Forschungsinstitut für
anwendungsorientierte
Wissensverarbeitung

an der Universität Ulm

Beiträge zum 7.Arbeitstreffen der GI-Fachgruppe 1.1.5/7.0.1 "Intelligente Lehr-/Lernsysteme" am Forschungsinstitut für anwendungsorientierte Wissensverarbeitung (FAW) Ulm

herausgegeben von R. Gunzenhäuser, C. Möbus und D. Rösner

FAW-TR-94003
Mai 1994

FAW-Dokumentation

Helmholtzstraße 16
D-89081 Ulm

Postfach 2060
D-89010 Ulm

PETRI-HELP:
Expertiseerwerb, Hilfeakzeptanz und Systemerweiterungen*

Jörg Folckers, Claus Möbus, Knut Pitschke, Olaf Schröder
Universität Oldenburg, Fachbereich Informatik, Abt. Lehr- Lernsysteme

PETRI-HELP ist ein Intelligentes Hilfesystem zur wissensbasierten Unterstützung von Personen bei der Modellierung mit Petrinetzen des Bedingungs-Ereignis-Typs (Möbus et al., 1993). Insbesondere Novizen sollen darin unterstützt werden, sich in die Modellierung mit Netzen einzuarbeiten. PETRI-HELP basiert auf der theoretischen Konzeption, daß

- Wissenserwerb durch Einsatz schwacher Heuristiken nach Stocksituationen stattfindet
- erfolgreich eingesetztes Wissen optimiert wird
- Problemlöseprozesse in die Phasen des Abwägens, Planens, Ausführens und Bewertens gegliedert werden können (Möbus et al., 1992; 1993).

Im Sinne dieser Konzeption stellt PETRI-HELP eine explorative Problemlöseumgebung dar, in der der Lernende in Stocksituationen zu seinen Netzentwürfen Hypothesen testen und auf Hilfeinformationen zugreifen kann.

Thema dieses Beitrags ist die Frage, wie PETRI-HELP den Erwerb von Wissen zur Modellierung mit Netzen fördert, und welche Rolle die Hilfen dabei spielen. Im ersten Teil wird ein kurzer Überblick über den Implementationsstand von PETRI-HELP gegeben. Im zweiten Teil werden empirische Einzelfalluntersuchungen dargestellt, die zum Ziel hatten, unsere Hypothesen darüber weiter zu konkretisieren, worin Modellierungsexpertise besteht, wie sie erworben wird und wie die von PETRI-HELP bereitgestellten Hilfen genutzt und von den Personen akzeptiert werden. Aus diesen Untersuchungen sowie aus unserer theoretischen Konzeption ergeben sich Erweiterungen von PETRI-HELP, die gegenwärtig implementiert werden. Sie werden im dritten Teil beschrieben.

1. Implementationsstand

PETRI-HELP besteht aus folgenden Komponenten (vgl. ausführlich Möbus et al., 1993; Pitschke, Möbus, 1993):

- einer Sequenz von zehn *Modellierungsaufgaben*. Jede Modellierungsaufgabe ist als Menge temporallogischer Formeln formuliert, die ein einfaches technisches, natürliches oder soziales System spezifizieren (Restaurant, Reparaturwerkstatt, Photosynthese, Automatisches Sperrdifferential etc.).

- einem *Editor* für die Konstruktion und Simulation von Bedingungs-Ereignis-Netzen

- einer *Hypothesentestumgebung*, in der der Lernende Hypothesen darüber formulieren kann, welche Formeln der Aufgabenspezifikation er durch den aktuellen Stand seines Lösungsentwurfs für erfüllt hält. Dies geschieht, indem der Lernende die entsprechenden Formeln der Aufgabenspezifikation auswählt. Das System überprüft dann die Hypothese durch Model-checking und informiert den Lernenden, welche Formeln der Hypothese erfüllt sind und welche nicht.

* gefördert durch die Stiftung Volkswagenwerk (Az. 210-70631/9-13-14/89)

• einer Komponente, die dem Lernenden *Vervollständigungs-* und *Korrekturvorschläge* für seinen Entwurf anbietet. Dem Lernenden werden bei Bedarf Stellen, Transitionen und Kanten vorgeschlagen, mit denen er seinen Entwurf erweitern kann und die ggf. zu löschen sind. Die Ergänzungsvorschläge beruhen auf Regeln, die das System in der Interaktion mit seinen Benutzern lernt (vgl. Möbus et al., 1993). Ergänzungs- bzw. Korrekturvorschläge werden so angeboten, daß, wenn der Benutzer sie in seinem Entwurf realisiert, eine möglichst kleine Obermenge der aktuell erfüllten Formeln erfüllt ist.

Hypothesentesten mit Rückmeldung und Vervollständigungs- / Korrekturvorschläge sind die in PETRI-HELP verfügbaren Hilfen.

2. Empirische Untersuchungen zu Expertiseerwerb und Hilfeakzeptanz

PETRI-HELP unterstützt Novizen beim Erwerb von Modellierungsexpertise, indem es Hypothesentesten mit Rückmeldung ermöglicht und Vervollständigungs- / Korrekturvorschläge bereitstellt. Was jedoch in PETRI-HELP fehlt und auch in Lehrbüchern zu Petrinetzen (z.B. Reisig, 1986; 1992) kaum thematisiert wird, ist eine *Designtheorie* der Entwicklung von Petrinetzen. Der Model-checking-Ansatz ermöglicht die Analyse *bereits konstruierter* Netzfragmente, unterstützt aber nicht den zeitlich vorgelagerten Prozeß ihrer Konstruktion. Daher sollte u.E. der von uns in PETRI-HELP realisierte Model-checking-Ansatz durch eine Designtheorie ergänzt werden. Wir versuchen, eine Designtheorie empirisch aus den Vorgehensweisen von Personen bei der Entwicklung von Netzen zu gewinnen. Eine solche Designtheorie kann dann zur Gewinnung von Planungshilfen für Petrinetzmodellierer genutzt werden. In empirischen Einzelfallstudien wurde daher untersucht,

- worin das von den Personen erworbene Modellierungswissen (Designwissen) besteht,
- wie dieses Wissen erworben wird.

Ein weiteres Ziel unserer empirischen Untersuchungen der Frage, welches Wissen Personen bei der Arbeit mit PETRI-HELP erwerben und wie dieses Wissen erworben wird, ist die Entwicklung einer *automatisierten Wissensdiagnose*. Ein sich online entwickelndes Lernermodell ist Voraussetzung für individualisierte, an das aktuelle Wissen des Benutzers angepaßte Hilfen (Möbus et al., 1993; Pitschke, 1993; Pitschke, Möbus, 1993). Zum anderen können Fragen nach der Akzeptanz und Effektivität der Hilfen sowie Fragen nach den möglichen Ursachen für intra- und individuelle Unterschiede in der Akzeptanz nur vor dem Hintergrund eines solchen Modells sinnvoll beantwortet werden. Möbus, Schröder & Thole (1992) haben gezeigt, daß ein Modell des Wissensstandes des Benutzers zu detaillierten Vorhersagen hinsichtlich Akzeptanz und Wirksamkeit von Hilfen führt.

Eine ähnliche Zielsetzung verfolgt Spohrer (1992) bei der Entwicklung von MARCEL, einer ablauffähigen Simulation eines Programmiernovizen. Während MARCEL aber den Einsatz *erworbenen* Wissens zur Problemlösung modelliert, interessiert uns der *Erwerb* von Wissen.

Eine automatisierte online-Wissensdiagnose setzt voraus, daß für die Anwendung von Wissensbeständen und von schwachen Heuristiken, für Stocksituationen usw. rechnerseitig erfaßbare Indikatoren definiert werden. Auf solche Indikatoren wird im folgenden Bezug genommen. In zukünftigen Arbeitsschritten müssen sie weiterentwickelt und anhand weiterer Daten wie z.B. Verbalisationen validiert werden.

Ein drittes Ziel unserer empirischen Untersuchungen bezieht sich schließlich auf die Frage, wie die von PETRI-HELP bereitgestellten Hilfen - Hypothesentesten mit Rückmeldung erfüllter / nicht erfüllter Formeln sowie Korrektur- / Ergänzungsvorschläge - von Personen genutzt und akzeptiert werden.

2.1 Zum Erwerb von Modellierungsexpertise

Zwei Personen arbeiteten in Einzelsitzungen mit dem jetzigen Implementationsstand von PETRI-HELP. Ihre Handlungen und Verbalisationen wurden mit der Videokamera aufgenommen. Die Protokolle wurden im Hinblick auf die Vorgehensweise, das erworbene Modellierungswissen sowie die aufgetretenen Stocksituationen und Heuristiken ausgewertet.

Zur Vorgehensweise. Im Hinblick auf die Vorgehensweise der Personen wurden die Videoprotokolle gemäß unserer theoretischen Konzeption in Phasen des *Abwägens* (Abwägen zwischen den als nächstes zu bearbeitenden Formeln mit dem Resultat der Auswahl einer Formelmenge zur Bearbeitung), des *Planens* einer Realisierung für die gewählte Formelmenge, der *Ausführung* des Plans sowie der *Bewertung* des Ergebnisses segmentiert. Dabei zeigte sich, daß ein Großteil der Protokolle als Aufeinanderfolge von folgenden Schritten oder Problemlöseoperatoren im Sinne von Newell & Simon (1972) besteht:

Operator **FA** (Fortschaltbedingung auswählen): Wahl einer Fortschaltbedingung,
- die möglichst wenige atomare Formeln enthält
- für deren atomare Formeln bereits möglichst viele Stellen realisiert sind (*abwägen*)

Planung der Realisierung der Fortschaltbedingung gemäß des verfügbaren Modellierungswissens (s.u.) (*planen*): Abrufen eines Designheuristik-Schemas (Operator **DA**) oder Planen mit elementareren Designbausteinen (Operator **DB**).

Operator **RE**: Realisierung des Plans (*ausführen*), d.h. der Netzentwurf wird um Stellen, Transitionen und Kanten erweitert.

Bewertung des Ergebnisses durch Formulierung und Prüfung einer Hypothese über die aktuell erfüllten Formeln (Operator **HT**: Hypothesentesten), durch "Augenschein" (**AU**, z.B. mentale Simulation des Netzes), oder durch Rechnersimulation des Netzes (**SI**) (*bewerten*)

Zum Modellierungswissen. In den Operatoren "DA" und "DB" ist das eigentliche domänenspezifische Modellierungswissen enthalten. Das dem Operator "DA" zugrundeliegende hypothetische Wissen besteht aus Schemata, die jeweils einer Formel ein Netzfragment zuordnen. Diese Schemata bezeichnen wir als *Designheuristiken*. Sie können als Ergebnisse von Problemlöseprozessen betrachtet werden, die wir ebenso wie empirische Indikatoren weiter unten diskutieren. Abb. 1 zeigt Beispiele. Die Designheuristik in Abb. 1 oben links besagt z.B., daß Formeln, die dem Schema "$\Box\ X \rightarrow \Diamond\ Y$" entsprechen, durch eine Transition mit dem Vorbereich {X} und dem Nachbereich {Y} realisiert werden ($\Box$ ist der "always"-Operator: "Es gilt immer, daß ...". $\Diamond$ ist der "eventually"-Operator: "Es gibt einen zukünftigen Zeitpunkt, zu dem gilt, daß ...") Die Designheuristiken wurden empirisch durch Untersuchung der Differenz der Hypothesen in jeweils zwei aufeinanderfolgenden Bewertungsphasen gewonnen: Z.B. hat die Person den Netzzustand N_i

realisiert und die Hypothese H_i formuliert. Anschließend erweitert sie den Netzzustand durch Edieren weiterer Stellen, Transitionen und Kanten zum Netzzustand N_{i+1} und formuliert die Hypothese H_{i+1}. Nun kann die Formeldifferenz $H_{i+1} - H_i$ mit der Netzdifferenz $N_{i+1} - N_i$ in Beziehung gesetzt werden. Diese Zuordnung lieferte in etwa 70% der Fälle unsere Designheuristiken. Empirische Einzelfalluntersuchungen mit einem frühen "Papier-und-Bleistift"-Entwicklungsstadium von PETRI-HELP ergaben ähnliche Ergebnisse (Möbus et al., 1992).

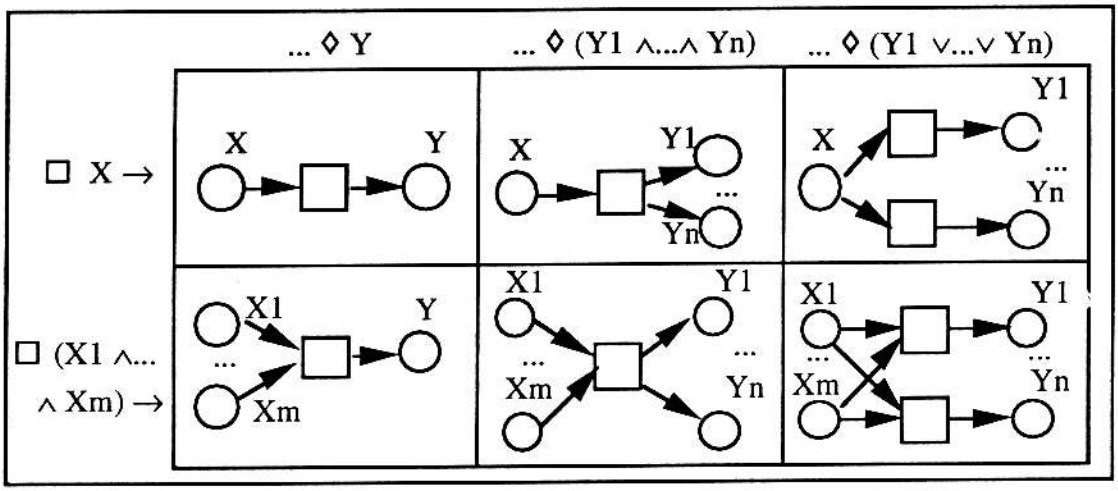

Abb. 1: Einige empirische Designheuristiken für PETRI-HELP

Als rechnerseitig erfaßbaren Indikator für eine Designheuristik betrachten wir also die Formeldifferenz zwischen zwei aufeinanderfolgenden Hypothesentests (H_i, H_{i+1}) und die Netzerweiterung zwischen diesen Hypothesentests. Ein weiterer Indikator sind Verschiebehandlungen. Wir vertreten folgende Hypothese (vgl. auch Möbus, Schröder & Thole, 1992): Wird eine Stelle oder Transition verschoben, so erfolgte ihre ursprüngliche Placierung mit einer anderen Designheuristik als der aktuell angewendeten. Denn anderenfalls hätte die Person die betreffende Stelle bzw. Transition von vornherein so placiert, daß keine Verschiebehandlung notwendig wird.

Auffallend an den Designheuristiken ist, daß die Personen meistens nur eine Formel zur Zeit bearbeiteten (vgl. Abb. 1, d.h. die Formeldifferenz $H_{i+1} - H_i$ enthält nur eine Formel), obwohl die gleichzeitige Bearbeitung mehrerer Formeln (also die Konstruktion von Teilsystemen) in PETRI-HELP in keiner Weise ausgeschlossen ist.

Abb. 1 zeigt darüber hinaus, daß man sich die Designheuristiken als aus elementareren Wissensbeständen zusammengesetzt denken kann. Es ist ja schon eine gewisse Expertise, wenn man eine Formel in ein Netzfragment umsetzen kann. Unsere Hypothese ist daher, daß man die Designheuristiken als "chunks" auffassen kann, die aus Formel- und Netzbausteinen aufgebaut sind (Produktionen einer heuristischen Ziel-Mittel-Relation). Die elementareren Wissensbestände, die zu diesen "chunks" führen, könnten aussehen wie die in Abb. 2 bis 4 dargestellten *Designbausteine*. Abb. 2 beschreibt die Entwicklung von Netzfragmenten durch Designbausteine im Sinne der Designheuristik in Abb. 1 oben links. Abb. 3 beschreibt auch die Entwicklung von Netzfragmenten im Sinne der Designheuristik in Abb. 1 oben Mitte.

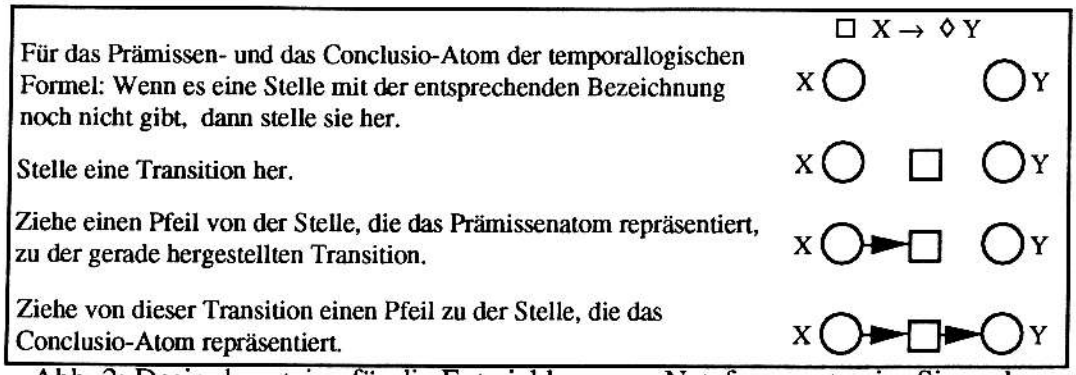

Abb. 2: Designbausteine für die Entwicklung von Netzfragmenten im Sinne der empirischen Designheuristik in Abb. 1 oben links

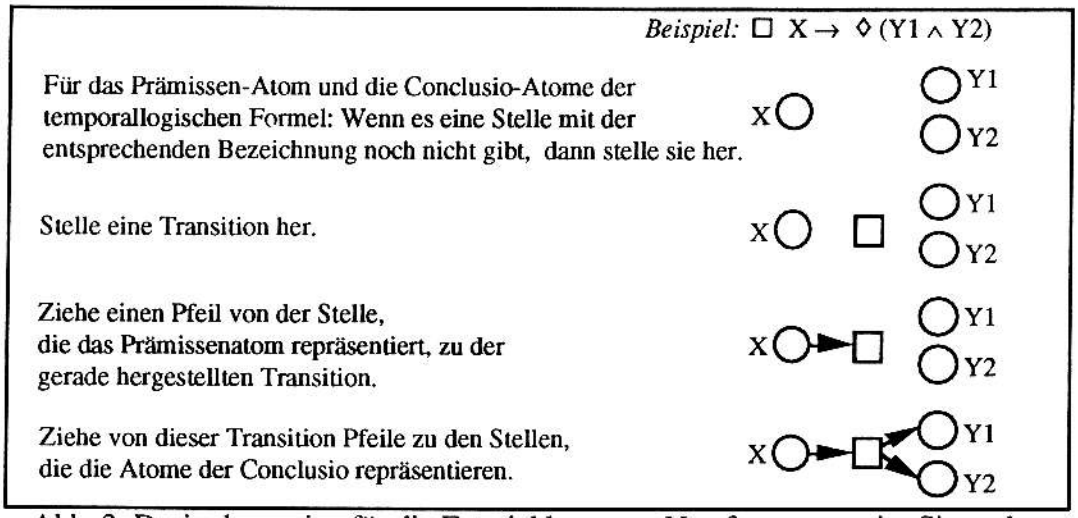

Abb. 3: Designbausteine für die Entwicklung von Netzfragmenten im Sinne der empirischen Designheuristiken in Abb. 1 oben links und oben Mitte

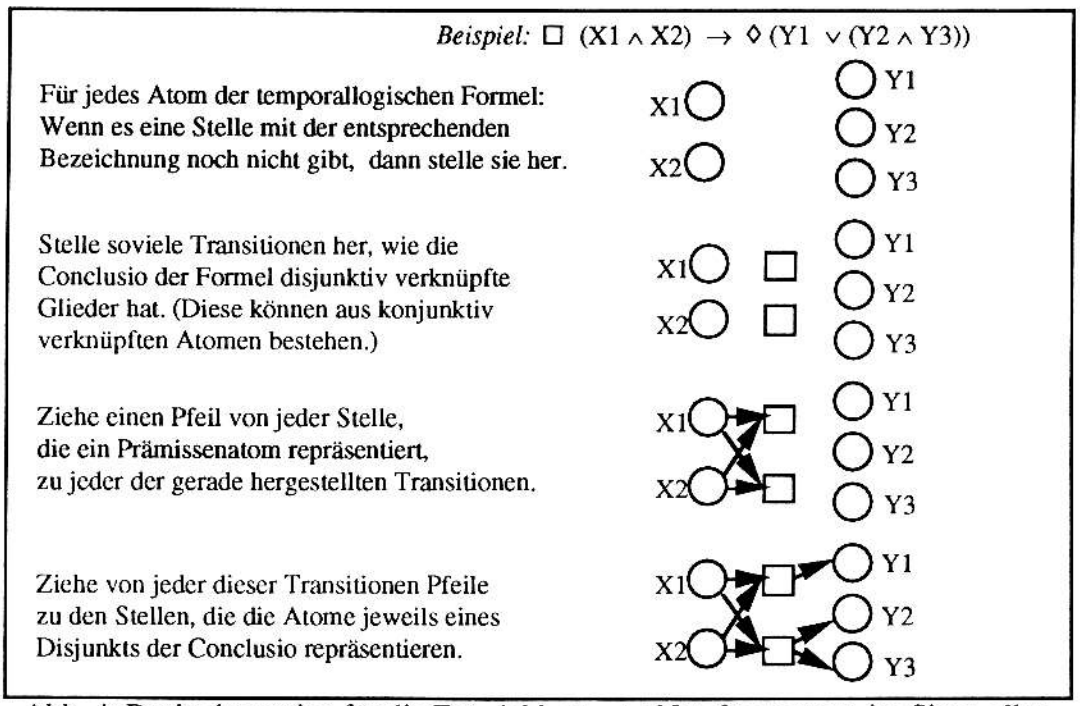

Abb. 4: Designbausteine für die Entwicklung von Netzfragmenten im Sinne aller in Abb. 1 dargestellten empirischen Designheuristiken

Der Übergang von Abb. 2 nach Abb. 3 kann als *Generalisierung* der ersten Regel (von einem auf mehrere Conclusio-Atome) sowie entsprechend der vierten Regel (von einem auf mehrere Pfeile zu den Stellen des Nachbereichs der Transition) beschrieben werden. Abb. 4 umfaßt alle in Abb. 1 dargestellten Designheuristiken. Der Übergang von Abb. 3 zu Abb. 4 kann durch weitere Generalisationen beschrieben werden, wobei hier nicht dargestellte Zwischenstufen möglich sind. Die in den Abb. 2 bis 4 dargestellten Regeln könnten auch durch einen Generalisationsgraphen (z.B. Goldstein. 1982) dargestellt werden.
Die in Abb. 2 bis 4 dargestellten hypothetischen Designbausteine liegen dem Operator "DB" zugrunde. "DA" ist also das Ergebnis von Wissensoptimierungsprozessen über dem elementareren Operator "DB".

Zu Stocksituationen und Heuristiken. Stocksituationen und Heuristiken stellen weitere Protokollkategorien dar. Sie sind zusammen mit den oben genannten Operatoren sowie weiteren Operatoren in Tabelle 1 zusammengestellt. Wie oben erwähnt, entsprechen die Protokollkategorien unseren theoretischen Modellphasen und -prozessen. Die Protokollkategorien sind also Abstraktionen des Protokolls. Für jede Protokollkategorie müssen daher mögliche empirische Indikatoren angegeben werden, wie in Tabelle 1 geschehen[1]. Dabei ist allerdings zu beachten, daß nicht alle Kategorien den von uns definierten Indikatoren eindeutig zugeordnet werden können (z.B. deutet das Führen des Cursors auf eine Formel nicht nur auf FA, sondern auch auf DA und DB hin).

Mit Hilfe dieser Protokollkategorien kann der Problemlöseprozeß als Problemverhaltensgraph (Newell & Simon, 1972) dargestellt werden. Der Problemverhaltensgraph besteht aus den hypothetischen Wissenszuständen des Problemlösers (diese können auch die externe Bildschirmsituation beinhalten) sowie aus Operatoren, die jeweils von einem Wissenszustand zu einem anderen führen.

Abb. 5 zeigt einen Teil des Problemverhaltensgraphen einer Person. Wissenszustände sind durch Knoten, Operatoranwendungen durch Kanten repräsentiert. Das Zurückgehen zu einem früheren Wissenszustand wird durch Wiederholung des betreffenden Knotens und einen senkrechten Strich repräsentiert, der den früheren Knoten und seine Wiederholung miteinander verbindet. Das Erleben von Stocksituationen und die Heuristiken, die zu ihrer Überwindung eingesetzt werden, stellen ebenfalls Übergänge zwischen Wissenszuständen dar. Stocksituationen, Heuristiken und die Anwendung ihrer Ergebnisse sind als gestrichelte Kanten dargestellt.

---

[1] Neben den in Tabelle 1 angegebenen rechnerseitig lesbaren Indikatoren sind Verbalisationen weitere wichtige Indikatoren, welche bei der Entwicklung oder bei der Offline-Analyse eines lauffähigen Modells eine Rolle spielen können. Z.B. erwarten wir unterschiedliche Verbalisationsmuster bei den Operatoren DA vs. DB. Eine Äußerung wie "eine Disjunktion modelliere ich am besten durch entsprechend viele Transitionen" ist z.B. einem Designbaustein (Operator DB) zuzuordnen, aber nicht einer Designheuristik (Operator DA).

Protokollkat.	Beschreibung	mögliche Indikatoren
	Operatoren:	
FA	Auswahl einer Fortschaltbedingung mit möglichst wenig atomaren Formeln; möglichst viele als Stellen realisiert	Führen des Cursors auf eine entsprech. Formel Bewegen des Cursors in der Werkzeugleiste.
DA	Planung eines Netzfragments mit Designheuristiken (Abrufen eines Designheuristik-Schemas)	Wie bei FA. Außerdem: Nachfolgende Realisation von Stellen, Transitionen, Kanten in beliebiger Reihenfolge. Innerhalb dieser Aktionen wird nicht gezögert und nichts verschoben.
DB	Planung eines Netzfragments mit elementaren Designbausteinen	Wie bei DA, aber anschl. werden zuerst Stellen, dann Transitionen, dann Kanten realisiert. Zögern u. Verschieben möglich.
RE	Realisation von Stellen, Transitionen, Kanten	
HT	Bewertung durch Hypothesentest,	Hypothesentesten bzw.
AU	"Augenschein",	Zeigen mit dem Cursor nacheinander auf Atome einer Formel
SI	oder Netzsimulation	Netzsimulation
SP	Stellenpositionierung: Für jedes Atom in der Spezifikation wird eine Stelle positioniert.	Für Atome verschiedener Formeln werden nacheinander Stellen positioniert.
LÖ	Löschen von Stellen, Transitionen und Kanten	(Löschen)
	Stocksituationen:	
S1	Für Formel wurde noch keine Designheuristik erworben. Formel kann nicht in Netzfragment umgesetzt werden.	Langes Zeitintervall ohne Edieraktionen
S2	Entscheidung zwischen zwei alternativen Plänen kann nicht getroffen werden (Beispiel: "Soll für jedes Auftreten einer atomaren Formel in der Aufgabenspezifikation eine Stelle kreiert werden?")	Langes Zeitintervall ohne Edieraktionen; Bewegung des Cursors in Editor und Formelfenster; Anklicken verschiedener Items in Werkzeugleiste
S3	Formel nicht erfüllt oder Person ist sich bzgl. Erfülltsein unsicher	Systemrückmeldung; Zeigen mit Cursor nacheinander auf Atome einer Formel
	Heuristiken:	
H1	Die aktuell betrachtete Formel, die zu Stocksit. führte, wird zurückgestellt. Eine andere Formel wird ausgewählt, die mit bisherigen Designheur. bearbeitet werden kann.	Positionierung des Cursors auf eine andere Formel
H2	Das bisher erworbene Designwissen wird induktiv erweitert. (Generalisierung, z.B. von Abb. 2 auf Abb. 3 bzw. von Abb. 1 oben links auf Abb. 1 oben Mitte)	Langes Zeitintervall ohne Edieraktionen; anschließend erstmalige Realisation eines Netzfragments
H3	Bei zwei oder mehr alternativen Plänen wird eine Entscheidung für einen Plan getroffen.	Schnelle Realisation eines Netzfragments nach Hin- u. Herklicken in Werkz.-leiste
H4	Die Teile der aktuell betrachteten Formel, die zur Stocksit. geführt haben, werden ignoriert.	Hypothese für Formel wird getestet, ohne daß alle ihre Atome als Stellen realis. sind.
H5	Netzsimulation	(Netzsimulation)
H6	Anforderung eines Ergänzungs- / Korrekturvorschlags	(Auswahl des Menü-Items)

Tabelle 1: Abstrakte Protokollkategorien (entsprechend den Modellphasen "abwägen", "planen" usw. sowie Stocksituationen und Heuristiken) und ihre möglichen Indikatoren

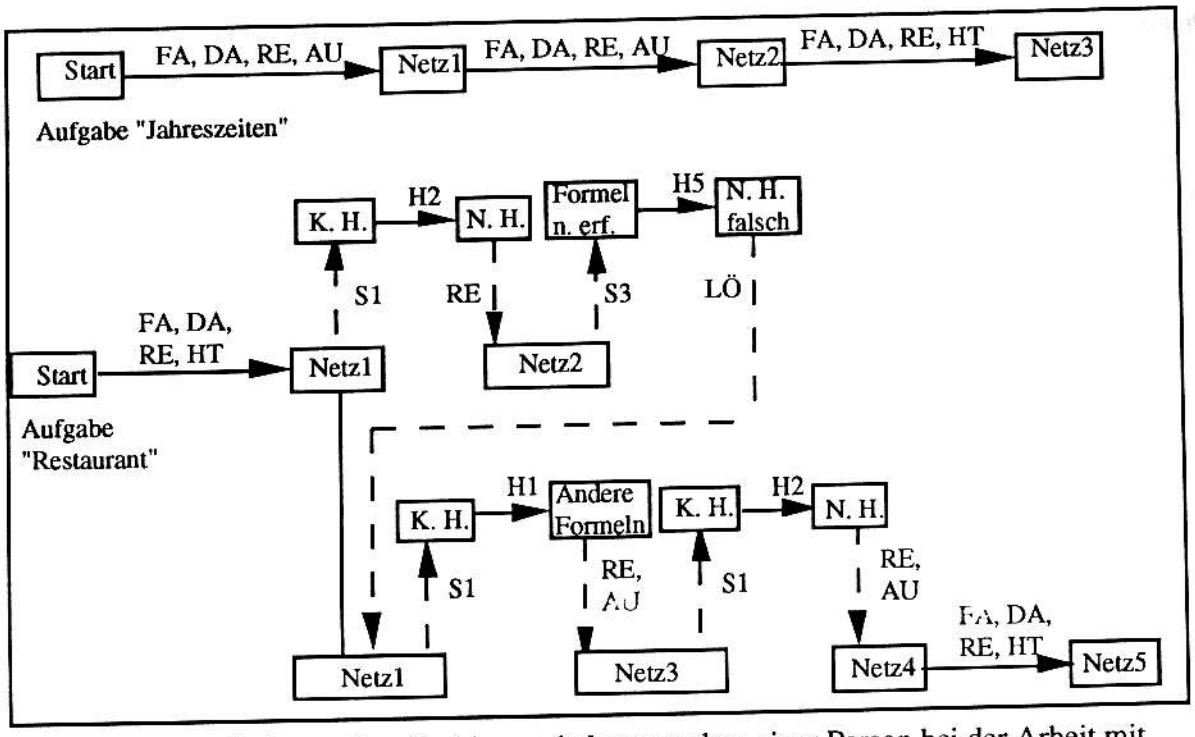

Abb. 5: Ausschnitt aus dem Problemverhaltensgraphen einer Person bei der Arbeit mit PETRI-HELP

Der obere Teil von Abb. 5 zeigt die Bearbeitung der Aufgabe "Jahreszeiten". Die Person konstruiert das Netz durch mehrmalige Formelauswahl, Anwendung einer Designheuristik, ihrer Realisation und der Bewertung des jeweiligen Ergebnisses, wobei ihr Modellierungswissen der Designheuristik in Abb. 1 oben links (bzw. Abb. 2) entspricht. "Netz1", "Netz2", "Netz3" bezeichnen verschiedene Entwurfsstadien des Netzes.

Bei der zweiten "Aufgabe, "Restaurant", wird zunächst ebenfalls diese Operatorfolge angewendet (Realisation der Formel "$\Box$ Ws $\rightarrow \Diamond$ WbA"[2]). Als nächstes wählt die Person die Formel "$\Box$ WbA $\rightarrow \Diamond$ (Ws $\wedge$ K)", und sie erlebt eine Stocksituation (S1), weil das bisher erworbene Designwissen (Abb. 1 oben links) zur Realisation dieser Formel nicht ausreicht (K.H. = keine Designheuristik). Durch die Heuristik H2 (induktive Wissenserweiterung) entwickelt die Person die neue (falsche) Designheuristk, die Formel "$\Box$ WbA $\rightarrow \Diamond$ (Ws $\wedge$ K)" durch das in Abb. 1 oben rechts dargestellte Netzfragment zu realisieren. (N.H. Dieser hypothetische Prozeß wird unten an einem anderen, richtigen Beispiel erläutert.) Die neue Designheuristik wird ausgeführt (RE) und das Netz entsprechend weiterentwickelt. Nun tritt jedoch eine Stocksituation S3 auf, die Person ist sich über die Richtigkeit der gerade ausgeführten Edierschritte im Unklaren. Zur Beantwortung dieser Frage wählt sie die Netzsimulation (H5) mit dem Ergebnis, daß sie die gerade ausgeführten Schritte für falsch hält. Daraufhin macht sie sie rückgängig (LÖ). Damit ist sie zu dem früheren Entwurfszustand "Netz1" zurückgekehrt. Sie steht wieder vor dem Problem, wie die Formel "$\Box$ WbA $\rightarrow \Diamond$ (Ws $\wedge$ K)" realisiert werden kann (S1). Sie entschließt sich, zunächst andere Formeln zu bearbeiten, die mit

---

[2] Bedeutungen der Abkürzungen: "Ws": "Wirt schläft"; "WbA": "Wirt ist bereit zur Auftragsannahme"; "K": "Küche hat Auftrag erhalten"; "Z": Essen wird zubereitet"; "E": "Essen ist fertig".

ihrem bisherigen Modellierungswissen (Abb. 1 oben links) bearbeitet werden können (Heuristik H1). Dies sind die Formeln "□ K → ◊ Z" und "□ Z → ◊ E". Sie werden ausgeführt, was in dieser Situation natürlich dazu führt, daß die Stocksituation S1, bezogen auf die Formel "□ WbA → ◊ (Ws ∧ K)", wiederum auftritt. Als Reaktion auf die Stocksituation findet nun ein Problemlöseprozeß statt mit dem Ergebnis, daß jedem Atom in der Conclusio der Formel eine Stelle zugeordnet wird. Die Formel wird also durch eine Transition mit dem Vorbereich {WbA} und dem Nachbereich {Ws, K} modelliert. Dieses Ergebnis des Problemlöseprozesses stellt eine induktive Erweiterung des bisherigen Modellierungswissens dar (Heuristik H2), die Abb. 1 oben Mitte bzw. dem aus zwei Generalisierungen bestehenden Übergang von Abb. 2 nach Abb. 3 entspricht. Es ist also neues Wissen erworben worden, und die Bearbeitung der Aufgabe kann erfolgreich beendet werden.
Der Problemverhaltensgraph kann als Ausgangspunkt dafür genutzt werden, um ein lauffähiges Modell des Problemlöse- und Wissenserwerbsprozesses sowie des Modellierungswissens (Designbausteine, -heuristiken) einer Person zu entwickeln (Newell & Simon, 1972). Zusammen mit rechnerseitig erfaßbaren Indikatoren dient ein solches Modell der automatischen online-Diagnose von Wissen, der Diagnose von Stocksituationen und der Bereitstellung wissensstandsangepaßter Hilfen. Darüber hinaus wird genau ersichtlich, in welchen Situationen und mit welchem Erfolg die Hilfen (Hypothesentesten, Ergänzungsvorschläge) genutzt werden. Abb. 6 zeigt einen Ansatz eines solchen Modells, das die o.g. Protokollkategorien in einem Und-Oder-Graphen zusammenfaßt.

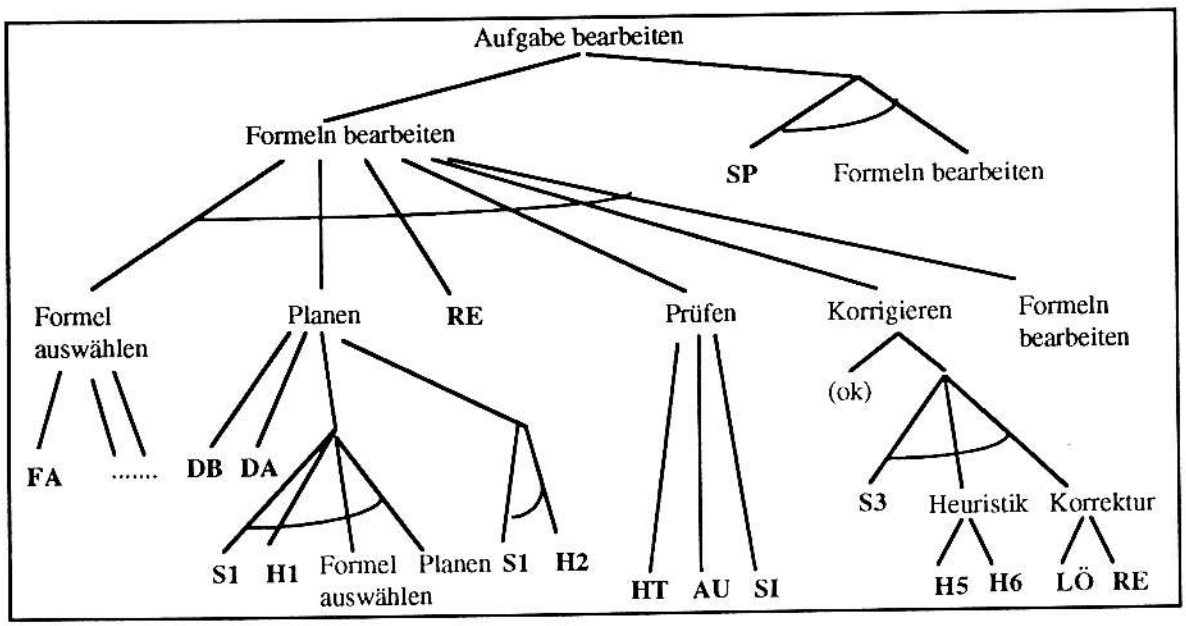

Abb. 6: Zusammenfassung von Protokollkategorien aus Tabelle 1 in einem Und-Oder-Graphen

2.2 Zur Akzeptanz der Hilfen

Neben den oben dargestellten Einzeluntersuchungen wurde PETRI-HELP im Rahmen einer Übungsgruppe mit 18 Teilnehmern eingesetzt. Die Teilnehmer bearbeiteten die zehn Modellierungsaufgaben in insgesamt neun Kleingruppen mit der Instruktion, Kritik und sonstige Kommentare in einem hierfür konstruierten Formblatt einzutragen. Zur Untersuchung der Akzeptanz der in PETRI-HELP bereitgestellten Hilfen wurden diese schriftlichen Rückmeldungen ausgewertet.

Insgesamt wurden die Hilfemöglichkeiten in PETRI-HELP als positiv beurteilt, wobei dem Hypothesentesten gegenüber der Anforderung von Ergänzungsvorschlägen der Vorzug gegeben wurde. Bezüglich des Hypothesentestens wurde von sieben Kleingruppen das Fehlen von Begründungen für unerfüllte Formeln kritisiert, und daß die Rückmeldung unerfüllter Formeln in manchen Fällen ohne Erklärung nicht weiterhelfe. Diese Kritik stimmt mit dem Verhalten der beiden Personen, deren Vorgehen im vorhergehenden Abschnitt beschrieben wurde, überein: Nach einer Stocksituation S3 (Formel nicht erfüllt bzw. unklar) wurde in 80% der Fälle das Netz simuliert (H5) oder ein Ergänzungs- / Korrekturvorschlag angefordert (H6). Die Information "Formel nicht erfüllt" reichte zur Überwindung der aktuellen Stocksituation in den meisten Fällen also nicht aus.
In Bezug auf die Ergänzungs- / Korrekturvorschläge wurde von ebenfalls sieben Gruppen kritisiert, daß in manchen Fällen zuviel Information gegeben werde, die zudem ebenfalls nicht erklärt sei[3]. In vier dieser Fälle wurde der Wunsch nach abstrakterer, nicht auf der Ebene von Stellen, Transitionen und Kanten angesiedelter Information geäußert (vgl. auch Schröder, Möbus, Pitschke, 1993). Zwei weitere Kritikpunkte betrafen die Verständlichkeit der Ergänzungs- und Korrekturvorschläge. Darüber hinaus berichteten alle Gruppen, daß sie die Ergänzungs- / Korrekturvorschläge nur dann angefordert hätten, wenn sie mit dem Hypothesentesten nicht weitergekommen seien. Dies stimmt auch mit dem Vorgehen der o.g. Einzelpersonen überein: Ergänzungs- / Korrekturvorschläge (H6) wurden stets nach einer Stocksituation S3 (Formel nicht erfüllt bzw. Erfüllung unklar) angefordert.
Diese Ergebnisse sind noch vorläufig und durch weitere Untersuchungen zu stützen. Zusammenfassend legen sie folgendes nahe:

- Für Rückmeldungen unerfüllter Hypothesen sollten Erklärungen bereitgestellt werden.
- Die Ergänzungs- / Korrekturvorschläge sollten
  - durch Erklärungen ergänzt werden
  - nicht zuviel Information auf einmal vorgeben
  - auch abstraktere Hinweise (statt der Vorgabe fertiger Lösungsfragmente) enthalten.

3. Systemerweiterungen

Die oben genannten Kritikpunkte an den Hilfen von PETRI-HELP führen zu verschiedenen Systemerweiterungen, an denen derzeit gearbeitet wird.

Erklärungen. Die Klassifikation von Formeln als erfüllt bzw. nicht erfüllt beruht auf dem Fallgraphen, in den der Netzentwurf des Benutzers überführt wird, und in dem die temporallogischen Formeln der Aufgabenspezifikation verifiziert werden. Die vorgeschlagenen Ergänzungen und Korrekturen zielen auf die Erweiterung der Menge erfüllter Formeln. Es liegt daher nahe, beide Arten

---

[3] Zu verschiedenen Varianten von als Hilfe gedachten Informationen haben wir Hypothesen formuliert (Möbus, Schröder & Thole, 1992). Bei der Vorgabe von zuviel Information, die zudem nicht erklärt ist, erwarteten wir zwei Effekte: Zum einen muß der Problemlöser die aktuell relevante Information aus dem Informationsangebot herausfiltern, was er als lästig empfinden sollte. Zum anderen ist ein Polarisierungseffekt zu erwarten: Die angebotene Information wird, da nicht erklärt, entweder passiv übernommen, oder es kommt zu Selbsterklärungseffekten. In Übereinstimmung hiermit äußerten einige Personen auch, daß sie die dargebotene Information nur passiv übernehmen würden. Der Filterungseffekt hingegen sollte mit zunehmender Benutzung des Systems geringer werden, da das System aus den Interaktionen mit seinen Benutzern lernt und zunehmend kleinere Informationsmengen anbieten kann.

von Hilfen in PETRI-HELP durch den Fallgraphen bzw. durch Markierungszustände des Netzentwurfs zu erklären. Beobachtungen der Erklärungsmuster von Personen, die Novizen bei der Arbeit mit PETRI-HELP betreuten, zeigten ebenfalls, daß nicht erfüllte Formeln sowie Ergänzungs- / Korrekturvorschläge durch Demonstration von Abfolgen von Markierungszuständen erklärt werden. Abb. 7 zeigt, wie die Rückmeldung einer nicht erfüllten Formel erklärt werden kann. Wir nehmen an, der Benutzer habe zu dem in Abb. 7 links dargestellten Netzentwurf eine Hypothese getestet, die die Formel "$\Box$ Wrs $\rightarrow \Diamond$ (Ws $\wedge$ K)" enthält. Das System meldet zurück, daß diese Formel nicht erfüllt sei. Die Erklärung dieser Rückmeldung besteht nun in der Demonstration, daß hieraus der in Abb. 7 rechts dargestellte Markierungszustand entstehen kann. (Abb. 7 rechts kann aus Abb. 7 links durch Netzsimulation hergestellt werden.) In diesem Zustand kann die Transition mit dem Vorbereich {Wrs} nicht mehr schalten, also kann die Stelle "K" trotz einer Marke auf "Wrs" keine Marke mehr bekommen: "$\Box$ Wrs $\rightarrow \Diamond$ (Ws $\wedge$ K)" ist nicht erfüllt.

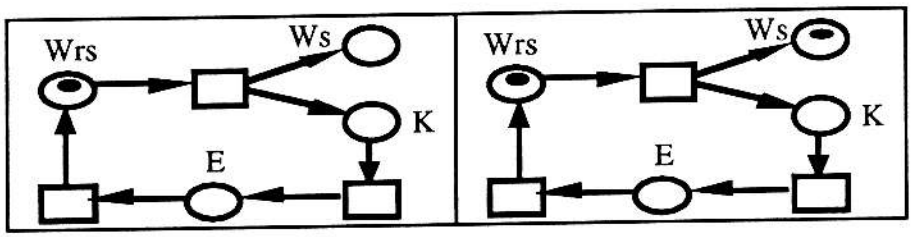

Abb. 7: Erklärung nicht erfüllter Formeln über Markierungszustände

Ergänzungsvorschläge können analog erklärt werden. Schlägt das System z.B. als Ergänzung für den Entwurf in Abb 7 links die fett gedruckten Teile in Abb. 8 vor, so besteht die Erklärung dieses Vorschlags in der Information, daß die Formel "$\Box$ Wrs $\rightarrow \Diamond$ (Ws $\wedge$ K)" nun erfüllt ist, da der dargestellte Markierungszustand, in der die Transition mit dem Vorbereich {Wrs} schalten kann, immer wieder erreicht werden kann. (Auch dies kann bei Bedarf durch Simulation gezeigt werden.)

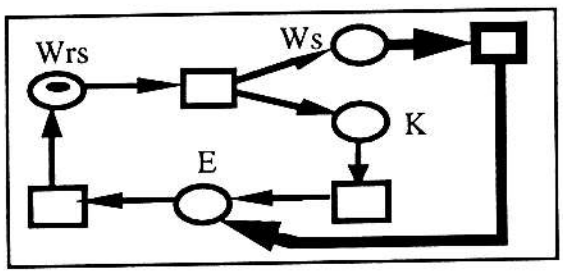

Abb. 8: Ergänzungsvorschlag

Vorgabe abstrakterer Hinweise. Gegenwärtig wird der von Olderog (1991) entwickelte Transformationsansatz bzw. seine in Wedig (1993) erfolgte Weiterentwicklung in PETRI-HELP integriert. Hiermit soll es den Problemlösern ermöglicht werden, Netze ausgehend von Spezifikationen regelgeleitet zu entwickeln, Hypothesen über (Teil-) Spezifikationen zu formulieren und Rückmeldungen, Korrektur- und Ergänzungsvorschläge auf der Ebene von Teilspezifikationen zu erhalten. Damit soll es möglich werden, Hilfen als abstrakte Hinweise statt als Vorgabe fertiger Lösungsfragmente zu geben und die Problemlöseprozesse bereits in sehr frühen Lösungsstadien zu unterstützen (Schröder, Möbus & Pitschke, 1993).

Unterstützung der Spezifikationsentwicklung. Das Modellierungswissen, das mit PETRI-HELP erworben werden kann, unterliegt der folgenden Einschränkung: Der Benutzer lernt, Netze *zu vorgegebenen Spezifikationen* zu entwickeln. Dies ist einerseits notwendig, damit das System Lösungsentwürfe von Benutzern analysieren und Rückmeldungen geben kann. Andererseits wird dadurch die "freie" Modellierung, der Entwurf von Netzen zu nicht oder nur umgangssprachlich vorliegenden Spezifikationen, noch nicht unterstützt. PETRI-HELP soll dahingehend weiterentwickelt werden, daß auch für die "freie", nicht durch eine fertig vorgegebene Spezifikation geleitete Modellierung Hilfen angeboten werden. Dieser wichtige Aspekt von Modellierungsexpertise wird in PETRI-HELP gegenwärtig integriert. Dabei soll das System den Benutzer durch einen Dialog in der Entwicklung einer temporallogischen Spezifikation unterstützen (vgl. auch Spohrer, 1992, für die Spezifikation von Programmieraufgaben). Hierzu kann der von Collins (1977) entwickelte Ansatz zur Systematisierung eines Sokratischen Dialogs genutzt werden (Tabelle 2). Der Benutzer wird zunächst nach den Akteuren befragt, die in dem zu modellierenden System beteiligt sein sollen (Schritt 1). Dann werden für jeden Akteur die möglichen Zustände erfragt (Schritt 2). Über die verschiedenen Zustände jedes Akteurs können Ausschlußbedingungen formuliert werden (Schritt 3a). Weitere Ausschlußbedingungen müssen erfragt werden (Schritt 3b). Im Schritt 4 folgt die Frage nach Vorbedingungen für jeden Zustand ("Ask for prior factors" bei Collins). Im Schritt 5 werden mögliche konjunktive und disjunktive Verknüpfungen zwischen den bis hierher entwickelten Formeln gebildet bzw. erfragt. Schritt 6 schließlich erfragt die Anfangsbedingungen.

Literatur

Collins, A., Processes in Acquiring Knowledge, in R.C. Anderson, R.J. Spiro, W.E. Montague (eds), Schooling and the Acquisition of Knowledge, Hillsdale: Erlbaum, 1977, 339-374

Goldstein, I.P., The Genetic Graph: A Representation for the Evolution of Procedural Knowledge, in D. Sleeman, J.S. Brown (eds), Intelligent Tutoring Systems, London: Academic Press, 1982, 51-77

Möbus, C., Pitschke, K., Schröder, O., Göhler, H., Gewinnung von Planregeln und Designheuristiken für PETRI-HELP, in V. Claus, U. Lichtblau (Hg), 3. Kolloquium der Arbeitsgruppe Informatik-Systeme, Bericht AIS-5, Universität Oldenburg, FB Informatik, April 1992

Möbus, C., Pitschke, K., Schröder, O., Folckers, J., Göhler, H., Erwerb von Modellierungswissen durch Hypothesentesten in PETRI-HELP, in M. Sonnenschein, U. Lichtblau (Hg), 5. Kolloquium der Arbeitsgruppe Informatik-Systeme, Bericht AIS-10, Universität Oldenburg, FB Informatik, Mai 1993

Möbus, C., Schröder, O., Thole, H.J., A Model of the Acquisition and Improvement of Domain Knowledge for Functional Programming, Journal of Artificial Intelligence in Education, 1992, 3(4), 449-476

Newell, A., Simon, H.A., Human Problem Solving, Englewood Cliffs: Prentice Hall, 1972

Olderog, E.R., Nets, Terms, and Formulas, New York: Cambridge, 1991

Pitschke, K., Zielidentifikation und Planen in Intelligenten Tutoriellen Systemen, in: A. Kobsa, W. Pohl (Hg), Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen, Berlin / Konstanz, Sept. 1993

Pitschke, K., Möbus, C., Improvement of Help Information by Accumulative Learning, in C. Möbus (Hg), 6. Arbeitstreffen der Fachgruppe Intelligente Tutorielle Lernsysteme der Gesellschaft für Informatik, Universität Oldenburg am 8./9.6.93, Interne Berichte des Fachbereichs Informatik, 1993, 67-72

Reisig, W., Petrinetze: Eine Einführung, Berlin: Springer, 1986

Reisig, W., Petri Nets - A Primer, Berlin: Springer, 1992

Schröder, O., Möbus, C., Pitschke, K., Design of Help for Viewpoint Centered Planning in Petri Nets, in P. Brna, S. Ohlsson, H. Pain (eds), Proceedings of the 5th Int. Conf. on Artificial Intelligence, and Education (AI-ED 93), Association of the Advancement of Computing in Education (AACE), 1993, 370-377

Spohrer, J.C., MARCEL - Simulating the Novice Programmer, Norwood: Ablex, 1992

Wedig, A., Kalküle zur Entwicklung von Petri-Netzen aus Spezifikationen, Universität Oldenburg, Fachbereich Informatik: Diplomarbeit, 1993

Schritt	Aktionen von PETRI-HELP	Aktionen des Benutzers
1. Akteure	Frage nach den Akteuren, die in dem zu spezifizierenden System beteiligt sein sollen	Angabe der Akteure: A_1 ... A_m
2. Zustände	Frage nach den möglichen Zuständen, in denen sich jeder Akteur befinden kann	Angabe der Zustände für jeden Akteur: A_1Z_1, ..., A_1Z_n, ..., A_mZ_1, ..., A_mZ_p
3. Ausschluß-bedingungen	a) Da für jeden Akteur gilt, daß sich seine Zustände ausschließen, können Ausschluß-bedingungen gebildet werden: $\Box$ $(\neg (A_1Z_1 \wedge A_1Z_2))$, ..., $\Box$ $(\neg (A_1Z_{n-1} \wedge A_1Z_n))$, ..., $\Box$ $(\neg (A_mZ_1 \wedge A_mZ_2))$, ..., $\Box$ $(\neg (A_mZ_{p-1} \wedge A_mZ_p))$	
	b) Frage für jeden Zustand: Welche Zustände anderer Akteure schließen sich mit ihm aus? Umsetzung der Angaben des Benutzers in Formeln: ..., $\Box$ $(\neg (A_iZ_k \wedge A_jZ_l))$, ...	Für jeden Zustand: Angabe der sich ausschließenden Zustände
4. Fortschalt-bedingungen	Frage nach den Vorbedingungen jedes Zustands Umsetzung der Angaben des Benutzers in Formeln: $\Box$ ($\bigwedge$ A_iZ_j vom Ben. angegeben $\rightarrow \Diamond A_1Z_1$), ..., $\Box$ ($\bigwedge$ A_iZ_j vom Ben. angegeben $\rightarrow \Diamond A_1Z_n$), ..., $\Box$ ($\bigwedge$ A_iZ_j vom Ben. angegeben $\rightarrow \Diamond A_mZ_1$), ..., $\Box$ ($\bigwedge$ A_iZ_j vom Ben. angegeben $\rightarrow \Diamond A_mZ_p$)	Für jeden Zustand: Angabe der Vorbedingungen
5. Konjunktive und disjunktive Verknüpfungen	a) Jedes Formelpaar, $\Box$ $P_1 \rightarrow \Diamond C$, $\Box$ $P_2 \rightarrow \Diamond C$ (versch. Prämissen, gleiche Conclusio) wird ersetzt durch $\Box$ $P_1 \vee P_2 \rightarrow \Diamond C$	
	b) Für jedes Formelpaar $\Box$ $P \rightarrow \Diamond C_1$, $\Box$ $P \rightarrow \Diamond C_2$ (gleiche Prämisse, versch. Conclusio): Frage: Liegen C_1 und C_2 gleichzeitig vor?	Antwort (ja / nein)
	Wenn ja: $\Box$ $P \rightarrow \Diamond C_1$, $\Box$ $P \rightarrow \Diamond C_2$ werden ersetzt durch $\Box$ $P \rightarrow \Diamond (C_1 \wedge C_2)$	
	c) Jedes verbliebene Formelpaar $\Box$ $P \rightarrow \Diamond C_1$, $\Box$ $P \rightarrow \Diamond C_2$ wird ersetzt durch $\Box$ $P \rightarrow \Diamond (C_1 \vee C_2)$	
6. Anfangszustände	Frage nach den Anfagnszuständen (1 Anfangszustand pro Akteur)	Angabe der Anfangszustände: A_1Z_i, ... A_mZ_j

Tabelle 2: Unterstützung der Spezifikationsentwicklung