

Schwerpunkte bei der Wissensmodellierung am Beispiel einer Prozeßsimulation

Volker Dötsch*

Universität Leipzig, Institut für Informatik
Postfach 920, 04009 Leipzig

Zusammenfassung

Der zunehmende Bedarf, wissensbasierte Systeme für realistische Anwendungen einzusetzen, stellt eine Herausforderung an die Modellierung der zugrunde liegenden Domäne dar. Bei wichtigen Entscheidungen wird dem Entwerfer zur Zeit nur ungenügende oder keine Unterstützung seitens der Entwurfssysteme gegeben. Um beim Systementwurf Erfahrungen zu sammeln, wurde mit Studenten ein Praxisseminar durchgeführt. Die dort gemachten Erfahrungen bezüglich der Bedeutung verschiedener Entwurfsentscheidungen werden nachfolgend beschrieben.

1 Einleitung, Motivation

Wissensbasierte Systeme werden immer häufiger in der Praxis eingesetzt, zum Beispiel bei der Planung, Steuerung, Überwachung, usw. Dies geschieht beispielsweise oft aus ökonomischen Gründen, häufig aber auch nur, um die Sicherheit von technischen Anlagen zu erhöhen bzw. auf medizinischem Gebiet, um die Patienten risikoärmer zu versorgen.

Durch die sich ständig weiter verbreitende Nutzung wissensbasierter Systeme steht häufiger die Forderung nach dem Einsatz derartiger Systeme bei immer komplexeren praktischen Anwendungen. Dies stellt eine Herausforderung an die Modellierung der zugrunde liegenden Domäne dar.

Einführungen zu Wissensrepräsentation, Wissensverarbeitung und Expertensystemen findet man z.B. in [Pup91], [Str91], [Bib93] oder [Hen91]. Speziell auf die Problematik der Wissensrepräsentation in komplexen Domänen wird in [Hel96] eingegangen. Einige kritische Betrachtungen zu Wissensverarbeitung und Expertensystemen finden sich in [BHJ⁺94], [GS96], [WF88] und im Kapitel 21 von [Pup91].

Um neben theoretischem Wissen auch praktische Erfahrungen bei der Modellierung von Anwendungen zu erlangen, wurde mit Studenten der Informa-

tik ein Praxisseminar durchgeführt. In ihm sollten sie ihr bereits in entsprechenden Lehrveranstaltungen erworbenes theoretisches Wissen zu Expertensystemen, Wissensrepräsentation, Wissensverarbeitung und Planung vertiefen und praktisch anwenden.

Wir wählten die wissensbasierte Steuerung technischer Prozesse als Anwendungsgebiet aus. Modelliert werden sollte der technische Prozeß selbst, eine autonome Prozeßsteuerungskomponente sowie Einflußmöglichkeiten des Nutzers auf die Steuerung. Außerdem sollten mögliche Konflikte zwischen automatischer Steuerung, Nutzervorgaben und den physikalischen Grenzen der technischen Anlage vom System selbst erkannt und möglichst „günstig“ beseitigt werden. Die Aufgabenstellung, ein erstellter Entwurf sowie das zur Modellierung verwendete Werkzeug werden in Abschnitt 2 kurz vorgestellt. Eine detailliertere Beschreibung findet man in [Döt98].

Realisiert werden sollte dieses Vorhaben von Studenten im Hauptstudium ohne Vorkenntnisse über das zu verwendende Werkzeug innerhalb eines einsemestrigen Praxisseminars. Aus diesem Grund war es nötig, von einer realistischen Prozeßsteuerungsaufgabe zu abstrahieren und sich bei der Modellierung auf einige wichtige Prozeßgrößen, Problematiken und Zusammenhänge zu beschränken.

Einige wichtige Punkte, die bei der Wissensmodellierung von Bedeutung sind, werden in den nächsten Abschnitten erläutert. Die Bedeutung der einzelnen Punkte soll an der zu modellierenden Seminaraufgabe veranschaulicht werden.

2 Anwendungsbeispiel und verwendetes Werkzeug

2.1 Die Aufgabenstellung

Modelliert wurde eine vereinfachte Prozeßsteuerung aus der chemischen Industrie. Eine Steuerungskomponente sowie der Nutzer steuern den im Reaktor ablaufenden Prozeß. Es gibt mehrere Prozeßparameter,

*e-mail: volkerd@informatik.uni-leipzig.de

über die der Zustand im Reaktor beeinflusst werden kann. So wird z.B. durch Erhöhen der Heizung die Temperatur im Reaktor vergrößert. Der aktuelle Zustand im Reaktor, z.B. die aktuelle Temperatur, wird mit Hilfe von Meßsensoren ermittelt, deren Werte als Beobachtungsergebnisse an einer zentralen Meßwert-sammelstelle zur Verfügung stehen. Beim Überschreiten festgelegter Grenzwerte werden entsprechende Alarme aktiviert. Eine gute Beschreibung der Problematiken, zu berücksichtigenden Größen sowie insbesondere die Berücksichtigung von Zeit bei verfahrenstechnischen Steuerungen findet man in [Arn96].

Das Ziel der Prozeßsteuerung ist es, den Prozeß so zu steuern, daß gerade noch kein kritischer Zustand erreicht wird (maximale Auslastung der Anlage).

Da die Zeit innerhalb des Praxisseminars nicht ausreichte, um eine wirklich realitätsnahe Anlagensteuerung zu modellieren, wurden die zugrunde liegenden Modelle von Reaktor, Steuerung, etc. stark vereinfacht. Wir konzentrierten uns nur auf einige Prozeßparameter und Steuerungsmöglichkeiten, verzichteten auf die Berücksichtigung von Zeit und ließen mögliche Störgrößen außer acht. Eine Übersicht über die Modellierungsaufgabe gibt Abbildung 1.

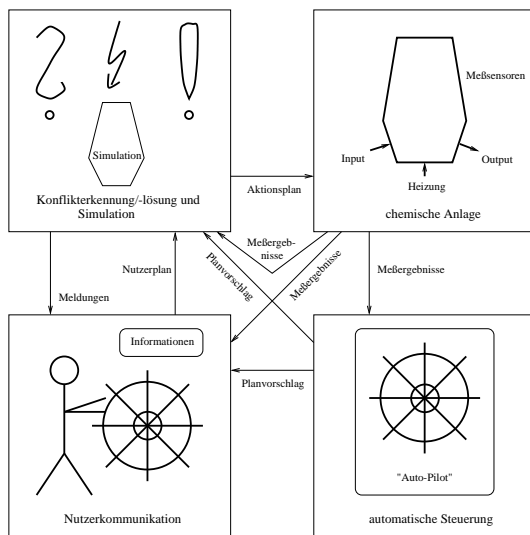


Abbildung 1: Die Modellierungsaufgabe

Zur Nutzerinformation werden die aktuellen Meßwerte und Alarme auf dem Bildschirm angezeigt. Der Planvorschlag der autonomen Steuerungskomponente wird dem Nutzer angezeigt und kann von ihm bestätigt oder mit einem anderen Plan überschrieben werden. Die Auswirkungen der Pläne werden mittels einer Simulationskomponente prognostiziert. Will z.B. der Nutzer den Prozeß in einen „kritischen Bereich“ (aktiver Alarm) steuern, ist ihm dies vom Simu-

lationssystem mitzuteilen und die Planausführung zu verhindern. Haben weder Nutzer noch automatische Steuerungskomponente den Prozeß „unter Kontrolle“, ist ein vordefinierter Standardplan auszuführen, der den Prozeß in einen sicheren Zustand zurückführt bzw. die Anlage „herunterfährt“. Anschließend wird die Prozeßführung wieder vom Nutzer bzw. der automatischen Steuerungskomponente übernommen.

Die Aktionsplanung ist STRIPS-artig. Es gibt Vorbedingungen, die erfüllt sein müssen, um eine Aktion auszulösen. Durch festgelegte Nachbedingungen zu jeder Aktion läßt sich anhand des aktuellen Reaktorzustandes der neue Zustand nach Aktionsausführung berechnen. Betrachtet wird nur der aktuelle Reaktorzustand, der durch die Aktionen verändert wird.

Im allgemeinen versteht man unter Planung die Suche nach Aktionsfolgen (Plan), die einen beschriebenen Ausgangszustand in einen geforderten Zielzustand überführen. In unserer Anwendung würde beispielsweise durch die Folge der drei Aktionen $inc(Heizung)$, $dec(In\ Mass)$, $inc(Out\ Mass)$ die Temperatur ansteigen und der Druck stark sinken.

STRIPS-artige Planungsansätze sind relativ eingeschränkt und lassen die Behandlung bestimmter dynamischer Aspekte nicht zu (siehe [AJ94]). Für komplexe Aufgaben, wie beispielsweise eine wissensbasierte Steuerung realer technischer Prozesse, sind flexiblere Ansätze (vgl. [AJ94]) erforderlich, die vermutlich weit über bisherige Ansätze hinaus gehen werden.

Da es uns bei der Modellierung aber nicht um die Verwirklichung eines ausgefeilten Ansatzes zur Planung ging, sondern um die Anwendung und Übung von Prinzipien der Wissensmodellierung, beschränkten wir uns bei der Planung auf einen STRIPS-artigen Ansatz. Wir lassen auch außer acht, daß die Aktionen im allgemeinen eine Ausführungszeit besitzen. Beispielsweise ist sofort nach Erhöhung der Heizung ($inc(Heizung)$) die Temperatur angestiegen – es dauert also nicht wie in der Realität eine gewisse Zeit, bis die Auswirkungen der Aktionen beobachtbar sind.

2.2 Das verwendete Werkzeug

Wir verwendeten zur Modellierung das wissensbasierte Werkzeug DESIRE (framework for **DE**sign and **Sp**ecification of **Int**eracting **RE**asoning components), welches an der Freien Universität Amsterdam entwickelt wurde und dort auch ständig weiterentwickelt wird. Anfangs wurde DESIRE zur Spezifikation beliebiger wissensbasierter Systeme entwickelt und benutzt, seit einigen Jahren jedoch speziell für den Entwurf von Multiagentensystemen weiterentwickelt. Neben dem User-Manual [Art97] läßt sich auch [BJT96] sowohl als Nachschlagewerk als auch für

eine Einführung zu DESIRE verwenden.

Die Gründe für die Auswahl von DESIRE waren vielfältig (siehe auch [Döt98]). An dieser Stelle seien stellvertretend nur ein paar von ihnen genannt:

- komfortabler grafischer Editor,
- ständig erreichbare Online-Hilfe,
- Trennung verschiedener Wissensarten,
- Verwendung einer mehrsortigen, dreiwertigen Prädikatenlogik,
- Möglichkeiten der Vererbung innerhalb der angelegten Modulhierarchie,
- automatische Generierung einer lauffähigen Implementation aus der erstellten Spezifikation,
- gut ausgebauter Debugger zur Fehlersuche bei gestarteter Implementation.

Eine erzeugte Implementation ist innerhalb der DESIRE-Entwicklungsumgebung ausführbar, so daß der modellierte Sachverhalt (z.B. unsere interaktive Prozeßsteuerung) sofort simuliert werden kann.

2.3 Der erstellte Entwurf

Es wurde eine kompositionale Spezifikation mit zwei Hierarchieebenen erstellt. Die oberste Ebene (Top Level) zeigt Abbildung 2. Man findet die vier Kompo-

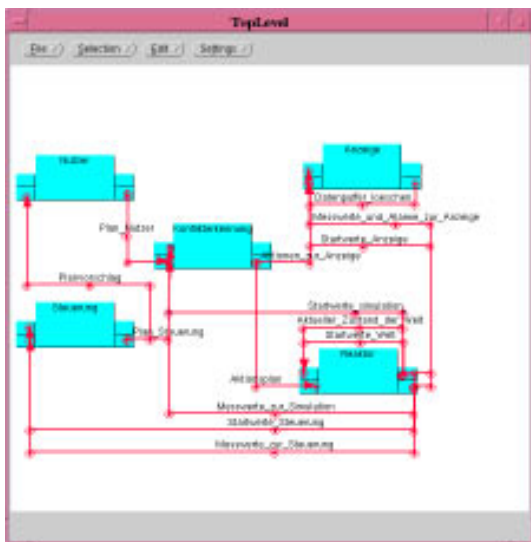


Abbildung 2: Die oberste Ebene des Entwurfes

nenten (Module) der Aufgabenstellung aus Abbildung 1 wieder. Die ständige Anzeige der aktuellen Daten wurde durch einen zusätzlichen Modul realisiert.

In dem Modul *Steuerung* verbergen sich auf tieferer Ebene die Agenten zur automatischen Plangenerierung. Dem Nutzer wird der vorgeschlagene Plan

in einem separaten Fenster auf dem Bildschirm dargestellt. Mit Hilfe weniger Mausklicks kann der Nutzer die vorgeschlagenen Aktionen löschen und eigene Vorgaben in die Aktionsliste eintragen. Innerhalb der *Konflikterkennung* wird auf einer tieferen Modellierungsebene die Ausführung der beiden Pläne simuliert. Die Pläne werden aufgrund gegebener Parameter und der Simulationsergebnisse bewertet. Der „bessere“ Plan bzw. einer der vordefinierten Standardpläne wird zur Ausführung ausgewählt. Die Begründung, warum gerade dieser Plan ausgeführt werden soll, wird mittels des Moduls *Ausgabe* auf dem Bildschirm angezeigt. Der ausgewählte Plan wird zu den Modulen *Reaktor* und *Anzeige* übertragen. Im Anzeigemodul wird dem Nutzer der auszuführende Plan angezeigt. Dies ermöglicht ein leichtes Nachvollziehen der Änderungen des Zustandes im Reaktor. Im Modul *Reaktor* wird der Plan ausgeführt. Mögliche Druck- und Temperaturänderungen werden mit Hilfe der internen Wissensbasis abgeleitet. Anschließend wird der neue Zustand im Reaktor „berechnet“, mit den vorgegebenen Grenzwerten für die Alarmer verglichen und diese entsprechend ein- bzw. ausgeschaltet. Nun werden wieder die aktuellen Meßdaten systemweit an die betreffenden Module verteilt und die Steuerungskomponente erstellt einen Plan, der dem Nutzer vorgeschlagen wird, usw.

Alle Module enthalten eigene Wissensbasen, in denen das zur Lösung der jeweiligen Aufgabe nötige Wissen in Form von Fakten und Regeln enthalten ist. Dabei wird in DESIRE strikt zwischen Domänenwissen und Steuerungswissen getrennt. Die *komplexen* Module (Steuerung, Top-Level, Konflikterkennung) enthalten Wissensbasen zur Ablaufsteuerung, die restlichen Module das Domänenwissen. Zum Domänenwissen gehört beispielsweise, unter welchen Bedingungen und in welchem Maße die Reaktortemperatur bei Aktionsausführung steigt oder fällt, usw. Die Steuerung betrifft den gesamten zeitlichen Ablauf, z.B. daß nur unmittelbar nach dem Start einer Simulation initialisiert werden muß; daß in dem Modul *Steuerung* erst der gesamte Plan generiert wird, bevor er dem Nutzer vorgeschlagen wird; daß erst beide Pläne simuliert werden müssen, um sie anschließend zu übertragen und zu vergleichen; usw.

Trotz der getroffenen Einschränkungen war die Modellierung nicht trivial. Die Aufgabenstellung vereinfachte sich aber zumindest so weit, daß die Modellierung sowie die Einarbeitung in DESIRE in der zur Verfügung stehenden kurzen Zeit möglich war.

Selbst bei diesem, gemessen an realen Aufgabenstellungen, sehr kleinen Entwurf wurde deutlich, wie wichtig eine adäquate Wissensrepräsentation beim Entwurf wissensbasierter Systeme für praktische An-

wendungen ist. Das verwendete Tool muß dafür geeignete Unterstützung bieten, insbesondere bei komplexen Entwürfen.

3 Schwerpunkte beim Entwurf

Schlagworte, die man im Zusammenhang mit Wissensmodellierung oft hört oder liest sind zum Beispiel Modularität, Erweiterbarkeit und Wiederverwendbarkeit. Was verbirgt sich hinter solchen und anderen Schlagworten und wie wichtig sind diese Eigenschaften? Wir wollen hier jedoch nicht der Frage nachgehen, was denn „Wissen“ z.B. eines Experten ist, worin es besteht oder wie man es erkennen, umschreiben und auch beschreiben kann. Auch die Ausdrucksfähigkeit von Wissensrepräsentationssprachen wird hier nicht untersucht, obwohl klar ist, daß eine adäquate Wissensmodellierung sehr von den Stärken und Schwächen des verwendeten Systems abhängt.¹ Auch die nachfolgende Nennung der Schwerpunkte bei der Wissensmodellierung läßt sich sicher noch um den einen oder anderen Punkt ergänzen.

Der gesamte Entwurf (also auch die Wissensmodellierung) wird bei komplexen Anwendungen im Team erstellt. Das benutzte Entwurfswerkzeug sollte Teamarbeit geeignet unterstützen. Bei DESIRE wird der gesamte Entwurf in *einer* Datei gespeichert, somit war keine Teamarbeit möglich – deshalb auch die stark eingeschränkte Aufgabenstellung. Um die gesamte Aufgabe des wissensbasierten Entwurfes auf ein Team zu verteilen, muß man diese Gesamtaufgabe in entsprechende Teilaufgaben zerlegen. Gemeint ist hier nicht die Teilung in Wissenserhebung, Modellierung, Verifikation, usw., sondern die Modularität des Entwurfes. In unserem Beispiel war es die Überlegung, wieviel und welche Module (Steuerung, Konflikterkennung, ...) sind sinnvoll, und was soll sich hinter den einzelnen Modulen verbergen? Diese Entscheidungen hat der Nutzer bereits zu Beginn der Modellierung zu treffen, obwohl oft noch nicht die ganze Tragweite der Entscheidungen überblickt wird. Hier sind Erfahrungen, Fingerspitzengefühl und gute Kenntnisse über die Struktur des zu modellierenden Wissens nötig, um die festgelegte Struktur nicht im nachhinein grundlegend ändern zu müssen.

Das Einführen von Abstraktionsniveaus (Module

werden in Submodule zerlegt) beim Entwurf ist vorteilhaft. Legt man alle Module in dasselbe Abstraktionsniveau, entsteht ein zu flacher und unübersichtlicher Entwurf. DESIRE unterstützt eine modulare Entwurfstechnik indem zwischen *komplexen* und *primitiven* Modulen unterschieden wird. Komplexe Module bestehen aus Submodulen und organisieren deren Zusammenarbeit und Kommunikation sowie den Datenaustausch mit den übergeordneten Modulen. Primitive Module werden nicht weiter zerlegt und dienen zur Lösung der entsprechenden Aufgaben. Durch eine solche Zerlegung von Modulen kann man zwischen Abstraktions- und Spezifikationsebenen auch innerhalb einzelner Module unterscheiden.

Jeder Modul ist in sich abgeschlossen, d.h. er besitzt seine eigene Signatur (Sorten, Objekte, Relationen, ...), Wissensbasis und Datenschnittstellen. Der Datenaustausch geschieht nur über die standardisierte Input/Output-Schnittstellen. Nach der Definition der Schnittstellen können die Module durch verschiedene Teammitglieder unabhängig voneinander implementiert werden. In solchen kompositionalen Entwürfen können bei Bedarf einzelne Module problemlos ersetzt werden, wenn der neue Modul die geforderte Funktionalität bereitstellt.

Ein weiterer wichtiger Punkt bei der Modularisierung ist die Trennung der verschiedenen Wissensarten. Dies erhöht nicht nur die Übersichtlichkeit, sondern ermöglicht ein schnelles Hinzufügen, Verändern oder Löschen einzelner Wissensstücke (in der Literatur stellenweise auch als Flexibilität bezeichnet). Clancey unterscheidet in [Cla83] strategisches Wissen (über die Vorgehensweise des Systems), strukturelles Wissen (über Beziehungen zwischen Objekten) und unterstützendes Wissen (das Zusatzwissen für Erklärungszwecke umfaßt). Häufig wird auch beispielsweise zwischen statischem und dynamischen Wissen, domänenspezifischem und nicht domänenspezifischem Wissen sowie strategischem und aufgabenbezogenem Wissen unterschieden. In DESIRE unterscheidet man zwischen Wissen auf Objektebene, Metaebene und Steuerungsebene. Steuerungswissen wird in den Wissensbasen der komplexen Module beschrieben, Wissen der Objektebene hingegen in primitiven Modulen. Die Möglichkeiten der Spezifikation von Steuerungswissen in den entsprechenden Wissensbasen sind derart vielfältig (z.B. einfache Fakten, Regeln mit verschiedenen temporalen, quantitativen oder auch qualitativen Bedingungen), daß an dieser Stelle nur auf [BJT96] verwiesen werden kann.

Unterstützung bei der Unterscheidung von statischem und dynamischen Wissen auf Objektebene wäre wünschenswert gewesen. So mußten Grenzwerte für die Schaltung der Alarme in derselben

¹Eine Checkliste für Wissensrepräsentationsformalismen, um die Mächtigkeit eines allgemeinen Expertensystemwerkzeuges zu bewerten, findet man im Anhang von [Pup91]. Die Auswahl eines geeigneten Entwurfswerkzeuges ist nicht einfach, da viele solcher Werkzeuge existieren, aber nur wenig vergleichende, praktische Erfahrungsberichte wie z.B. [HK87], [KKP87], [HMM88], [SHL+93], [TW92] oder [PPR+98] über den Umgang mit diesen Werkzeugen publiziert sind.

Wissensbasis festgelegt werden, in der auch die Regeln für die Zustandsänderung im Reaktor definiert wurden. Eine bessere Trennung käme auch der Wiederverwendbarkeit² von Modulen zugute. Hätte man z.B. ein feineres Reaktormodell (mehr Parameter, stärkere gegenseitige Beeinflussung), so ist dies unabhängig von etwaige Gefahrenstufen (Alarmen) oder ähnlichem. Eine Trennung derartigen Wissens hätte weiterhin den Vorteil, daß die einzelnen Wissensbasen kompakter (überschaubarer, handhabbarer und kleiner) gehalten werden können. Dies wiederum wirkt sich positiv auf die zur Modifikation und bei der Inferenz benötigte Zeit aus. Sollten sich viele und große Wissensbasen nicht vermeiden lassen, ist die Möglichkeit des Nachladens benötigter Wissensbasen bei Bedarf sehr vorteilhaft. Eine weitere Strukturierung der Wissensbasen durch Bilden von Regelgruppen oder durch das Priorisieren bestimmter Regeln kann in bestimmten Fällen bedeutsam sein, wird jedoch von DESIRE nicht unterstützt. Bei manchen Entwurfssystemen hat man auch die Möglichkeit den Ableitungsmechanismus explizit zu steuern. In DESIRE kann man in den einzelnen Modulen zwar die Ableitungsziele dynamisch (per Bedingung) vorgeben, die Strategie selbst (z.B. forward oder backward chaining) jedoch nicht beeinflussen.

In einigen Fällen erscheint die dynamische Wissensstrukturierung (erst zur Laufzeit des Systems) von großer Bedeutung zu sein. Zum Beispiel könnten verschiedene Sichtweisen (von einem Facharzt oder einem Spezialisten) auf denselben Sachverhalt mit Hilfe von dynamischem Umstrukturieren (Zerlegen oder Zusammenfassen) von Modulen modelliert werden. Auch sind Situationen vorstellbar, in denen ein dynamisches Strukturieren von Wissensbasen (Bilden von Regelgruppen oder Vergeben von Prioritäten) nötig ist. Bei solchen dynamischen Aspekten der Wissensstrukturierung ist der Nutzer auf die Unterstützung durch das Entwurfssystem angewiesen. In unserem kleinen Anwendungsbeispiel waren solche dynamischen Strukturierungen nicht nötig, wären von DESIRE auch nicht unterstützt worden.

Die Bedeutung einer adäquaten Wissensrepräsentation kann sehr schön an dem „klassischen“, medizinischen Diagnosesystem INTERNIST-1 illustriert werden. Mit INTERNIST-1 können ca. 75% der Krankheiten in der Inneren Medizin diagnostiziert werden. Das System zeichnet sich besonders durch die Größe seiner Wissensbasis aus. Im Kapitel 18 von [Pup91] schreibt Puppe über die in [MPM82] beschriebene Evaluation von INTERNIST-1 folgendes: „Probleme

sind vor allem die Darstellung und Auswertung von zeitlichem, funktionalem und anatomischem Wissen und die Repräsentation verschiedener Abstraktionsebenen (Sichten) von medizinischem Wissen. Die Bedeutung dieser Mängel zeigt die Evaluation von INTERNIST, bei der über die Hälfte der falschen Diagnosen auf Schwächen der Wissensrepräsentation und nur etwa 1/3 auf Fehler in der Wissensbasis zurückgeführt wurden ([MPM82], Tabelle 5).“

4 Zusammenfassung, Ergebnis

Das Verwenden von Abstraktionsniveaus (Aufgabenzerlegung) und die Modularisierung des Wissens innerhalb der Module ist von entscheidender Bedeutung für die Wartbarkeit, Effizienz und Adäquatheit des entworfenen wissensbasierten Systems. Unter dem Begriff Modularisierung lassen sich viele oben genannte Kriterien zusammenfassen.

Bereits bei unserem kleinen Beispiel der Prozeßsimulation zeigte sich die Bedeutung fast aller oben genannter Schwerpunkte. Wieviel mehr gewinnen sie bei komplexeren Problemstellungen an Bedeutung.

Die Unterstützung von Modularität und Wissensstrukturierung durch DESIRE erwiesen sich als vorteilhaft, wird wahrscheinlich aber im allgemeinen, insbesondere bei realistischen Anwendungen nicht ausreichen. Es kann vermutet werden, daß es sich bei anderen Systemen zum Entwurf wissensbasierter Systeme ähnlich verhält.

Besonders anschaulich charakterisiert Jantke in [Jan96], wann eine gefundene Wissensrepräsentation *adäquat* erscheint: „Wenn es gelingt, daß Strukturen des dargestellten Wissens sich in den Strukturen der Repräsentation wiederfinden, und wenn die Wissensverarbeitung auf der Ausnutzung dieser Strukturierung beruht, dann scheint für den jeweiligen Anwendungsfall eine adäquate Wissensrepräsentation gefunden worden zu sein.“

Die Bedeutung einer adäquaten Wissensrepräsentation ist sicher unumstritten. Anleitungen und Systeme für den Entwurf wissensbasierter Systeme bieten in vielen Fällen bereits Unterstützung im Entwurfsprozeß. Von einer „Schritt für Schritt“-Anleitung beim Entwurf, wie zum Beispiel für das Lösen von Gleichungssystemen, scheint man aber noch weit entfernt. Hier sind die Grundlagenforschung sowie die schnelle Umsetzung neuer Erkenntnisse bei der Weiterentwicklung entsprechender Entwurfssysteme gefragt.

²Beim Unterstützen der Wiederverwendbarkeit von Entwürfen durch das Konzept der *generic task models* sei hier nur auf [Cha86] und [Cha88] verwiesen.

Literatur

- [AJ94] O. Arnold and K.P. Jantke. Therapy plans as hierarchically structured graphs. In *Int. Workshop on Graph Grammars and their Applic. to Computer Science, Williamsburg, USA*, Nov. 1994.
- [Arn96] O. Arnold. *Die Therapiesteuerungskomponente einer wissensbasierten Systemarchitektur für Aufgaben der Prozeßführung*, vol. 130 of *Diss. zur KI*. Infix, 1996.
- [Art97] AI Group, Dep. of Computer Science, Free Univ. Amsterdam. *Desire Graphical Editor and General Manager System for DESIRE Specifications - User Manual*, 1997.
- [BHJ⁺94] B. Busch, T. Herrmann, K. Just, and M. Rittenbruch. *Systeme für Experten statt Expertensysteme*. Infix Verlag, 1994.
- [Bib93] W. Bibel. *Wissensrepräsentation und Inferenz*. Vieweg Verlag, Braunschweig, 1993.
- [BJT96] F. Brazier, C. Jonker, and J. Treur. *Syllabus - Ontwerp van Kennissystemen*. AI Group, Dep. of Computer Science, Free Univ. Amsterdam, Netherlands, 1996.
- [Cha86] B. Chandrasekaran. Generic tasks in knowledge based reasoning: high-level building blocks for expert system design. *IEEE Expert, Fall 1986*, 1(3):23–30, 1986.
- [Cha88] B. Chandrasekaran. Generic Tasks as Building Blocks for Knowledge-Based Systems: The Diagnosis and Routine Design Examples. *The Knowledge Engineering Review*, 3(3):183–210, 1988.
- [Cla83] W.J. Clancey. The Epistemology of a Rule-Based Expert System: A Framework for Explanation. *AI*, 20(3):215–251, 1983.
- [Döt98] V. Dötsch. Modellierung einer wissensbasierten, interaktiven Prozeßsteuerung. Rep. 07-1998, Univ. Leipzig, Juni 1998.
- [GS96] J. Gamper and F. Steimann. Medizinische Expertensysteme - Eine kritische Betrachtung. *APIS-Zeitschr. f. Polit., Ethik, Wissensch. u. Kultur im Gesundheitsw.*, 1996.
- [Hel96] B. Heller. *Modularisierung und Fokussierung erweiterbarer komplexer Wissensbasen auf der Basis von Kompetenzeinheiten*, vol. 120 of *DISKI, Dissertationen zur Künstlichen Intelligenz*. Infix, 1996.
- [Hen91] R.-D. Hennings. *Informations- und Wissensverarbeitung - Theoretische Grundlagen Wissensbasierter Systeme*. Walter de Gruyter, Berlin, 1991.
- [HK87] P. Harmon and D. King. *Expertensysteme in der Praxis*. Oldenbourg Verl., 1987.
- [HMM88] P. Harmon, R. Maus, and W. Morrissey. *Expert Systems Tools and Applications*. John Wiley & Sons, New York, 1988.
- [Jan96] K.P. Jantke. Wissensrepräsentation und die moderne Informatik. Forschungsber. FITL-96-6, Forschungsinst. für InformationsTechnologien Leipzig, Sept. 1996.
- [KKP87] D. Karras, L. Kredel, and U. Pape. *Entwicklungsumgebungen für Expertensysteme*. Walter de Gruyter, Berlin, 1987.
- [MPM82] R. Miller, H. Pople, and J. Myers. INTERNIST-1, An Experimental Computer-based Diagnostic Consultant for General Internal Medicine. *New England Journal of Medicine*, 307(8):468–476, 1982.
- [PPR⁺98] F. Puppe, B. Puppe, B. Reinhardt, S. Schewe, and H.P. Buscher. Evaluation medizinischer Diagnostik-Expertensysteme zur Wissensvermittlung. *Informatik, Biometrie und Epidemiologie in Medizin und Biologie*, 29(1):48–59, 1998.
- [Pup91] F. Puppe. *Einführung in Expertensysteme*. Springer-Verlag, Berlin, 1991.
- [SHL⁺93] S. Schlegelmilch, B. Heller, T. Linke, and J. Meyer-Fujara. Evaluation hybrider Expertensystemtools. Reportreihe Modularisierung wissensbasierter Systeme MOSYS-13, Univ. Bielefeld, März 1993.
- [Str91] P. Struß. *Wissensrepräsentation*. Oldenbourg Verlag, 1991.
- [TW92] J. Treur and T. Wetter. *Formal Specification Of Complex Reasoning Systems*. Ellis Horwood Ltd., London, 1992.
- [WF88] T. Winograd and F. Flores. *Understanding Computers and Cognition: A New Foundation for Design*. Ablex, Norwood, N.J., USA, 1988.