Requirements Engineering in MIKE

Rainer Perkuhn
Institut für Angewandte Informatik und Formale Beschreibungsverfahren
Universität Karlsruhe, 76 128 Karlsruhe
e-mail: perkuhn@aifb.uni-karlsruhe.de

Abstract

Sowohl bei der Entwicklung konventioneller Software als auch bei der Entwicklung wissensbasierter Systeme fehlen z.Z. systematische Ansätze, Anforderungen an das zu entwickelnde Produkt „ingenieurmäßig“ zu erheben. Die Probleme, mit denen sich der Software Engineer konfrontiert sieht, ähneln denen der Wissensakquisition im Knowledge Engineering. Der an der Universität Karlsruhe am Institut AIFB entwickelte Ansatz MIKE ([AFL93]) beschreibt eine systematische Vorgehensweise zur Entwicklung wissensbasierter Systeme. Die Beschreibung der spezifischen Anforderungen an wissensbasierte Systeme ist Gegenstand der aktuellen Forschung; mit MIKE steht aber bereits das Gerüst zur Verfügung, mit denen die Anforderungen im Laufe der weiteren Entwicklungsphasen verwaltet werden können.

MIKE

MIKE ist ein methodischer Ansatz zur systematischen Entwicklung wissensbasierter Systeme und steht als Akronym für „modellbasiertes inkrementelles Knowledge Engineering“. Zentrale Aspekte sind der Entwurf von expliziten Modellen, die das Problemlösungs- und das Domänenwissen beschreiben, sowie das wiederholte Durchlaufen der einzelnen Schritte der Entwicklungsphasen, um sich inkrementell dem gewünschten System anzunähern.

Die Entwicklung eines wissensbasierten Systems (WBS) umfaßt in MIKE folgende Schritte. Das Wissen eines Experten, d.h. eines Spezialisten des Anwendungsbereichs, wird über Interviews und evtl. Fragebögen festgehalten - oder aus Fachbüchern, Fallbeschreibungen u.ä. extrahiert. Diese Dokumente und Protokolle stellen das Erhebungsmodell dar, das durch weitere gezielte Interviews verfeinert werden kann und das für den Knowledge Engineer, d.h. den Spezialisten für die Analyse des für die Entwicklung eines WBS benötigten Wissens, als Ausgangspunkt der Wissensstrukturierung dient. Aus dem Erhebungsmodell werden relevante Konzepte und Aktivitäten herausgearbeitet und als Elemente (Knoten) des Strukturmodells beschrieben. Die Verbindung der Elemente durch Kanten vordefinierter Typen ergibt ein semiformales Modell des Domänen- und Problemlösungswissens. Vom Strukturmodell ausgehend führt die Formalisierung zu einer funktionalen Spezifikation der Problemlösungskomponente des WBS, zum Modell der Expertise (vgl. KADS [SWB93], [WSB92]), für das in MIKE die formale Beschreibungssprache KARL ([Fen95b]) zur Verfügung gestellt wird. Das Modell der Expertise ist zum einen ausführbar ([Ang93]), so daß der aktuelle Stand der Modellierung durch den Experten und evtl. den späteren Benutzer evaluiert werden kann; zum anderen ist es durch die Einbeziehung auch nicht-funktionaler Anforderungen erweiterbar zum Design-Modell ([Lan95]), das schließlich als Grundlage der Implementierung dient.

Alle Entwicklungsschritte sind möglichst strukturerhaltend in dem Sinne, daß sich die Struktur in einem Modell in den Modellen der folgenden Phasen wiederfindet.

Anforderungen an wissensbasierte Systeme

Die Anforderungen an ein WBS kann man grob unterteilen in Anforderungen an das benötigte Anwendungsbereichswissen und die benötigte Problemlösungsmethode (zum Begriff Problemlösungsmethode vgl. [Pup90]), sowie Anforderungen an die Benutzungsschnittstelle und weitere Systemeigenschaften, wie z.B. das Laufzeitverhalten, - in Abhängigkeit davon, wer über das Wissen über diese Anforderungen verfügt (vgl. [GuT94], S. 99ff). Aus KE-Sicht sind primär die ersteren von Interesse, die gemeinhin von einem Experten zu erfragen sind. Die letzteren, die ein späterer Benutzer in einer partizipativen Entwicklung durch z.B. exploratives Prototyping evaluieren kann, haben ihr Korrelat auch bei der Entwicklung konventioneller Software (s. [Dav93]), sollen aber an dieser Stelle nicht weiter erörtert werden, ebensowenig wie Anforderungen an die Modellierung des Anwendungsbereichswissens.

Anforderungen an Problemlösungsmethoden

Eine systematische (formale) Beschreibung der Anforderungen an Problemlösungsmethoden (PLMen) wäre der beste Ansatz für Requirements Engineering i.e.S. und die Wiederverwendung von PLMen oder Teilen davon - doch gibt es erst wenige Ansätze in diesem Bereich. Eine Idee besteht darin, PLMen gemäß bestimmter Eigenschaften zu Klassen zusammenzufassen, so daß man dadurch eine Vorauswahl treffen kann ([Pup90]). Vergleichbar ist ein - allerdings in der derzeitigen Form noch experimentell zu nennender - Schritt in MIKE: die Eigenschaftsbestimmung ([Neu94]). Über einen Fragebogen ermitteln der Experte und der Knowledge Engineer gemeinsam die Eigenschaften von Anwendungsproblemen, die dann mit Eigenschaften in einer Bibliothek zur Verfügung gestellter PLMen abgeglichen werden. Die Eigenschaften umfassen die Typen Basisdaten (z.B. Ist-Zustand), Problemlösungswissen (z.B. Fallsammlung), Eingabeeigenschaften (z.B. Regeln), Ausgabedaten (z.B. Tupel) und Tätigkeiten/Vorgehensweisen (z.B. Vergleich). Aufgrund der Charakterisierung werden im Idealfall eine oder mehrere PLMen ermittelt, von denen sich die geeigneteste im Lauf der weiteren Erhebung herauskristallisieren wird. Diese dient ggf. modifiziert als Teil des Modells der Expertise und als Anleitung für die weitere Datenerhebung. Eine PLM ist eine generische Beschreibung der Problemlösung, die über abstrakten Einheiten, wie z.B. Symptom oder Komponente, formuliert ist. Neben den Lösungsschritten der PLM, die mit denen verglichen werden müssen, die der Experte bei seinem Vorgehen beschrieben hat, müssen auch die abstrakten Einheiten anhand der Protokolle auf ihr domänenspezifisches Korrelat abgebildet werden können. Ein Symptom kann in einer medizinischen Anwendung etwa der hohe Blutdruck sein, für einen Automechaniker wäre der unrunde Lauf eines Motors ein Symptom. Arzt und Mechaniker setzen aber die gleiche PLM für die Diagnose ein.

Das Modell, das der Knowledge Engineer entwickelt, muß nicht vollkommen und in allen Details mit dem Vorgehen des Experten übereinstimmen, doch jede Modellierungsentscheidung, die davon abweicht, ist besonders gut zu dokumentieren. Denn die Expertenaussagen stellen die beste Rechtfertigung für eine Modellierungsentscheidung dar.

Eine PLM an sich ist in dem inkrementellen Prozeß nur wenig von Bedeutung, wenn sie nicht auf das Expertenwissen zurückgeführt werden kann. Erst durch die Dokumentation der Modellierungsentscheidungen läßt sich eine Modifikation der PLM über eine Nacherhebung von

Expertenwissen in einen konsistenten Prozeß einbetten.
Welche Eigenschaften geeignet sind, PLMen zu beschreiben, und wie formal man diese fassen kann, wird z.Z. diskutiert ([Fen95a]). Unabhängig davon hat man durch die Fragebogentechnik eine Anleitung zur Erhebung von Kategorien von Anforderungen. Die vorgeschlagene PLM bietet ein Grundgerüst für das Modell der Expertise, die bestimmte Anforderungen erfüllt, so daß dann zu überprüfen ist, ob die Anforderungen auch die geforderten sind.

Nachvollziehbarkeit von Anforderungen

Ohne Vorgaben besteht die übliche Vorgehensweise darin, direkt aus den natürlichsprachlichen Protokollen die Anforderungen abzuleiten. Dies ist zwar verbunden mit sämtlichen Nachteilen bei der Interpretation natürlicher Sprache (vgl. [Bal82], S. 97), aber MIKE unterstützt konsequent den Aspekt der Nachvollziehbarkeit. Dies gilt entsprechend auch für die o.g. zielgerichtete Datenerhebung.
Von jedem Element des Strukturmodells wird ein Verweis angelegt zu der Stelle in den Protokollen, aus der dieses Element abgeleitet wurde. Ebenso verdeutlichen Verbindungen zwischen dem Modell der Expertise und dem Strukturmodell, an welchen Stellen die formalen KARL-Einheiten ihr semiformales Gegenstück im Strukturmodell haben ([FLN94]). Durch die sanften und dokumentierten Übergänge zwischen den einzelnen Modellen und insbesondere durch die Ausführbarkeit des Modells der Expertise können Experte und Knowledge Engineer gemeinsam überprüfen, wie sich die Anforderungen bei der PLM niederschlagen. Werden die Erwartungen des Experten verletzt, muß der Knowledge Engineer seine Modelle modifizieren und das dazu benötigte Wissen vom Experten nacherheben. Durch die Rückverfolgbarkeit bis zu den Protokollen ist sofort nachvollziehbar, welche Anforderungen zu Unrecht aus welchen Expertenaussagen abgeleitet wurden. Über diese Aussagen muß sich der Knowledge Engineer Klarheit verschaffen, indem er gezielt den Experten zu diesen Punkten befragt. Somit ist gewährleistet, daß das, was sich als Anforderung im implementierten System niederschlägt, stets zurückzuverfolgen ist bis zu den Aussagen des Experten.

„Konventionelle“ Anforderungen

Prinzipiell spielt es bei MIKE keine Rolle, ob die zu entwickelnden Komponenten als konventionelle oder als wissensbasierte Software zu realisieren sind. Im Modell der Expertise wird die Integration konventioneller Komponenten aber bisher nicht explizit berücksichtigt. Ebensowenig werden explizit ausgearbeitete Techniken zur Verwaltung großer Datenmengen zur Verfügung gestellt - außer der Möglichkeit einer hierarchischen (objektzentrierten) Modellierung.

Ausblick

Aktueller Forschungsgegenstand in MIKE ist es, Anforderungen an PLMen und Teilen davon zu formulieren. Darauf aufbauend liesse sich die Fragebogentechnik o.ä. dahingehend verfeinern, daß man zu einer i.e.S. ingenieurmäßigen Erfassung von Requirements kommen kann. Darüberhinaus ermöglicht dies den Aufbau einer Bibliothek von PLMen oder Teilen davon, in der über Anforderungsbeschreibungen nach wiederverwendbaren Komponenten recherchiert werden kann.

Literatur

[AFL93] Angele, J./Fensel, D./Landes, D./ Neubert, S./Studer, R.: Model-based and incremental knowledge engineering: the MIKE approach. in: Cuena, J. (ed.): *Knowledge-*

Oriented Software Design. IFIP-Transactions A-27, North-Holland, Amsterdam 1993, S. 139 - 168

[Ang93] Angele, Jürgen: Operationalisierung des Modells der Expertise. Infix, St. Augustin 1993

[Bal82] Balzert, Helmut: Die Entwicklung von Software-Systemen. Bibliographisches Institut, Mannheim 1982

[Dav93] Davis, Alan M.: Software Requirements. Prentice Hall, Englewood Cliffs NJ 1993

[Fen95a] Fensel, Dieter: Assumptions and Limitations of Problem-Solving Method: A Case Study. in: Proceedings of the 9th Knowledge Acquisition for Knowledge-Based Systems Workshop KAW '95 (Banff, Kanada, 26.2. - 3.3.), 1995, S. 27-1 - 27-19

[Fen95b] Fensel, Dieter: The Knowledge Acquisition and Representation Language (KARL). Kluwer, Boston 1995

[FLN94] Fensel, D./Landes, D./Neubert, S./Studer, R.: Integrating semiformal and formal methods in knowledge-based systems development. in: Proceedings of the 3rd Japanese Knowledge Acquisition Workshop JKAW '94 (Hatoyama, Japan, 7.-9. November) 1994, S. 73 - 87

[GuT94] Guida, Giovanni/Tasso, Carlo: Design and Development of Knowledge-Based Systems. Wiley&Sons, Chichester 1994

[Lan95] Landes, Dieter: Die Entwurfsphase in MIKE. Methode und Beschreibungssprache. Infix, St. Augustin 1995

[LaS95] Landes, D./Studer, R.: The Treatment of Non-Functional Requirements in MIKE. erscheint in: Proceedings of the 5th European Software Engineering Conference ESEC '95 (Barcelona, Spanien, 25.9. - 28.9.)

[Neu94] Neubert, Susanne: Modellkonstruktion in MIKE. Methoden und Werkzeuge. Infix, St. Augustin 1994

[Pup90] Puppe, Frank: Problemlösungsmethoden in Expertensystemen. Springer, Berlin 1990

[SWB93] Schreiber, G/Wielinga, B.J./ Breuker, J.A. (eds.): KADS. A Principled Approach to Knowledge-Based System Development. Academic Press, London 1993.

[WSB92] Wielinga, B.J./Schreiber, A. Th./Breuker, J.A.: KADS: A Modelling Approach to Knowledge Engineering. in: *Knowledge Acquisition*, Vol. 4, No. 1, 3/1992, S. 5 - 53