mehr zum thema:
www.w3.org/2001/sw/

ONTOLOGIEN IM PRAXISEINSATZ: ERFAHRUNGEN BEI DER AUDI AG

Bereits 1998 beschrieb Tim Berners Lee seine Vision vom „Semantic Web", einer Weiterentwicklung des WWW, in dem Computer Informationen verstehen und diese zueinander in Beziehung setzen können. Dabei spielen Ontologien die entscheidende Rolle: Sie bieten eine formale und von Maschinen verarbeitbare Repräsentation für Informationen. Mittlerweile existieren Standards für die Repräsentation von Ontologien, Vorgehensmodelle für deren Entwicklung sowie robuste Modellierungswerkzeuge. Ontologien gewinnen damit auch in der Unternehmenswelt immer mehr an Interesse. Audi setzt Ontologien bereits seit mehreren Jahren in unterschiedlichen Anwendungen der Fahrzeugentwicklung ein. Dieser Beitrag führt in die Technologie ein und stellt ein aktuelles Projekt vor. Abschließend werden entscheidende Faktoren für den erfolgreichen Einsatz von Ontologien in Unternehmen diskutiert.

die autoren

Dr. Claudia Heß
(E-Mail: claudia.hess@systec-cax.de)
arbeitet bei der SysTec-CAx GmbH im Bereich Innovative Technologien.

Julian Lambertz
(E-Mail: julian.lambertz@systec-cax.de)
arbeitet bei der SysTec-CAx GmbH im Bereich Innovative Technologien.

Thomas Syldatke
(E-Mail: thomas.syldatke@audi.de)
ist bei der Audio AG in Ingolstadt als CAx-Architekt mit der Integration von Informationsbausteinen aus den Bereichen CAD, CAE und CAT beschäftigt und hat dazu unter anderem das hier vorgestellte Modell entwickelt.

Wunderwaffe „Ontologien"?

Bezüglich der Nutzung, Vernetzung und Sicherung von Wissen stehen viele Unternehmen vor ähnlichen Problemen: Daten und Informationen sind verteilt und in den unterschiedlichsten Formaten und Systemen abgelegt, beispielsweise in Wikis, Office-Dokumenten, XML-Dateien, PDF-Dateien oder Datenbanken. Ein Großteil des eigentlichen Wissens existiert außerdem nur in den Köpfen der Mitarbeiter und – je nach Unternehmensbereich – gibt es spezielle Vokabulare. Eine effiziente Nutzung des gesammelten Unternehmenswissens scheitert häufig am fehlenden Überblick, welches Wissen wo vorhanden ist und wie es integriert werden kann. In den letzten Jahren hat die Forschung im Bereich *Semantic Web* eine Reihe von Technologien, Methoden und Standards entwickelt, die genau dieses Problem adressieren. Wir interessieren uns in diesem Kontext insbesondere für die Aspekte Ontologien und Regeln.

Der Begriff Ontologie stammt aus der Philosophie und bezeichnet die Lehre vom Sein. Sie fragt nach den grundsätzlichen und allgemeinen Strukturen der Wirklichkeit. In den 90er Jahren wurde der Begriff in die Informatik übernommen. Er wird – entgegen seiner ursprünglichen Form – nun auch im Plural verwendet.

Ontologien haben in der Informatik keinen Anspruch auf Allgemeingültigkeit. Stattdessen beziehen sie sich auf konkrete Anwendungsdomänen und beschreiben Fachterminologien. Ziel ist, dass ein Computer die Inhalte und Zusammenhänge, also die Semantik, „versteht". Dafür werden alle relevanten Begriffe der Anwendungsdomäne in einer Modellierungssprache für Ontologien maschinenverarbeitbar definiert und in Beziehung zueinander gesetzt – es entsteht eine Konzeptualisierung der Domäne. So können auch dezentrale und heterogene Wissensquellen beschrieben und vernetzt werden. Zusätzlich kann Expertenwissen in die Ontologie aufgenommen werden. Die Ontologie schafft somit eine einheitliche Sicht auf vorhandene Informationen (für einen Überblick und weiterführende Links zum Thema Ontologie vgl. z. B. [Gru08]).

Da die Modellierungssprachen für Ontologien auf formaler Logik basieren, können die aus der Logik bekannten Inferenzmechanismen genutzt werden (vgl. z. B. [Baa03]). Daraus kann ein Computer dann sogar intelligente Schlussfolgerungen ziehen und beispielsweise implizit modellierte Abhängigkeiten explizit machen. Sollen komplexe, funktionale bzw. bedingte Abhängigkeiten formuliert werden, ergänzen Regeln die Ausdrucksmächtigkeit von Ontologien. Auch ohne die explizite Verbindung über Prädikate bewirken sie eine Vernetzung von Begriffen.. So können kausale Aussagen der Form „Wenn A, dann B" modelliert werden. In Kombination mit Ontologien kann durch Regeln weiteres Herleitungswissen formalisiert werden.

Ontologie-Entwicklung

Im Folgenden beschreiben wir, wie Ontologien aufgebaut sind, und stellen die wichtigsten Modellierungssprachen vor.

Grundbausteine

Ontologien setzen sich aus den folgenden Grundbausteinen zusammen:

- *Klassen*, auch als Konzepte bezeichnet, fassen gleiche Dinge unter einem Begriff zusammen. Sie werden für gewöhnlich hierarchisch in so genannten Taxonomien strukturiert.
- *Individuen* sind bestimmte Ausprägungen einer Klasse.
- *Eigenschaften* werden für Klassen definiert. Sie werden genutzt, um Individuen dieser Klasse näher zu beschreiben.
- *Beziehungen* definieren Verbindungen zwischen zwei Klassen. Sie stehen

NUR für Abonnenten: Haben Sie schon Ihren Log-In für den kostenlosen Download aller Artikel-Pdfs? Kontakt: Susanne.Herl@sigs-datacom.de!

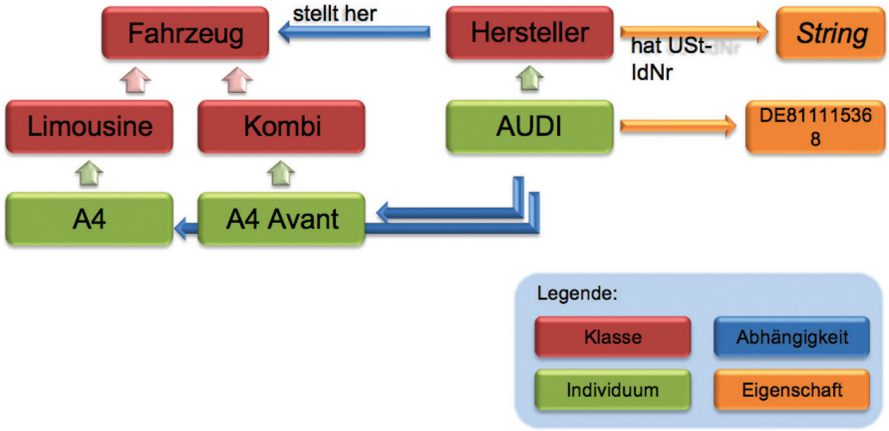

Abb. 1: Beispiel zur ontologischen Modellierung.

damit zur Verfügung, um zwei konkrete Individuen zu verknüpfen.

Abbildung 1 illustriert die Modellierung anhand eines Beispiels. Die Abbildung zeigt die Klasse Fahrzeug mit den Unterklassen Limousine und Kombi sowie die Klasse Hersteller. Unterklassen erben die für die Oberklasse definierten Eigenschaften und Beziehungen. So ist definiert, dass ein Hersteller ein Fahrzeug herstellt. Das gilt dementsprechend auch für eine Limousine und einen Kombi. Außerdem ist für einen Hersteller die Eigenschaft definiert, dass er eine Umsatzsteuer-Identifikationsnummer besitzt. Dieser Teil der Ontologie stellt die grundsätzlichen Begriffe und Beziehungen zur Beschreibung von Sachverhalten bereit. Hierauf aufbauend werden die tatsächlichen Sachverhalte definiert: Es kommen die realen Ausprägungen der Klassen – d. h. die Individuen – sowie die für sie geltenden Eigenschaften und Abhängigkeiten hinzu. Das Beispiel definiert den A4 als ein Individuum der Klasse Limousine und den A4 Avant als ein Individuum vom Typ Kombi. Außerdem wird modelliert, dass Audi ein Hersteller ist und den A4 sowie den A4 Avant herstellt. Auch die Eigenschaft hat USt-IdNr hat eine konkrete Ausprägung.

Neben der formalen Beschreibung der Anwendungsdomäne ist das Schlussfolgern – auch Inferenz genannt – die eigentliche Stärke von Ontologien. Ohne auf die Details der zu Grunde liegenden formalen Logik einzugehen, soll ein Beispiel das Prinzip verdeutlichen.

Angenommen, unsere Beispiel-Ontologie wird in einem allgemeineren Kontext benötigt, in dem es verschiedenste Arten von Produkten gibt. Dafür wird sie um eine Oberklasse Produkt über Fahrzeug ergänzt sowie um die Beziehung, dass Hersteller Produkte herstellen. Weiter wird Automobilhersteller als Unterklasse von Hersteller eingeführt und es wird modelliert, dass Automobilhersteller Fahrzeuge herstellen. Auf dieser Basis schlussfolgert eine Inferenzmaschine, dass Audi ein Automobilhersteller ist, da der A4 und der A4 Avant von dem Unternehmen hergestellt werden und implizit Fahrzeuge sind. In diesem Beispiel mögen die Ableitungen trivial und offensichtlich erscheinen. In realen Projekten enthält die Ontologie allerdings nicht nur eine Handvoll Klassen und Individuen. Nach unseren Erfahrungen kann die Zahl schon in kleineren prototypischen Anwendungen in Unternehmen in die Zehntausende steigen. Mit Hilfe einer Inferenzmaschine können damit nicht nur explizit definierte Relationen erkannt, sondern auch implizite Zusammenhänge geschlussfolgert und genutzt werden.

Ontologien und Regeln

Anhand der gezeigten Beispiel-Ontologie kann auch der Einsatz von Regeln illustriert werden. Eine Regel könnte zum Beispiel besagen: Wenn ein Automobilhersteller ein Fahrzeug herstellt, so entwickelt dieser auch das Fahrzeug. Hieraus lässt sich – obwohl nicht explizit modelliert – ableiten, dass der A4 und der A4 Avant von Audi entwickelt werden, denn der Automobilhersteller Audi stellt beide Modelle her und beides sind Fahrzeuge.

Ontologische Modellierung

Zur Modellierung einer Ontologie werden typischerweise zunächst Interviews mit Experten des jeweiligen Anwendungsbereichs geführt. Aus den Aussagen werden – ähnlich wie bei der Erstellung von Datenbank-Modellen – die wesentlichen Objekte bzw. Klassen sowie deren Attribute und Beziehungen herausgefiltert. Im Unterschied zur Datenbank-Modellierung wird bei der Ontologie-Modellierung keinerlei Rücksicht auf die geplante Verarbeitungslogik genommen, sondern der Anwendungsbereich wird lediglich formal beschrieben. Welche Antworten die so modellierte Ontologie später liefert, hängt vom Umfang der definierten Klassen und Individuen sowie deren Beziehungsdichte ab. Die Nutzung des modellierten Wissens ist von Anfang an möglich, was eine iterative Vorgehensweise erlaubt. Mit zunehmender „Füllung" der Ontologie können immer komplexere Anfragen verarbeitet werden.

Modellierungssprachen

Im Rahmen der Semantic-Web-Aktivitäten des World Wide Web Consortiums (W3C) wurde eine Reihe formaler Sprachen zur Beschreibung von Informationen standardisiert (vgl. [W3C01]). Das berühmte Schichtenmodell in Abbildung 2, der *Layercake* (vgl. [W3C07]), stellt diese gemeinsam dar. Sie nutzen grundlegende Technologien wie *Unique Resource Identifier (URLs)* und *XML*. Das *Resource Description Framework (RDF)* (vgl. [W3C-a]) erlaubt es, Aussagen über Ressourcen, wie z. B. Web-Seiten, zu treffen und diese mit Metadaten zu annotieren. Der derzeitige W3C-Standard zur Repräsentation von Ontologien ist die *Web*

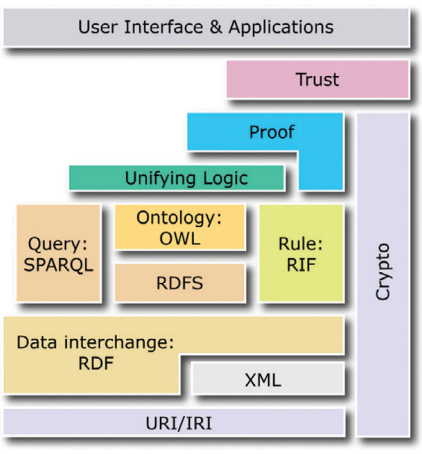

Abb. 2: Der Layercake.

Abb. 3: Das Schichtenmodell der CAx-Architektur.

Ontology Language (OWL) (vgl. [W3C04]). In der OWL finden sich die oben beschriebenen Grundbausteine „Klassen", „Individuen", „Eigenschaften" und „Abhängigkeiten" direkt wieder. Mit *OWL Lite*, *OWL DL* und *OWL Full* gibt es verschiedene Varianten der Sprache mit steigender Ausdrucksmächtigkeit. Der Nachfolger *OWL2* wird aktuell entwickelt.

Für die Kombination von Ontologien und Regeln gibt es eine Reihe von Ansätzen, allerdings ist dies noch ein Forschungsthema. So wurde zum Beispiel die *Semantic Web Rule Language (SWRL)* (vgl. [W3C-b]) vorgeschlagen, die die OWL um Regeln erweitert. Die SWRL konnte sich bisher allerdings nicht durchsetzen, da mit ihr einerseits nur relativ einfache Regeln formuliert werden können, andererseits aber logische Schlussfolgerungen algorithmisch schwer sind. Zur Modellierung von Regeln und Ontologien wird auch *F-Logic* (vgl. [Kif05]) eingesetzt, eine Sprache zur Logikprogrammierung. Aktuell wird das *Rule Interchange Format (RIF)* vom W3C entwickelt (vgl. [W3C05]). RIF soll den Austausch von Regeln zwischen verschiedenen Regelsprachen ermöglichen. Die Integration mit Ontologie-Modellierungssprachen ist auch hier ein wichtiger Aspekt.

Ontologien bei der Audi AG

Um den Einsatz von Ontologien zu demonstrieren, stellen wir im Folgenden eine Anwendung vor, die in der Technischen Entwicklung von Audi eingesetzt wird. In dem Projekt werden Ontologien verwendet, um verschiedenste Informationen aus dem Bereich der Fahrzeugentwicklung in einer einheitlichen, maschinenverarbeitbaren Form zu beschreiben und miteinander zu verknüpfen. Konkret soll Prozesswissen aus den folgenden Bereichen gebündelt werden:

- Computer Aided Design (CAD)
- Computer Aided Engineering (CAE)
- Computer Aided Testing (CAT)

Als Sammelbegriff für CAD, CAE und CAT verwenden wir im Folgenden den Begriff *CAx*. In diesen Bereichen werden verschiedene CA-Methoden eingesetzt, um Fahrzeuge zu konstruieren, die neu entwickelten Fahrzeuge und Teile abzusichern und anschließend für den Serieneinsatz freizugeben. Diese Erprobungs- und Absicherungsmethoden beziehen sich auf konkrete Funktionen des entwickelten Fahrzeugs, zum Beispiel hinsichtlich der Fahrzeugsicherheit oder des Komforts. Die CA-Methoden werden wiederum durch eine Vielzahl spezialisierter IT-Systeme unterstützt. Somit spannt sich ein komplexes Beziehungsnetzwerk auf, das wir als die CAx-Architektur der Technischen Entwicklung von Audi bezeichnen. Um die Gesamtkomplexität zu reduzieren, haben wir die verschiedenen Arten der in der CAx-Architektur repräsentierten Informationen in Schichten zusammengefasst. Abbildung 3 zeigt dieses Modell.

Die CAx-Architektur

Beginnend mit den Anforderungen, bietet das Schichtenmodell eine geschäftsorientierte Sichtweise auf die Fahrzeugentwicklung, die bis hin zu den eingesetzten IT-Systemen und den verarbeiteten Daten reicht:

- Die oberste Schicht definiert die Anforderungen an ein Fahrzeug.
- Die darunter liegende Schicht beschreibt ein Fahrzeug anhand seiner Funktionen und Eigenschaften. In der Technischen Entwicklung erfolgen Absicherung und Freigabe von neu entwickelten Fahrzeugen und ihren Teilen auf dieser Ebene. Abhängigkeiten zwischen Funktionen und Eigenschaften sollen erkannt und analysiert werden.
- Die im Laufe des Entwicklungsprozesses angewendeten Absicherungsmethoden werden in der Schicht „CA-Methoden" aus Entwicklersicht beschrieben. Ziel ist es, die Entwickler bei der Auswahl einer geeigneten Methode bzw. der Methodenkomposition zu unterstützen.
- Die Schicht „Prozesse" beschreibt den Fahrzeugentwicklungsprozess und setzt Wissen aus den anderen Schichten in diesen Prozesskontext.
- Die Schicht „Services und Anwendungen" folgt der Modellierung einer *service-orientierten Architektur (SOA)*. Sie enthält die vorhandenen CAx-Systeme und beschreibt diese anhand der Services, die sie anbieten.
- Als Geschäftsobjekte werden die von den CAx-Systemen genutzten Daten implementierungsunabhängig repräsentiert. Die Schicht „Geschäftsobjekte" ermöglicht somit eine einheitliche Sicht auf vorhandene Datenquellen.
- Die Schicht „Daten" beschreibt die in den CAx-Systemen real vorhandenen Daten.

Jede Schicht der CAx-Architektur wird als eine Ontologie modelliert. Abhängigkeiten werden nur zwischen Klassen und Individuen der gleichen Schicht bzw. aufeinander folgender Schichten modelliert. So sind beispielsweise CA-Methoden mit CA-Services verknüpft – und diese wiederum mit den sie implementierenden Anwendungen. Aufgrund dieser modellierten Beziehungen kann die Information, welches System eine CA-Methode letztendlich

NUR für Abonnenten: Haben Sie schon Ihren Log-In für den kostenlosen Download aller Artikel-Pdfs? Kontakt: Susanne.Herl@sigs-datacom.de!

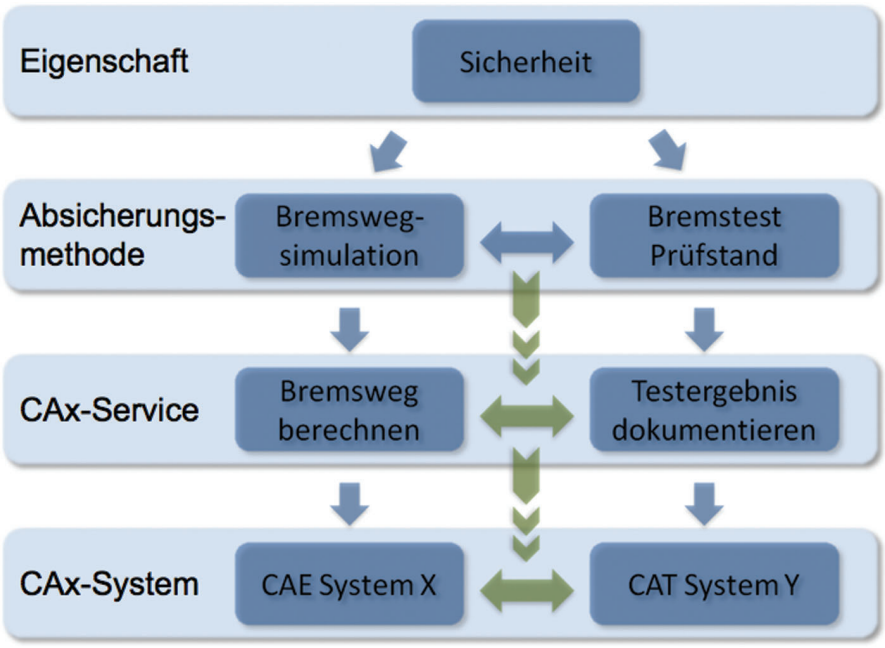

Abb. 4: Beispiel aus der CAx-Architektur.

anbietet, direkt von der Inferenzmaschine abgeleitet werden. Durch ihre Modellierung mittels Ontologien wird somit nicht nur ein zentraler Einstiegspunkt in die Informationen bereitgestellt, sondern auch die maschinengestützte Analyse von Abhängigkeiten ermöglicht. Das Beispiel in Abbildung 4 illustriert dieses Vorgehen.

Das Bild zeigt einen exemplarischen Ausschnitt der CAx-Schichtenarchitektur. Die Absicherung der Eigenschaft „Sicherheit" beinhaltet unter anderem den Test des Bremsverhaltens des Fahrzeugs. Als Absicherungsmethoden sind die Simulation des Bremswegs durch eine Software sowie das physikalische Testen auf einem Prüfstand modelliert. Zwei CAx-Systeme *X* und *Y* bieten jeweils die hierbei verwendeten Services an: Bremsweg berechnen und Testergebnis dokumentieren.

Interessiert sich ein Benutzer beispielsweise dafür, welche Absicherungsmethoden zur Freigabe der Eigenschaft zur Verfügung stehen, so zeigt die CAx-Architektur Alternativen auf. In dem Beispiel sind dies für die Eigenschaft „Sicherheit" die beiden Methoden „Bremswegsimulation" und „Bremstest Prüfstand". In Prozessinformationen kann zudem noch hinterlegt werden, wann im Entwicklungsprozess welche Methode am besten geeignet ist. Ausgehend von der Eigenschaft kann der Benutzer durch die Schichtenarchitektur bis hin zu den implementierenden CAx-Systemen navigieren und erhält so die Information, welche Systeme die gewählte Methode anbieten.

Eine andere Benutzerin hingegen möchte wissen, welche Abhängigkeiten zwischen verschiedenen CAx-Systemen bestehen, und nutzt dafür die CAx-Architektur. Basierend auf der Beziehung zwischen den Absicherungsmethoden „Bremswegsimulation" und „Bremstest Prüfstand" kann eine Abhängigkeit zwischen den beiden implementieren Systemen „CAE-System X" und „CAT-System Y" abgeleitet werden. Beispielsweise kann es erforderlich sein, Daten zwischen diesen Systemen auszutauschen und diese zu vergleichen.

Wie das Beispiel zeigt, können je nach Informationsbedarf gezielte Anfragen an die Ontologie gestellt werden. Ergebnisse der Anfrage beinhalten neben dem explizit modellierten Wissen auch die von einer Inferenzmaschine abgeleiteten Abhängigkeiten. Beziehungen, wie zwischen zwei Systemen, können somit auch noch in einem komplexen Abhängigkeitsgeflecht transparent gemacht werden. In Zukunft können die Ontologien der CAx-Architektur auch die Basis für eine semantische Integration der Daten bilden.

Realisierung der CAx-Architektur

Für den Zugriff auf die CAx-Architektur wurde eine web-basierte Anwendung entwickelt. Die Benutzungsoberfläche verwendet das Open-Source-Framework „Google Web Toolkit" (vgl. [Goo]). Die Ontologien in den verschiedenen Schichten sind in F-Logic repräsentiert. Der kommerzielle „OntoBroker" der Firma ontoprise GmbH wird als Inferenzmaschine eingesetzt (vgl. [ont]).

Erfahrungen mit der Nutzung der Architektur

Das seit Beginn 2009 eingeführte System wird derzeit von ca. 25 Mitarbeitern aus den Bereichen Anforderungsmanagement, IT- und CAx-Architektur, Support sowie von Software-Projektleitern genutzt, um die vielschichtigen Abhängigkeiten zwischen fachlichen Services und deren IT-Implementierungen aufzudecken und so die Auswirkungen geplanter Projektvorhaben besser abschätzen zu können. Mit den derzeit mehreren hundert abgebildeten Systemen ist bezüglich der Performance bisher keine technische Grenze erkennbar.

Herausforderungen

Für den Erfolg der CAx-Architektur sind unserer Meinung nach zwei Aspekte entscheidend:

- der intuitive Zugriff auf die Informationen der CAx-Architektur
- deren Aktualität

Die web-basierte Anwendung zur CAx-Architektur bietet daher eine intuitiv bedienbare Benutzungsoberfläche, die kein Vorwissen über Ontologien erfordert. Mittels verschiedener Visualisierungen können die Benutzer die Ontologie entlang den definierten Beziehungen browsen und Abhängigkeiten analysieren. Abbildung 5 zeigt eine der Visualisierungen des oben beschriebenen Beispiels. Zudem gibt es die Möglichkeit, Reports in Echtzeit zu generieren und sich Informationen zusammenfassen zu lassen. Für häufig benötigte Informationen stehen Standardanfragen bereit, beispielsweise für die Frage nach dem Ansprechpartner eines bestimmten Systems.

Der zweite wichtige Aspekt betrifft die Aktualität der in der CAx-Architektur enthaltenen Informationen. Die Aktualität von Ontologien ist vor allem bei sich schnell entwickelnden Anwendungsdomänen ein Problem (vgl. [Hep07]). Die aktuell vorhandene Ontologie hinkt häufig der Realität hinterher; während die nächste

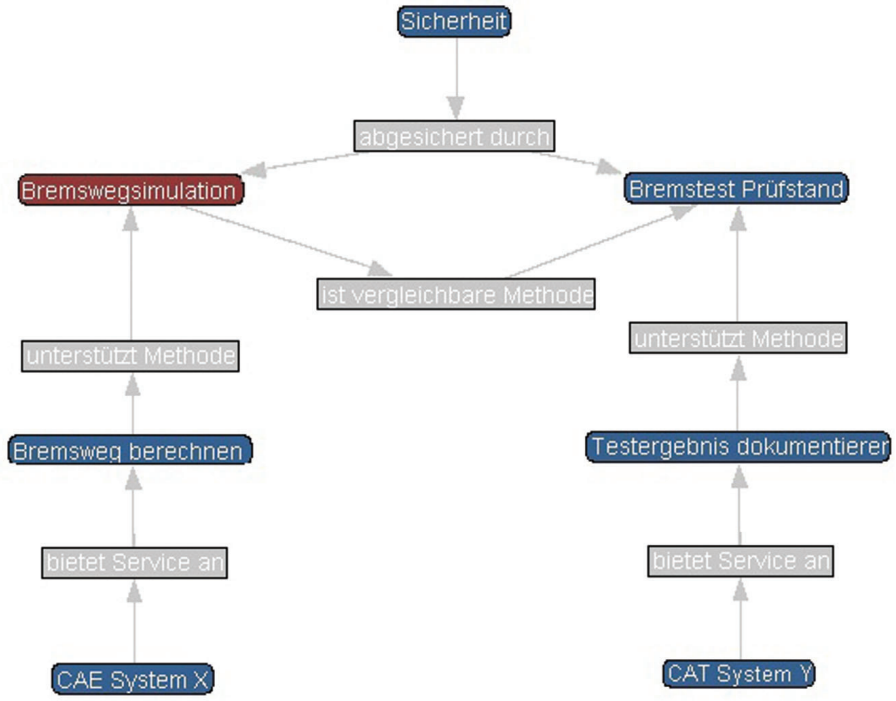

Abb. 5: Visualisierung in der CAx-Architektur.

Version erstellt wird, hat sich die Domäne bereits wieder gewandelt. Um dieses Problem anzugehen, haben wir einen innovativen Ansatz entwickelt, der mittels Web-2.0-Prinzipien Geschäftsexperten in die Entwicklung der CAx-Architektur einbindet, ohne Kenntnisse im Bereich der ontologischen Modellierung vorauszusetzen. Als Konsequenz kombiniert die Benutzungsoberfläche die Nutzung und Weiterentwicklung der CAx-Architektur.

So ähnlich wie bekannte Web-2.0-Anwendungen – z. B. Wiki- und Tagging-Systeme – benötigt man keine detaillierte Einarbeitung. Anwender können die gefundenen Informationen gleich kommentieren, korrigieren und auch bewerten. Die Änderungen bzw. Ergänzungen sind sofort sichtbar. Nutzer der CAx-Architektur können sich somit direkt dort einbringen, wo ihre Kernkompetenzen liegen, und zum schrittweisen Wachstum der Ontologie beitragen. Wenn ein Benutzer beispielsweise ein neues Individuum einträgt, kann er, wie aus Wikis bekannt, mit einer ersten kurzen Erläuterung beginnen. Andere Benutzer können die Beschreibung ergänzen. Diskussionsseiten, eine Änderungshistorie und Funktionen zur Qualitätsbewertung unterstützen die Benutzer bei ihrer kollaborativen Arbeit.

Die Web-Anwendung zur CAx-Architektur wird derzeit in ausgewählten Abteilungen der Technischen Entwicklung eingesetzt Die Benutzer nahmen an einer Schulung teil. Ein Redaktionsteam unterstützt die Anwender bei der Nutzung und Pflege der Ontologie. Es kümmert sich beispielsweise um den automatisierten Import größerer Mengen an Informationen in die Ontologie, z. B. aus Office-Dateien. Die Rückmeldungen der Geschäftsexperten fließen kontinuierlich in die Weiterentwicklung der Web-Anwendung ein. So wurden mit der zweiten Version beispielsweise Wizards eingeführt, die eine Reihe von häufigen Bearbeitungsschritten zusammenfassen. Gespräche mit Geschäftsexperten gaben auch wichtige Hinweise auf den Ausbau der Ontologie.

Erfolgsfaktoren für Ontologien

Ontologien sind mittlerweile kein reines Forschungsthema mehr, sondern bieten sich für den praktischen Einsatz in Unternehmen an. Basierend auf unseren Erfahrungen möchten wir auf mehrere Faktoren hinweisen, die wir als kritisch für den Erfolg ontologiebasierter Anwendungen halten.

Zu Beginn des Projekts steht die Auswahl der Ontologie-Entwicklungsteams. Ontologien werden typischerweise gemeinsam von Ontologie-Modellierern und Experten der Anwendungsdomäne entwickelt. Erstere sollten sich mit verfügbaren Sprachen, Standards und Tools sehr gut auskennen, um die für die zu realisierende Anwendung am besten geeigneten auswählen zu können. Eine gewisse Kenntnis der Anwendungsdomäne ist Grundvoraussetzung, um mit den Domänenexperten effizient zusammenzuarbeiten. Diese kennen die im Anwendungsbereich relevanten Informationsquellen und mögliche Interviewpartner. Wichtig ist auch ihre Fähigkeit, miteinander zu kommunizieren und so das Wissen aus dem Anwendungsbereich in eine Ontologie zu übertragen und somit zu formalisieren.

Häufig bietet sich ein iteratives Vorgehen an, wobei schon erste Versionen der Ontologie den zukünftigen Nutzern zur Verfügung gestellt werden und deren Rückmeldungen wieder in die Modellierung einfließen. Dabei kann die Verständlichkeit der Ontologie direkt geprüft werden: Benutzer sollten die modellierten Zusammenhänge verwenden können, ohne sich detailliert in die komplexen logischen Hintergründe einzuarbeiten. In einer leicht verständlichen Ontologie ist die Bedeutung der Klassen, Eigenschaften und Individuen klar (vgl. auch [Hep07]). Sinnvolle Labels und detaillierte Kommentare zu den Elementen der Ontologie sollten selbstverständlich sein. Des Weiteren beeinflusst die Komplexität der Modellierung die Lesbarkeit der Ontologie. Wir empfehlen, die einfachste Sprache zu nehmen, mit der alles Erforderliche ausgedrückt werden kann. Häufig ist eine Ausdrucksmächtigkeit, wie sie in den gezeigten Beispielen verwendet wurde, ausreichend.

Für einen dauerhaften Erfolg ist es wichtig, dass die Anwender aus der Ontologie direkten Nutzen ziehen und in ihrer täglichen Arbeit unterstützt werden. Visualisierungen der Ontologie sowie Möglichkeiten, diese zu browsen, spielen eine wichtige Rolle. Ein weiterer kritischer Punkt ist die Aktualität der Ontologie. Da sich die modellierte Domäne häufig rasch verändert, sollte man sich bereits bei Projektstart darüber Gedanken machen, wie die Ontologie kontinuierlich weiterentwickelt werden kann. Ein Ansatz besteht darin, die Benutzer über Web-2.0-Ansätze in die Weiterentwicklung der Ontologie einzubinden.

NUR für Abonnenten: Haben Sie schon Ihren Log-In für den kostenlosen Download aller Artikel-Pdfs? Kontakt: Susanne.Herl@sigs-datacom.de!

Fazit

Der Beitrag stellt eine konkrete Anwendung von Ontologien in einem Projekt in der Technischen Entwicklung von Audi vor. Ontologien sind ein innovativer Ansatz zur Formalisierung und effizienten Nutzung von Wissen. Der vorgestellte Ansatz eignet sich insbesondere für die Repräsentation und Exploration von Vernetzungswissen. Es ist gut denkbar, diesen Ansatz auch auf andere als den hier vorgestellten Anwendungsbereich zu übertragen. Man sollte jedoch nicht vergessen, dass Ontologien eine relativ junge Technologie sind. Trotz großer Fortschritte müssen sie erst noch die Reife, Standardisierung, Performance und nicht zuletzt auch Akzeptanz erreichen, die in anderen Bereichen, zum Beispiel der relationalen Datenbanken, bereits erreicht sind. Auch wenn sich zunehmend akzeptierte Standards herausbilden – wie OWL für die Repräsentation von Ontologien – gibt es noch einen großen Bedarf bei der Integration von Ontologien und Regeln.

Das Angebot an Werkzeugen zur Modellierung, Verwaltung und Verwendung von Ontologien wächst stetig. Einige sind mittlerweile sehr robust und werden im industriellen Umfeld eingesetzt. Die Performance der Werkzeuge – insbesondere der Inferenzmaschinen – hat sich in den letzten Jahren stark verbessert. Inferenz ist mittlerweile auch mit großen Ontologien mit vielen Individuen möglich. Die Performance hängt jedoch wesentlich von der verwendeten Ontologiesprache ab. Mit zunehmender Ausdrucksmächtigkeit der Sprache geht immer auch eine signifikante Erhöhung der Komplexität und damit eine Verschlechterung der Performance einher. Unsere Erfahrungen haben jedoch gezeigt, dass auch eine Sprache mit einer geringeren Ausdrucksmächtigkeit ausreichen kann und zudem die Verständlichkeit für den Anwender erhöht.

Auch wenn es derzeit noch einige offene Punkte gibt, die vor einem breiteren Einsatz von Ontologien gelöst werden müssen, sind die bisher erzielten Ergebnisse sehr vielversprechend. Ontologien erlauben auf einfache Art und Weise die schrittweise Aufnahme, Erweiterung bzw. Modifikation von beliebigen Einzelinformationen mit der Möglichkeit, diese semantisch zu einem Ganzen zu verknüpfen und maschinell auszuwerten. Nutzer der Ontologie können das derart modellierte Wissen, insbesondere die Abhängigkeiten, explorieren und analysieren. ■

Literatur & Links

[Baa03] F. Baader, The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, 2003

[Ber98] T. Berners-Lee, The Semantic Web Roadmap, 1998, siehe: www.w3.org/DesignIssues/Semantic.html

[Gru08] T. Gruber, Ontology, in: L. Liu, M. Tamer Özsu (Hrsg.), The Encyclopedia of Database Systems, Springer-Verlag, 2008, siehe: http://tomgruber.org/writing/ontology-definition-2007.htm

[Goo] Google Web Toolkit, siehe: http://code.google.com/intl/de-DE/webtoolkit/

[Hep07] M. Hepp, Possible Ontologies – How Reality Constrains the Development of Relevant Ontologies, in: IEEE Internet Computing. S. 90-96, 2007

[Kif05] M. Kifer, Rules and Ontologies in F-Logic, in: Reasoning Web, Springer Lecture Notes in Computer Science. Vol. 3564, S. 22-34, 2005

[ont] Web-Seite der Firma ontoprise GmbH, (W3C), siehe: www.ontoprise.de/

[W3C-a] World Wide Web Consortium, Resource Description Framework (RDF), siehe: www.w3.org/RDF/

[W3C-b] World Wide Web Consortium, Semantic Web Rule Language, siehe: www.w3.org/Submission/SWRL/

[W3C01] World Wide Web Consortium, Semantic Web Activity, 2001, siehe: www.w3.org/2001/sw/

[W3C04] World Wide Web Consortium, Ontology Language, 2004, siehe: www.w3.org/2004/OWL/

[W3C05] World Wide Web Consortium, Rule Interchange Format (RIF), 2005, siehe: www.w3.org/2005/rules/wiki/RIF_Working _Group

[W3C07] World Wide Web Consortium, Schichtenmodell (Layercake) des Semantic Web, siehe: www.w3.org/ 2007/03/layerCake.png