Vom Knowledge Engineering zum Knowledge Management: Entwurf, Aufbau und Wartung von Wissensbanken

Ulrich Reimer

Rentenanstalt / Swiss Life
Informatik-Forschungsgruppe
Postfach
CH–8022 Zürich
ulrich.reimer@swisslife.ch

Knowledge Engineering (KE) umfaßt alle Tätigkeiten, die die Erhebung und Modellierung von Wissen und dessen Implementation in einer Wissensbasis zum Gegenstand haben. Eine solche Wissensbasis bildet den Kern eines wissensbasierten Systems. Darunter versteht man ein System, welches domänenspezifisches Wissen besitzt, das für seine Funtkionalität essentiell ist. Dieses Wissen ist explizit durch Symbolstrukturen in einer dedizierten Komponente des Systems – der Wissensbasis – repräsentiert. Ein System mit gleicher Funktionalität kann selbstverständlich auch nicht-wissensbasiert realisiert sein, indem das nötige Wissen im Programmcode codiert ist. Der Vorteil, ein System wissensbasiert zu implementieren, besteht darin, daß durch die Verwendung von Wissensrepräsentationssprachen, die im Vergleich zu allgemeinen Programmiersprachen eine eingeschränkte Ausdrucksfähigkeit aufweisen, Systeme mit komplexer Funktionalität schneller realisierbar werden. Ferner wird durch die deklarative Repräsentation des Wissens dieses inspizierbar. Dadurch

- kann das Wissen leichter erweitert werden,
- wird es eher wiederverwendbar,
- wird es verifizierbar, da automatische Konsistenzprüfungen möglich werden,
- können Erklärungen generiert werden, warum das System in einer konkreten Situation ein bestimmtes Verhalten zeigt.

Man unterscheidet zwei prinzipiell unterschiedliche Arten des Vorgehens beim KE. Beim *Rapid Prototyping* wird das für die Anwendung relevante Wissen erhoben und unmittelbar mit den zur Verfügung stehenden Repräsentationsformalismen beschrieben. Wissensbeschreibung und -implementation fallen zusammen. Für in Betrieb zu setzende Systeme wählt man dagegen ein *modellbasiertes Vorgehen*, wo zunächst ein konzeptuelles Modell des zu erfassenden Expertenwissens erstellt wird. Dieses Modell ist so beschrieben, daß es möglichst nahe der Denkweise des Fachexperten ist. Erst der anschließende Schritt der Implementation dieses Modells beschreibt das Wissen in einer formal interpretierbaren Notation. Die Erhebung des konzeptuellen Modells beginnt zunächst mit dem prinzipiellen Problemlösungsvorgehen auf noch recht abstrakter Ebene und verfeinert dies zunehmend. Oft gibt es schon ein generisches Modell, das man als Ausgangspunkt heranziehen kann, denn für einen gegebenen Typ von Problemstellung (wie Diagnose, Entwurf, etc.) treten nur bestimmte Arten des Vorgehens bei der Problemlösung auf.

Die wohl bekannteste KE-Methodik, die auf einem modellbasierten Ansatz beruht, ist KADS [Schreiber et al. 93]. KADS unterscheidet mehrere Typen von Modellen. Die wichtigsten sind das Aufgabenmodell (*task model*), das die Funktionalität des anvisierten Systems in verschiedene Aufgaben, die für das System anfallen, untergliedert. So hat beispielsweise ein System, das einen Experten bei der Reparatur von technischen Geräten unterstützt, zunächst die Aufgabe der Fehlerdiagnose zu lösen und anschließend die Aufgabe der geeigneten Fehlerbehebung, die durch verschiedene Randbedingungen, die einzuhalten sind, u.U. recht komplex sein kann. Das Kooperationsmodell (*model of cooperation*) beschreibt, wie die Aufgaben, die auf unterschiedliche Agenten (unter ihnen der Benutzer) aufgeteilt sein können, miteinander zusammenspielen. Das Expertisenmodell (*model of expertise*) bildet den Kern und modelliert die Expertise, die für die Lösung der einzelnen Aufgaben, die im Aufgabenmodell festgelegt sind, nötig ist. Dazu werden mehrere Arten von Wissen unterschieden, u.a.:

- Wissen über die Begriffe in der zugrundeliegenden Domäne, ihre Eigenschaften und Beziehungen untereinander,
- Wissen über die verschiedenen Arten von Inferenzen, die auf dem Domänenwissen möglich sind,
- Wissen darüber, wie die verschiedenen Inferenztypen zu kombinieren sind, um eine bestimmte Aufgabe zu lösen (Kontrollwissen).

Während die Erstellung des Expertisenmodells von KADS weitgehend der klassischen Aufgabe des KE entspricht, tangieren die weiteren Modelltypen von KADS zunehmend Aspekte, wie sie auch sonst im Software Engineering (SE) auftreten (siehe auch [Schreiber/Wielinga 93]). Dies spiegelt die Entwicklung, die das Gebiet des KE genommen hat, wider. Es hat sich mehr und mehr in das Gebiet des SE integriert, indem alle Systemaspekte betrachtet werden und nicht nur der Aufbau der Wissensbasis – der jedoch nach wie vor ein spezieller Gegenstandsbereich bleibt.

Ursprünglich lediglich für die Beschreibung von Problemlösungsmodellen gedacht, hat die Operationalisierung eines mit KADS oder einem vergleichbaren System erstellten Modells durch dessen Interpretation zunehmend an Bedeutung gewonnen [Schreiber 93, Voß/Karbach 93, Angele et al. 96].

Die Erstellung eines konzeptuellen Modells wird sehr erleichtert, wenn man auf eine Bibliothek generischer Modelle zurückgreifen kann. Ein wesentlicher Teil der Forschungstätigkeit ist vor allem in den 80er Jahren in die Identifizierung und Beschreibung geeigneter generischer Modelle geflossen [Puppe 90]. Ein Beispiel ist die heuristische Klassifikation, die bei Diagnoseproblemen Anwendung findet. Es handelt sich dabei um eine Verfeinerung der einfachen assoziativen Klassifikation. Statt aus vorliegenden Symptomen direkt auf eine Diagnose zu schließen, werden Symptome zunächst generalisiert und auf abstraktere Symptome abgebildet. Erst dann erfolgt die Anwendung einer assoziativen Klassifikationsregel (der heuristische Abgleich), die den Symptomen eine Diagnose zuordnet. Anschließend wird die u.U. noch zu allgemeine Diagnose konkretisiert.

Neben dem Aufbau von Wissensbasen ist auch deren Wartung und Evolution über die gesamte Lebensdauer hinweg eine wichtige Aufgabe, für die geeignete Werkzeuge benötigt werden (siehe Kap.7 in [Morik et al. 93]).

Das Gebiet des KE war bis vor einigen Jahren weitgehend mit Aspekten der Entwicklung und Wartung wissensbasierter Systeme befaßt. Mit der zunehmenden Erkenntnis, daß Wissen für Organisationen einer der wichtigsten Produktionsfaktoren ist – wenn nicht (in den industrialisierten Ländern) der wichtigste überhaupt – hat sich in den letzten Jahren die Perspektive auf die Unterstützung des Managements der Ressource Wissen im Unternehmen erweitert – nicht nur im Gebiet des

KE, sondern auch in benachbarten Gebieten innerhalb und außerhalb der Informatik. Man spricht von Knowledge Management (KM). Methoden und Verfahren, die im Gebiet des KE entwickelt wurden, haben eine hohe Relevanz für das KM, da es im Kern nach wie vor um die Erhebung, Beschreibung und Wartung von Wissen geht.

Im Zentrum des KM steht das sogenannte Unternehmensgedächtnis (UG), wo das für das Unternehmen relevante Wissen abgelegt ist [Abecker et al. 97]. Auf dem UG können wissensbasierte Systeme aufsetzen, wenn das betreffende Wissen genügend formal repräsentiert ist, aber das ist nur einer von vielen Aspekten. In das UG gehören neben anderem zum Beispiel eine Beschreibung des Organisationsaufbaus, das unternehmensweite Datenmodell, die Beschreibung der Geschäftsprozesse sowie alle Weisungen und Vorschriften. Neuartige Aufgaben, die sich im Zusammenhang mit dem UG ergeben, erfordern völlig neue Lösungsansätze.

- Die *Wiederverwendung von Wissen* für andere Aufgaben als die, wofür es ursprünglich modelliert wurde, muß möglich sein, da die erneute Modellierung viel zu teuer wäre und die entstehenden Redundanzen kaum beherrschbar. Ansätze hierfür betreffen standardisierte, implementationsunabhängige Austauschformate für Wissen [Genesereth/Ketchpel 94, Genesereth 95] sowie die Verwendung von Ontologien (s.u.), um die in die Repräsentationen eingeflossenen Sichtweisen explizit zu machen (eine ähnliche Motivation wie für die Entwicklung von KADS). Vokabular, das man zur Beschreibung eines Weltausschnitts verwendet.

- Der Aufbau eines UG erfordert die *Integration von Wissen*, das aus unterschiedlichsten Wissensquellen stammt. Der Grad an Formalisierung des Wissens kann variieren – von kaum formalisiert, wie Hypertexte, die den Ablauf von Geschäftsprozessen dokumentieren, über mittelmäßig formalisiert, wie die Beschreibung von Geschäftsprozessen mit einem Werkzeug zur Geschäftsprozeßmodellierung, bis hin zur formalen Repräsentation von Geschäftsprozessen und den Gesetzen und Weisungen, die sie steuern. Die Formalisierung der Ontologien, die den einzelnen Wissensquellen zugrunde liegen, erlauben bis zu einem gewissen Grad die automatische Integration von Wissen, da die den unterschiedlichen Repräsentationen innewohnenden Annahmen explizit gemacht werden (z.B. welcher Zeitbegriff verwendet wird, ob zwischen definierenden und charakterisierenden Eigenschaften unterschieden wird, welche Arten von Begriffen es gibt – wie zählbar, nicht zählbar, räumlich, zeitlich – vgl. [Gruber 92, Guarino et al. 94]).

- Ein UG ist von geringem Wert, wenn das darin enthaltene Wissen nicht an den Orten, wo es jeweils gerade benötigt wird, bereit steht. Idealerweise erfolgt die Bereitstellung automatisch, da ein Mitarbeiter sich u.U. gar nicht bewußt ist, daß Wissen, welches er benötigt, im UG vorhanden ist (möglicherweise weiß er noch nicht einmal, welches Wissen er benötigt). Ein sehr interessantes Paradigma ist in diesem Zusammenhang das des *Just-in-time Knowledge Delivery* [Cole et al. 97]. sorgen, daß ein Mitarbeiter neues Wissen (typischerweise Dokumente) automatisch mitgeteilt bekommt.

- Es werden Mechanismen für die dezentrale *Erfassung von Wissen* benötigt, so daß dort, wo es erarbeitet wird, auch gleich die Erfassung erfolgt. Hier sind zum Beispiel Systeme, die Gruppenarbeit unterstützen, interessant, da durch die fortlaufende Dokumentation der Gruppenarbeit wichtige Erkenntnisse und Erfahrungen festgehalten sind. Relevant für die Wissenserfassung sind ferner auch Ansätze des Maschinellen Lernens und des automatischen Textverstehens.

- Ein UG wird nicht nur durch diverse Software-Systeme angefragt, sondern auch direkt durch Benutzer. Ein einheitliches, zugrundeliegendes Begriffssystem (das allein schon für die Integration von Wissen verschiedener Art und aus unterschiedlichen Quellen nötig ist) erleicht die *Anfrage an ein UG* ungemein. Ein solches Begriffssystem ist in der Regel aber recht groß, so daß die Gefahr besteht, daß der Benutzer nicht weiß, in welchen Begriffen er eine Anfrage formulieren muß.

Die Aufgaben, die im Rahmen eines KM anfallen, sind in verwandter Weise auch in benachbarten Gebieten anzutreffen. Die Wiederverwendung von Wissen hat Parallelen zur Wiederverwendung von Software-Komponenten. Die Integration von Wissen ist seit den 70er Jahren im Datenbankbereich als Schema- und Datenintegration immer wieder ein Thema, zuletzt vermehrt im Kontext von Data Warehouses. Das Zusammenfügen von Wissen unterschiedlichen Grades an Formalisierung ist etwas, das im Requirements Engineering wiederzufinden ist als das Integrieren von Anforderungen, die sowohl in der Form natürlichsprachlicher Äußerungen als auch als formale Anforderungsspezifikationen sowie in verschiedenen halbformalen Formaten vorliegen können. Die Handhabung einer umfangreichen Terminologie beim Anfragen eines UG hat gewisse Ähnlichkeiten mit dem Umgang großer Klassenbibliotheken objektorientierter Systeme und tritt natürlich auch im Zusammenhang mit unternehmensweiten Datenmodellen auf. Dies sind nur einige Parallelen, die unmittelbar ins Auge fallen, weitere lassen sich identifizieren: Möglichkeiten für Synergien gibt es viele.

Danksagung
Ich danke meinen Kollegen Jörg-Uwe Kietz und Martin Staudt für die Diskussionen, die zur Klärung und Pointierung verschiedener Aspekte beigetragen haben.

Literatur

[Abecker et al. 97] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kühn, M. Sintek: Towards a Well-Founded Technology for Organizational Memories. In: B.R. Gaines, R. Uthurusamy (eds): Artificial Intelligence in Knowledge Management. Papers from the 1997 AAAI Spring Symposium. Menlo Park: AAAI Press, 1997.
http://ksi.cpsc.ucalgary.ca:80/AIKM97/AIKM97Proc.html

[Angele et al. 96] J. Angele, S. Decker, R. Perkuhn, R. Studer: Modeling Problem-Solving Methods in New KARL. In: Proc. 10th Banff Workshop on Knowledge Acquisition for Knowledge-Based Systems, 1996.
http://ksi.cpsc.ucalgary.ca/KAW/KAW96/KAW96Proc.html

[Cole et al. 97] K. Cole, O. Fischer, P. Saltzman: Just-in-Time Knowledge Delivery. In: Communications of the ACM, Vol.40, No.7, 1997, pp.49-53.

[Genesereth 95] M.R. Genesereth: Knowledge Interchange Format Specification. http://logic.stanford.edu/kif/specification.html

[Genesereth/Ketchpel 94] M.R. Genesereth, S.P. Ketchpel: Software Agents. In: Communications of the ACM, Vol.37, No.7, 1994, pp.48-53.

[Guarino et al. 94] N. Guarino, M. Carrara, P. Giaretta: Formalizing Ontological Commitments. In: Proc. 12th National Conf. on Artificial Intelligence, 1994, pp.560-567.

[Gruber 92] T.R. Gruber: A Translation Approach to Portable Ontology Specifications. In: Knowledge Acquisition, Vol.5, 1992, pp.199–220.

[Morik et al. 93] K. Morik, S. Wrobel, J.-U. Kietz, W. Emde: Knowledge Acquisition and Machine Learning: Theory, Methods, and Applications. London: Academic Press, 1993.

[Puppe 90] F. Puppe: Probemlösungsmethoden in Expertensystemen. Berlin: Springer-Verlag, 1990.

[Schreiber 93] G. Schreiber: Operationalizing Models of Expertise. In: [Schreiber et al. 93].

[Schreiber/Wielinga 93] G. Schreiber, B. Wielinga: KADS and Conventional Software Engineering. In: [Schreiber et al. 93].

[Schreiber et al. 93] G. Schreiber, B. Wielinga, J. Breuker (eds): KADS: A Principled Approach to Knowledge-Based System Development. London: Academic Press, 1993.

[Voß/Karbach 93] A. Voß, W. Karbach: MODEL-K: Making KADS Run. In: [Schreiber et al. 93].