap

4.1 OntoMap – Graphisches Mapping von Ontologien

OntoStudio ist eine Entwicklungsumgebung zur Modellierung von Ontologien und zum Aufbau semantischer Anwendungen und eine Weiterentwicklung von OntoEdit [SAS03, S⁺02]. OntoStudio ist modular aufgebaut und basiert auf Eclipse von IBM, das als Quasi-Standard für Editoren-Frameworks gilt. Es unterstützt die Ontologiestandards RDF(S), OWL und F-Logic. Des Weiteren erlaubt es das Einbinden von existierenden Datenbankschemata indem diese in Ontologien umgewandelt werden. Ein weiteres Plug-in zu OntoStudio ist OntoMap [MSS03], das die Erstellung und Verwaltung von Abbildungen zwischen mehreren Ontologien über eine graphische Benutzeroberfläche erlaubt.

In Abb. 3 wird diese Oberfläche angezeigt. Die Ontologie auf der linken Seite wird mit der Ontologie auf der rechten Seite über Pfeile verknüpft. Nutzer brauchen sich über die logische Repräsentation keine Gedanken machen, sondern können mit einfachen Drag-and-Drop Aktionen die Entsprechungen der zwei Ontologien verbinden. Weitere automatische Konsistenzüberprüfungen helfen mögliche Fehler zu identifizieren. In OntoMap können Begriffe, Relationen, Instanzen und Attribute aufeinander abgebildet werden.

Diese Abbildungen werden dann als logische Regeln im Hintergrund abgelegt. Bei Anfragen an die beteiligten Ontologien werden diese im Hintergrund ausgeführt und die Ergebnisse transparent angezeigt. Dem Nutzer zeigt sich das Wissen in einer homogenen integrierten Form.

Nichtsdestotrotz ist die manuelle Erstellung von Abbildungen eine zeitaufwändige Tätigkeit. Der im vorherigen Abschnitt vorgestellte automatische Prozess kann deshalb über OntoMap aufgerufen werden. Durch simple Anklicken wird der Prozess gestartet. Innerhalb kurzer Zeit werden die Ergebnisse in der gewohnten Benutzeroberfläche ange-

zeigt. Anschließend kann der Nutzer diese anpassen und innerhalb OntoStudios ausführen lassen.

Die Integration von FOAM reduziert den manuellen Aufwand für das Erstellen von Mappings je nach Szenario erheblich. Diese Anwendung wurde im Rahmen des EU-Projektes SEKT⁴ eingesetzt und weiterentwickelt.

4.2 Bibster – P2P-basierter Austausch von bibliographischen Metadaten

Bibster[HSB⁺04] ist ein Peer-to-Peer-basiertes System, das den Austausch von bibliographischen Metadaten über Arbeitsgruppen und Organisationen hinaus ermöglicht. Bibster behandelt ein typisches Szenario eines Forschers, der regelmäßig nach neuen Veröffentlichungen sucht und die genauen Daten dieser benötigt. Bisher war es notwendig, vorhandene zentrale Internetseiten wie DBLP oder CiteSeer aufzurufen und zu hoffen, die benötigten (aktuellen) Informationen zu finden. Das neue System erlaubt es den eigenen Computer, Gruppen von Rechnern oder das ganze dezentrale Peernetzwerk nach den Informationen zu durchsuchen. Die Anfragen sind nicht auf Stichworte beschränkt, sondern erlauben komplexe semantische Relationen einzubauen, unter anderem auch die Auswahl des gewünschten Themengebietes. Ferner können die gefundenen Ergebnisse geändert werden und in die eigene Wissensbasis integriert werden. Bibster ist auf dem SWAP-System⁵ aufgebaut, das Wissen in Form von Ontologien repräsentiert. Dieses Wissen wird weiterhin genutzt um Anfragen effizient im Peernetzwerk zu weiterzuleiten. Außerdem erlaubt diese Repräsentation das Beantworten von komplexen Fragestellungen. Da die Forscher dieses Wissen sowieso bei sich verwalten, ist ein unmittelbarer Nutzen durch leichtere Suchbarkeit gewährleistet. Die Suche auch auf anderen Peers ist ein direkter Mehrwert.

Wenn man in diesem Szenario Anfragen stellt, erhält man gewöhnlich eine große Anzahl an Ergebnissen, wovon ein großer Teil Duplikate sind. Dies geschieht, weil Informationen dezentral und redundant auf den verschiedenen Peers gespeichert werden. Um dem Nutzer eine möglichst einfache Bedienung zu ermöglichen, müssen diese Duplikate herausgefiltert werden und als ein integrierter Eintrag angezeigt werden. Hierzu müssen die Duplikate identifiziert werden, welches wiederum auf unserem beschriebenen Prozess basiert. Publikationen, aber auch die beteiligten Personen und Organisationen, werden auf Basis ihrer semantischen Eigenschaften miteinander verglichen. Dabei können in diesem festgelegten Anwendungsszenario einzelne spezielle Eigenschaften wie Autoren oder Publikationstyp besonders gewichtet werden. Als Ergebnis steht schließlich eine Liste mit Duplikaten zur Verfügung. In einem letzten Schritt werden diese Duplikate noch integriert. Dabei wird jeweils der am häufigsten auftretende Wert für eine bestimmte Relation bzw. der ausführlichste Eintrag übernommen.

Bibster ist Open Source Software und kann über das Internet heruntergeladen werden.⁶

⁴<http://www.sekt-projekt.com>

⁵<http://swap.semanticweb.org>

⁶<http://bibster.semanticweb.org>

5 Fazit und aktuelle Trends

Wissensintegration ist eine Grundvoraussetzung um Interoperabilität zwischen verschiedenen Anwendungen zu ermöglichen. In diesem Beitrag haben wir einen wichtigen Beitrag zur Wissensintegration mit Ontologien geleistet. Insbesondere sind wir auf das Problem der Heterogenität eingegangen. Zunächst haben wir Ontologien vorgestellt, die als gemeinsame explizite und formale Schemata dienen können. Der zweite Teil dieses Beitrages beschäftigte sich mit der semantischen Heterogenität, die durch abweichende Modellierung entstehen kann, beispielsweise durch unterschiedliche Terme. Hierzu haben wir einen sechsstufigen Prozess präsentiert, der durch Untersuchung der Ähnlichkeit von semantischen Eigenschaften in der Ontologie gleiche Entitäten herausfinden kann. In zwei Beispielanwendungen wurde die praktische Funktionsweise eines solchen Ansatzes aufgezeigt.

Weitere aktuelle Trends im Bereich der (semi-)automatischen Wissensintegration sind der Umgang mit komplexen Abbildungen, wenn beispielsweise ein Begriff durch einen anderen, mittels Relation eingeschränkten, Begriff dargestellt werden muss. Genauso ist der Umgang mit probabilistischen oder Fuzzyontologien ein offenes Forschungsthema, auch bezüglich der Integration. Schließlich sind Strukturen nicht statischer Natur. Deshalb muss die Integration mit Veränderungen in den zugrundeliegenden Wissensstrukturen umgehen können.

Acknowledgements: Wir danken insbesondere Steffen Staab (Universität Koblenz) für die stets sehr kooperative and zielführende Zusammenarbeit sowie unseren Kolleginnen und Kollegen in den Projekten SWAP und SEKT für die tatkräftige Unterstützung. Teile der hier vorgestellten Arbeiten wurden von der EU in den Projekten IST-2001-34103 SWAP und IST-2003-506826 SEKT finanziert.

Literatur

- [AS01] R. Agrawal und R. Srikant. On integrating catalogs. In *Proc. of the 10th Int. World Wide Web Conf. (WWW-10)*, Seiten 603–612, Hong Kong, Hong Kong, 2001. ACM Press.
- [dBF05] J. de Bruijn und C. Feier. SEKT Deliverable D4.6.1: Report on Ontology Mediation for Case Studies. Bericht, June 2005.
- [DDH03] A. Doan, P. Domingos und A. Halevy. Learning to match the schemas of data sources: A multistrategy approach. *VLDB Journal*, 50:279–301, 2003.
- [EHHS05] M. Ehrig, P. Haase, M. Hefke und N. Stojanovic. Similarity for Ontologies - A Comprehensive Framework. In D. Bartmann et al., Hrsg., *Proc. of the 13th European Conf. on Information Systems (ECIS)*, Regensburg, Germany, May 2005.
- [ES04a] Marc Ehrig und York Sure. Ontology Mapping – An Integrated Approach. In Christoph Bussler, John Davis, Dieter Fensel und Rudi Studer, Hrsg., *Proceedings of the First European Semantic Web Symposium (ESWS-2004)*, Jgg. 3053 of *Lecture Notes in Computer Science*, Seiten 76–91, Heraklion, Greece, May 2004. Springer.

- [ES04b] M. Ehrig und S. Staab. QOM – Quick Ontology Mapping. In F. van Harmelen, S. McIlraith und D. Plexousakis, Hrsg., *Proc. of the 3rd Int. Semantic Web Conf. (ISWC-2004)*, LNCS, Seiten 683–696, Hiroshima, Japan, 2004. Springer.
- [ESS05] M. Ehrig, S. Staab und Y. Sure. Bootstrapping Ontology Alignment Methods with APFEL. In Y. Gil, E. Motta und V. R. Benjamins, Hrsg., *Proc. of the 4th Int. Semantic Web Conf. (ISWC-2005)*, LNCS, Galway, Ireland, November 2005. Springer.
- [HSB⁺04] P. Haase, B. Schnizler, J. Broekstra, M. Ehrig, F. van Harmelen, M. Menken, P. Mika, M. Plechawski, P. Pyszlak, R. Siebes, S. Staab und C. Tempich. Bibster – a semantics-based bibliographic Peer-to-Peer system. *Journal of Web Semantics*, 2(1):99–103, 2004.
- [Kle01] M. Klein. Combining and relating ontologies: an analysis of problems and solutions. In A. Gómez-Pérez et al., Hrsg., *Proc. of WS on Ontologies and Inf. Sharing at IJCAI-01*, Seattle, WA, USA, August 2001.
- [Lev65] I. V. Levenshtein. Binary Codes capable of correcting deletions, insertions, and reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848, 1965.
- [MGMR02] S. Melnik, H. Garcia-Molina und E. Rahm. Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching. In *Proc. of the 18th Int. Conf. on Data Engineering (ICDE-2002)*, Seite 117. IEEE Computer Society, 2002.
- [MSS03] A. Maier, H.-P. Schnurr und Y. Sure. Ontology-based Information Integration in the Automotive Industry. In D. Fensel, K. Sycara und J. Mylopoulos, Hrsg., *Proc. of the 2nd Int. Semantic Web Conf.: The Semantic Web (ISWC-2003)*, Jgg. 2870 of LNCS, Seiten 897–912, Sanibel Island, FL, USA, 2003. Springer.
- [NM00] N. F. Noy und M. A. Musen. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *Proc. of the 17th National Conf. on Artificial Intelligence (AAAI-2000)*, Seiten 450–455, Austin, TX, USA, July 2000. AAAI Press / The MIT Press.
- [NM01] N. F. Noy und M. A. Musen. Anchor-PROMPT: Using Non-Local Context for Semantic Matching. In *WS on Ontologies and Information Sharing at the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI-2001)*, Seiten 63–70, Seattle, WA, USA, August 2001.
- [S⁺02] Y. Sure et al. OntoEdit: Collaborative Ontology Development for the Semantic Web. In I. Horrocks und J.A. Hendler, Hrsg., *Proc. of the Int. Semantic Web Conf.: The Semantic Web (ISWC2002)*, Jgg. 2342 of LNCS, Seiten 221–235, Sardinia, Italy, 2002. Springer.
- [SAS03] Y. Sure, J. Angele und S. Staab. OntoEdit: Multifaceted Inferencing for Ontology Engineering. *Journal on Data Semantics, LNCS*, 2800:128–152, 2003.
- [SE05] Pavel Shvaiko und Jérôme Euzenat. A Survey of Schema-based Matching Approaches. *Journal on Data Semantics (JoDS)*, IV, 2005.
- [vEK04] L. van Elst und M. Kiesel. Generating and Integrating Evidence for Ontology Mappings. In *14th Int. Conf. on Knowledge Engineering and Knowledge Management (EKAW)*, LNCS 3257. Springer, 2004.