

# PETRI-HELP:

## Expertiserwerb, 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., 1993a, b). 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.

### 1. Implementationsstand

PETRI-HELP besteht aus folgenden Komponenten (vgl. ausführlich Möbus et al., 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.

- 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.

## 2. Expertiseerwerb und Hilfeakzeptanz

PETRI-HELP soll Novizen beim Erwerb von Modellierungsexpertise mit Hilfen unterstützen. In empirischen Einzelfallstudien wurde daher untersucht,

- worin das von den Personen erworbene Modellierungswissen besteht
- wie dieses Wissen erworben wird
- wie die von PETRI-HELP bereitgestellten Hilfen - Hypothesentesten mit Rückmeldung erfüllter / nicht erfüllter Formeln sowie Korrektur- / Ergänzungsvorschläge - von Personen genutzt und akzeptiert werden.

Dabei unterliegt das Modellierungswissen, das mit PETRI-HELP erworben werden kann, von vornherein einer Einschränkung, da PETRI-HELP mit vorgegebenen Spezifikationen arbeitet: Der Benutzer lernt, Netze *zu vorgegebenen Spezifikationen* zu entwickeln. Dies ist einerseits notwendig, um Lösungsentwürfe von Benutzern analysieren und Rückmeldungen geben zu können. Andererseits wird dadurch die "freie" Modellierung, der Entwurf von Netzen zu nicht oder nur umgangssprachlich vorliegenden Spezifikationen, noch nicht unterstützt. Dieser Punkt wird im Abschnitt 3 "Systemerweiterungen" aufgegriffen.

### 2.1 Zum Erwerb von Modellierungsexpertise

Die empirische Untersuchung der Frage, welches Wissen die Personen bei der Arbeit mit PETRI-HELP erwerben und wie dieses Wissen erworben wird, ist vor allem vor dem Hintergrund des Ziels einer *automatisierten Wissensdiagnose* von Bedeutung. Ein sich online entwickelndes Lernermodell ist Voraussetzung für individualisierte, an das aktuelle Wissen des Benutzers angepaßte Hilfen (Möbus et al., 1993; Pitschke, 1993). Zum anderen können Fragen nach der Akzeptanz und Effektivität von Hilfen sowie Fragen nach den möglichen Ursachen für intra- und individuelle Unterschiede in der Akzeptanz nur dann sinnvoll beantwortet werden, wenn ein solches Modell vorliegt. Möbus, Schröder & Thole (1992) haben gezeigt, daß ein Modell des aktuellen Wissensstandes des Benutzers zu detaillierten Vorhersagen hinsichtlich Akzeptanz und Wirksamkeit verschiedener Hilfeinformationen führt.

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.

Zwei Personen arbeiteten in Einzelsitzungen mit dem jetzigen Implementationsstand von PETRI-HELP. Ihre Handlungen und Verbalisationen wurden mit der Videokamera aufgenommen.<sup>1</sup> 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*)

Operator **DA** (Designheuristik-Schema abrufen): Planung der Realisierung der Fortschaltbedingung gemäß des verfügbaren Modellierungswissens (s.u.) (*planen*)

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**) oder durch Simulation des Netzes (**SI**) (*bewerten*)

*Zum Modellierungswissen.* Im Operator "DA" ist das eigentliche domänenspezifische Modellierungswissen enthalten. Dieses hypothetische Wissen besteht aus Schemata, die jeweils einer Formel ein Netzfragment zuordnen. Diese Schemata bezeichnen wir als *Designheuristiken*. Abb. 1 zeigt Beispiele.

Die Designheuristik in Abb. 1 oben links besagt z.B., daß Formeln, die dem Schema " $[\ ] X \rightarrow \diamond Y$ " entsprechen, durch eine Transition mit dem Vorbereich  $\{X\}$  und dem Nachbereich  $\{Y\}$  realisiert werden. Die Designheuristiken wurden empirisch durch Untersuchung der Differenz der Hypothesen in jeweils zwei aufeinanderfolgenden Bewertungsphasen gewonnen: Z.B. hat die Person den Netzzustand  $N_j$  realisiert und die Hypothese  $H_j$  formuliert. Anschließend erweitert sie den Netzzustand durch Edieren weiterer Stellen, Transitionen und Kanten zum Netzzustand  $N_{j+1}$  und formuliert die Hypothese  $H_{j+1}$ . Nun kann die Formeldifferenz  $H_{j+1} - H_j$  mit der Netzdifferenz  $N_{j+1} - N_j$  in Beziehung gesetzt werden. Diese Zuordnung lieferte in etwa 70% der Fälle unsere Designheuristiken. Empirische Einzelfalluntersuchungen mit einem

---

<sup>1</sup> Die Interaktionen einer Reihe weiterer Personen mit Vorversionen des Systems wurden ebenfalls mit der Videokamera aufgenommen und begleitend zur Systementwicklung analysiert (Möbus et al., 1992; 1993).

frühen "Papier-und-Bleistift"-Entwicklungsstadium von PETRI-HELP ergaben ähnliche Ergebnisse (Möbus et al., 1992).

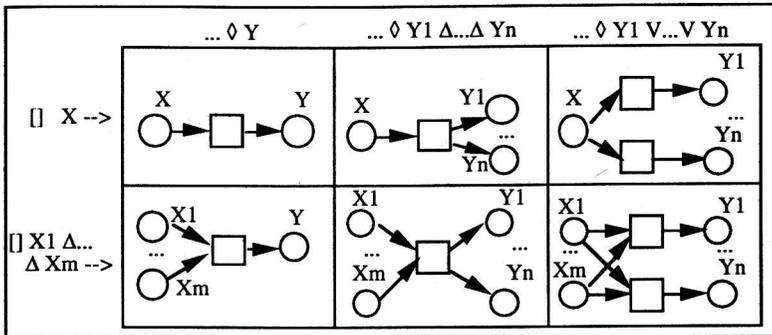


Abb. 1: Einige empirische Designheuristiken für PETRI-HELP

Als rechnerseitig erfassbaren Indikator für eine Designheuristik betrachten wir also die Formeldifferenz zwischen zwei aufeinanderfolgenden Hypothesentests ( $H_i, H_{i+1}$ ) und das, was zwischen diesen beiden Hypothesentests dem Netzentwurf hinzugefügt wurde. Ein weiterer Indikator sind Verschiebungen. 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 Verschiebung 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 elementarerem 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.

*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. Für jede Protokollkategorie sind mögliche empirische Indikatoren angegeben. Dabei ist allerdings zu beachten, daß nicht alle Protokollkategorien 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 hin).

Protokollkat.	Beschreibung	mögliche Indikatoren
<i>Operatoren:</i>		
FA	Auswahl einer Fortschaltbedingung mit möglichst wenig atomaren Formeln; möglichst viele als Stellen realisiert	Führen des Cursors auf eine entsprechende Formel
DA	Planung eines Netzfragments mit Designheuristiken (Abrufen eines Designheuristik-Schemas)	Bewegen des Cursors in der Werkzeugeiste
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 (meist am Anfang einer Aufgabenbearb.): Für jedes Atom in der Spezifikation wird eine Stelle positioniert.	Für Atome versch. Formeln werden nacheinander Stellen pos.
LÖ	Löschen von Stellen, Transitionen und Kanten	(Löschen)
<i>Stocksituationen:</i>		
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, Formelfenster; Anklicken verschiedener Items in Werkzeugeiste
S3	Formel nicht erfüllt oder Person ist sich bzgl. Erfülltsein unsicher	Systemrückmeldung Zeigen mit Cursor nacheinander auf Atome einer Formel
<i>Heuristiken:</i>		
H1	Die aktuell betrachtete Formel, die zu Stocksit. geführt hat, wird zurückgestellt. Dafür wird eine andere Formel ausgewählt, die mit bisherigen Designheuristiken bearbeitet werden kann.	Positionierung des Cursors auf eine andere Formel
H2	Das bisher erworbene Designwissen wird induktiv erweitert. (Z.B. Generalisierung von Abb. 1 oben links auf Abb. 1 oben Mitte)	Langes Zeitintervall ohne Edieraktionen; anschl. erstmalige Realisation eines Netzfragments
H3	Bei zwei oder mehr alternativen Plänen wird eine Entscheidung für einen Plan getroffen.	Schnelle Realisation eines Netzfragments nach vorherigem Hin- und Herklicken auf 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 realisiert sind.
H5	Netzsimulation	(Netzsimulation)
H6	Anforderung eines Ergänzungs- / Korrekturvorschlags	(Auswahl des Menü-Items)

Tabelle 1: Protokollkategorien und mögliche Indikatoren

Mit Hilfe dieser Protokollkategorien kann der Problemlöseprozeß der Personen 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. 2 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.

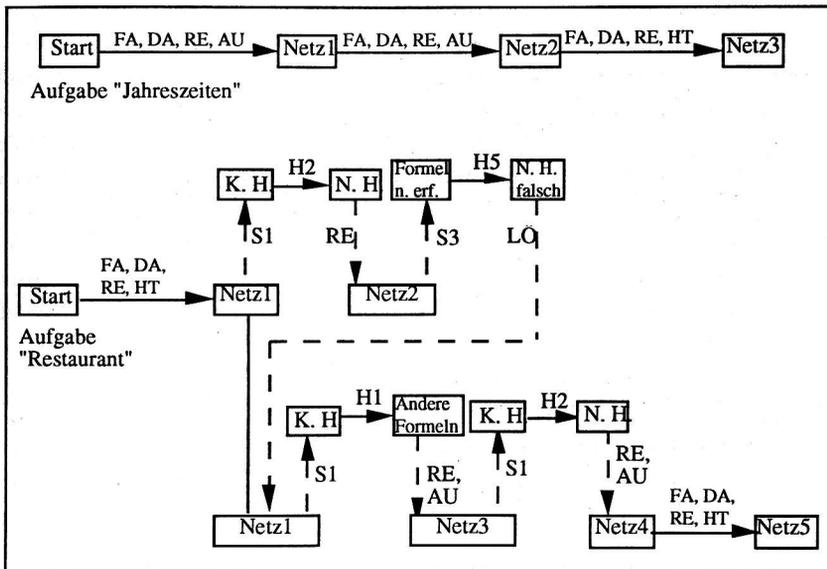


Abb. 2: Ausschnitt aus dem Problemverhaltensgraphen einer Person bei der Arbeit mit PETRI-HELP

Der obere Teil von Abb. 2 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 entspricht. "Netz1", "Netz2", "Netz3" bezeichnen verschiedene Entwurfsstadien des Netzes.

Bei der zweiten "Aufgabe, "Restaurant", wird zunächst ebenfalls diese Operatorfolge angewendet (Realisation der Formel " $\square Ws \rightarrow \diamond WbA$ "<sup>2</sup>). Als nächstes wählt die Person die Formel " $\square WbA \rightarrow \diamond (Ws \wedge K)$ ", 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 Designheuristik (N.H.), die Formel " $\square WbA \rightarrow \diamond (Ws \wedge K)$ " durch das in Abb. 1 oben rechts dargestellte Netzfragment zu realisieren. 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 " $\square WbA \rightarrow \diamond (Ws \wedge K)$ " realisiert werden kann (S1). Sie entschließt sich, zunächst andere Formeln zu bearbeiten, die mit ihrem bisherigen Modellierungswissen (Abb. 1 oben links) bearbeitet werden können (Heuristik H1). Dies sind die Formeln " $\square K \rightarrow \diamond Z$ " und " $\square Z \rightarrow \diamond E$ ". Sie werden ausgeführt, was in dieser Situation natürlich dazu führt, daß die Stocksituation S1, bezogen auf die Formel " $\square WbA \rightarrow \diamond (Ws \wedge K)$ ", 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 entspricht. Durch Problemlösen ist eine neue Designheuristik erworben worden. Mit dem so erweiterten Wissen kann die Bearbeitung der Aufgabe erfolgreich beendet werden.

Der Problemverhaltensgraph kann als Ausgangspunkt dafür genutzt werden, um ein lauffähiges Modell des Problemlöse- und Wissenserwerbsprozesses sowie des Modellierungswissens (Designheuristiken) einer Person zu entwickeln (Newell & Simon, 1972). Zusammen mit rechnerseitig erfaßbaren Indikatoren dient ein solches Modell für die automatische online-Wissensdiagnose, für die Diagnose von Stocksituationen und damit zur Bereitstellung wissensstandsangepaßter Hilfen. Darüber hinaus wird genau ersichtlich, in welchen Situationen und mit welchem Erfolg die Hilfen (Hypothesentesten, Ergänzungsvorschläge) genutzt werden. Abb. 3 zeigt einen Ansatz für ein solches Modell, das den größten Teil der o.g. Protokollkategorien in einem Und-Oder-Graphen zusammenfaßt.

<sup>2</sup> 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".

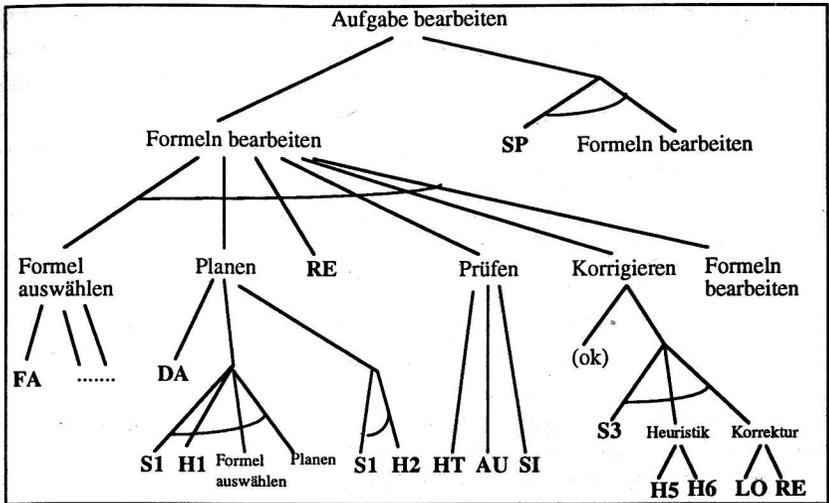


Abb. 3: Zusammenfassung von Protokollkategorien aus Tabelle 1 in einem Und-Oder-Graphen

## 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 bloße Rückmeldung unerfüllter Formeln in manchen Fällen nicht weiterhelfe, sondern daß diese Rückmeldung erklärt werden müsse. Diese Kritik stimmt mit dem Verhalten der beiden Einzelpersonen, deren Vorgehen im vorhergehenden Abschnitt beschrieben wurde, überein: Nach einer Stocksituation S3 (Formel nicht erfüllt bzw. Erfüllung unklar) wurde in 80% der Fälle das Netz simuliert (H5) oder ein Ergänzungs- / Korrekturvorschlag angefordert (H6), d.h. die Information "Formel nicht erfüllt" war zur Überwindung der aktuellen Stocksituation in den meisten Fällen nicht hinreichend.

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<sup>3</sup>. In vier dieser Fälle wurde der Wunsch nach abstrakterer, nicht auf der Ebene von Stellen, Transitionen und Kanten angesiedelter Information geäußert. 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 die Rückmeldungen unerfüllter Hypothesen sollten Erklärungen bereitgestellt werden.
- Die Ergänzungs- / Korrekturvorschläge sollten
  - durch Erklärungen ergänzt werden
  - nicht zuviel Information auf einmal vorgeben
  - durch abstraktere Hinweise (statt der Vorgabe fertiger Lösungsfragmente) ergänzt werden.

### 3. Systemerweiterungen

Die oben genannten Kritikpunkte an den Hilfen von PETRI-HELP führen zu verschiedenen Systemerweiterungen, an denen derzeit gearbeitet wird. Insbesondere wird der von Olderog (1991) entwickelte Transformationsansatz bzw. seine in Wedig (1993) erfolgte Weiterentwicklung gegenwärtig in PETRI-HELP implementiert. Hiermit soll es den Problemlösern ermöglicht werden, Netze ausgehend von Spezifikationen regelgeleitet zu entwickeln, Hypothesen über (Teil-) Spezifikationen zu formulieren und Rückmeldungen, Korrektur- und Ergänzungs-vorschläge auf der Ebene von Teilspezifikationen zu erhalten. Damit soll es möglich werden, Hilfen als abstrakte Hinweise statt als Vorgabe fertiger Lösungsfragmente zu geben und unterstützende Information bereits in sehr frühen Lösungsstadien bereitzustellen (Schröder, Möbus & Pitschke, 1993).

Eine weitere wichtige, bereits in der Einleitung angesprochene Erweiterung betrifft die Unterstützung der "freien", nicht durch eine fertig vorgegebene Spezifikation geleiteten Modellierung. Dieser wichtige Aspekt von Modellierungsexpertise wird in PETRI-HELP gegenwärtig integriert. Dabei soll das System den Benutzer durch einen Dialog in der

---

<sup>3</sup> 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 mit zunehmendem Lernfortschritt kleinere Informationsmengen anbieten kann.

Entwicklung einer Spezifikation unterstützen. Hierzu wurden einige der von Collins (1977) entwickelten Regeln zur Systematisierung eines Sokratischen Dialogs auf das Problem der Entwicklung einer temporallogischen Spezifikation eines Systems übertragen. Die Entwicklung einer Spezifikation durch den Benutzer soll in sechs Schritten unterstützt werden:

1. *Akteure*. Der Benutzer wird zunächst nach den Akteuren befragt, die in dem zu modellierenden System beteiligt sein sollen.

2. *Zustände*. Für jeden Akteur werden die möglichen Zustände erfragt, in denen er sich befinden kann.

3. *Ausschlußbedingungen*. Da sich jeder Akteur nur jeweils in einem Zustand befinden kann, können über die verschiedenen Zustände jedes Akteurs Ausschlußbedingungen formuliert werden. Weitere Ausschlußbedingungen müssen erfragt werden.

4. *Fortschaltbedingungen*. Für jeden Zustand werden die Vorbedingungen erfragt ("Ask for prior factors" bei Collins). Die vom Benutzer angegebenen Informationen werden zur Bildung temporallogischer Fortschaltbedingungen genutzt.

5. *Konjunktive und disjunktive Verknüpfungen*. Dem Benutzer werden konjunktive und disjunktive Verknüpfungen zwischen den bis hierher entwickelten Formeln vorgeschlagen. (Z.B. wird die Ersetzung der Formeln " $\square P_1 \rightarrow \diamond C$ " und " $\square P_2 \rightarrow \diamond C$ " durch " $\square P_1 \vee P_2 \rightarrow \diamond C$ " vorgeschlagen.  $P_1, P_2$  sind Prämissen,  $C$  eine Conclusio.)

6. *Anfangsbedingungen*. Schließlich werden die Anfangsbedingungen erfragt.

Im Anschluß an die Implementierung dieser Spezifikationsentwicklung sind wiederum empirische Erprobungen vorgesehen. Darüber hinaus ist dann eine Komponente zu entwickeln, die Inkonsistenzen in den von Benutzern entwickelten Spezifikationen erkennt und sie dem Benutzer mitteilt, bzw. die den Benutzer bei der Suche nach Inkonsistenzen unterstützt.

### 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
- 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 (Cambridge Tracts in Theoretical Computer Science 23), 1991
- Pitschke, K., Zielidentifikation und Planen in Intelligenten Tutoriellen Systemen, in: A. Kobsa, W. Pohl (Hg), Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen, Workshop auf der KI93, Berlin, Sept. 1993
- 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 (AAE), 1993
- Wedig, A., Kalküle zur Entwicklung von Petri-Netzen aus Spezifikationen, Universität Oldenburg, Fachbereich Informatik: Diplomarbeit, 1993

**AIS --- AIS --- AIS --- AIS**

7. Kolloquium  
der Arbeitsgruppe Informatiksysteme

Michael Sonnenschein, Hans Fleischhack  
(Hrsg.)

Bericht Nr. AIS-18 - Juni 1994

**Arbeitsgruppe Informatik-Systeme**

**FB Informatik - Universität Oldenburg**

**Herausgeber der Berichtsreihe:**

**Dr. Hans Fleischhack  
Prof. Dr. Michael Sonnenschein**

**Anschrift der Autoren:**

**Fachbereich Informatik  
Carl von Ossietzky Universität Oldenburg  
Postfach 2503  
26111 Oldenburg**

**© Die Autoren 1994**

## **Vorwort**

Der vorliegende Bericht stellt die Ergebnisse dar, die beim siebten Kolloquium der Arbeitsgruppe Informatik-Systeme am 7. 1. 1994 präsentiert wurden.

Die Arbeitsgruppe Informatik-Systeme wurde mit dem Ziel gegründet, verschiedene Arbeitsbereiche des Fachbereichs Informatik stärker miteinander in Verbindung zu bringen. Dies soll unter anderem dazu dienen, methodische Grundlagenarbeit zur Vorbereitung des Oldenburger Forschungs- und Entwicklungsinstituts für Informatik-Werkzeuge und -Systeme (OFFIS) zu leisten.

Aus Mitteln der Stiftung Volkswagenwerk (Az. 210-70631/9-13-14/89) stehen der Arbeitsgruppe vom 1. 7. 1990 bis zum 31. 12. 1995 Mittel zur Finanzierung von 7,5 wissenschaftlichen Mitarbeiterstellen, Hilfskräften, Sachmitteln und Investitionen zur Verfügung.

**Michael Sonnenschein  
Hans Fleischhack**