AIS --- AIS --- AIS --- AIS

5. Kolloquium
der Arbeitsgruppe Informatik-Systeme

Michael Sonnenschein, Ulrike Lichtblau
(Hrsg.)

Bericht Nr. AIS-10 - Mai 1993

Arbeitsgruppe Informatik-Systeme

FB Informatik - Universität Oldenburg

Herausgeber der Berichtsreihe:

Dr. Ulrike Lichtblau
Prof. Dr. Michael Sonnenschein

Anschrift der Autoren:

Fachbereich Informatik
Carl von Ossietzky Universität Oldenburg
Postfach 2503
2900 Oldenburg

Vorwort

Aus Mitteln der Stiftung Volkswagenwerk (Az. 210-70631/9-13-14/89) stehen dem Fachbereich Informatik der Universität Oldenburg für die Arbeitsgruppe Informatik-Systeme vom 1.7.1990 bis zum 30.6.1994 Mittel zur Finanzierung von 7,5 wissenschaftlichen Mitarbeiterstellen, Hilfskräften, Sachmitteln, Mieten und Investitionen zur Verfügung.

Die Arbeitsgruppe wurde mit dem Ziel gegründet, verschiedene Arbeitsbereiche des Fachbereichs 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. OFFIS nahm am 1.1.1992 seine Arbeit auf. Die Aufbauphase des Instituts wird in etwa abgeschlossen sein, wenn die Arbeitsgruppe Informatik-Systeme ihre Arbeit beendet.

In halbjährlichen Abständen veranstaltet die Arbeitsgruppe Kolloquien zum Austausch von Ergebnissen. Der vorliegende Bericht entstand im Anschluß an das fünfte Treffen, das am 8.1.1993 stattfand.

Prof. Dr. Michael Sonnenschein
Dr. Ulrike Lichtblau

Erwerb von Modellierungswissen durch Hypothesentesten in PETRI-HELP

Projektleiter: Prof. Dr. Claus Möbus
Bereich Angewandte Informatik, Abteilung Lehr- Lernsysteme

Mitarbeiter: Knut Pitschke, Dr. Olaf Schröder
Wissenschaftliche Hilfskräfte: Jörg Folckers, Hermann Göhler

PETRI-HELP ist ein in der Entwicklung befindliches Intelligentes Hilfesystem zur individualisierten Unterstützung von Personen bei der Modellierung mit Petrinetzen des Bedingungs-Ereignis-Typs (Möbus et al., 1992a, b). Dieser Beitrag gliedert sich in vier Teile. Im ersten Teil werden der Begriff des Modellierungswissens und die sich daraus ergebenden Anforderungen an das System diskutiert. Im zweiten Teil folgt eine zusammenfassende Beschreibung des aktuellen Implementierungsstandes von PETRI-HELP. Der dritte Teil stellt empirische Erfahrungen und Ergebnisse vor, die seit Herbst 1992 mit dem System gewonnen wurden. Es wird dargestellt, welche Aspekte von Modellierungswissen PETRI-HELP gegenwärtig unterstützt und welche nicht. Der vierte Teil beschreibt Arbeitsschwerpunkte der Modifikation und der Weiterentwicklung von PETRI-HELP. Sie ergeben sich aus den gegenwärtig nicht berücksichtigten Aspekten des Modellierungswissens und aus den empirischen Ergebnissen. Ein kurzer Ausblick schließt sich an.

1. Modellierungswissen

Den Ausgangspunkt für eine Taxonomie des Wissens, das bei einer Tätigkeit wie der Modellierung mit Petrinetzen benötigt wird, bildet unsere ISP-DL-Theorie (impasse-success-problem solving driven learning; Möbus, 1991; Möbus et al., 1992b), welche Ansätze des Wissenserwerbs (z.B. Anderson, 1989; van Lehn, 1991) und des Handelns und Problemlösens (z.B. Gollwitzer, 1990) miteinander verbindet:

- Neues Wissen wird durch den Einsatz von Heuristiken in Stocksituationen erworben.
- Bereits erworbenes Wissen wird durch erfolgreiche Problembearbeitungen optimiert.
- Problemlöseprozesse bestehen aus vier Teilphasen: Abwägen, Planen, Ausführen, Bewerten.

In Bezug auf die Modellierung mit Petrinetzen entsprechen diesen vier Problemlösephasen:

- *Spezifikationswissen* zur Erarbeitung einer Problembeschreibung
- *Transformationswissen* zur Entwicklung eines Lösungsplans
- *Implementationswissen* zur Konstruktion eines Netzentwurfs
- *Evaluationswissen* zur Bewertung einer Problembeschreibung, eines Lösungsplans oder eines Netzentwurfs.

Jede dieser vier Teiltätigkeiten einer Modellierung kann wiederum in die Problemlösephasen des Abwägens, Planens, Ausführens und Bewertens aufgegliedert werden (Tabelle 1, zu dem fett umrahmten Bereich siehe weiter unten).

	Abwägen	Planen	Ausführen	Bewerten
Spezifikations-wissen	Wissen um Ziele, Motive, Vorgaben	Heuristiken der Informationssuche	Wissen zur Informationsgewinnung	Wissen zur Beurteilung von Eindeutigkeit, Widerspruchsfreiheit
Transformations-wissen	Wissen um Planungsstrategien und -operatoren	Heuristiken zur Auswahl und Sequenzierung von Planungsoperatoren	Ausführungswissen: Planerstellung	Wissen zur Beurteilung von Vollständigkeit, Ausführbarkeit
Implementations-wissen	Wissen um Implementationsstrategien und -operatoren	Heuristiken zur Auswahl und Sequenzierung von Implementationsoperatoren	Ausführungswissen: Implementation	Wissen zur Beurteilung der Vollständigkeit
Evaluations-wissen	Wissen um Evaluationsquellen	Evaluationsheuristiken	Ausf.wissen: Evaluation (Prüfhypothesen)	Wissen zur Interpretation des Ergebnisses

Tabelle 1: Arten von Modellierungswissen

So besagt die erste Zeile von Tabelle 1, daß bei der Erarbeitung einer Spezifikation zunächst Wissen um Ziele, Motive und evtl. Vorgaben aktualisiert und gegeneinander *abgewogen* werden muß. Dann muß *geplant* weden, welche Informationen evtl. beschafft werden müssen, welche Informationsquellen in welcher Reihenfolge "angezapft" werden sollen usw. Dazu ist entsprechendes heuristisches Wissen notwendig. Die *Ausführung* erfordert Wissen zur Beschaffung der benötigten Information und ihrer Integration zu der Problemspezifikation. Diese muß schließlich *bewertet* werden (z.B. eindeutig? widerspruchsfrei?), was Wissen um Bewertungskriterien, ihre Anwendung, Gewichtung usw. erfordert.

Aus dieser Taxonomie folgt, daß PETRI-HELP den Petrinetzmodellierer beim Erwerb und Einsatz dieser verschiedenen Wissensarten unterstützen sollte. Als weitere Anforderung ergibt sich hieraus, daß empirische Indikatoren für die verschiedenen Problemlösephasen gefunden werden müssen, die das System online erkennen kann und die die Bereitstellung von Hilfen ermöglichen, die der jeweiligen Problemlösephase angemessen sind.

2. Implementationsstand von PETRI-HELP

Neben dem Kriterium, daß PETRI-HELP verschiedene Problemlösephasen unterstützen sollte, ist das System an weiteren aus der ISPDL-Theorie folgenden Designkriterien orientiert (Möbus et al., 1992b, c):

• Der Benutzer wird Informationen nur akzeptieren bzw. von sich aus aktiv anfordern, wenn er sich in einer Stocksituation befindet. Das System stellt also Hilfen *bereit*, ohne aber den Benutzer zu unterbrechen.

• Der Benutzer wird als Hilfe gedachte Informationen nur akzeptieren und nutzen, wenn sie auf sein Vorwissen Bezug nehmen, und wenn der Benutzer sein Vorwissen so weit wie möglich einsetzen kann. Die Informationen und Hilfen sollten daher den *Wissensstand* des Benutzers berücksichtigen. Dieser Aspekt ist in PETRI-HELP bisher nicht realisiert.

• Die selbständige Nutzung von Informationen und Hilfen sowie der Einsatz von Vorwissen werden durch einen *Hypothesentestansatz* in besonderem Maße unterstützt. Der Lernende kann Prüfhypothesen über seinen Lösungsentwürfen oder Teilen davon formulieren und vom System untersuchen lassen. Er bekommt Rückmeldungen und gegebenenfalls Ergänzungs- und Korrekturvorschläge. Diesen Ansatz haben wir bereits in dem Problemlösemonitor ABSYNT realisiert (Möbus, Schröder, Thole, 1992).

PETRI-HELP besteht aus folgenden Komponenten:

• einer Sammlung von derzeit zehn *Entwurfsaufgaben*. Jede Entwurfsaufgabe ist als Menge *temporallogischer Formeln* (Kröger, 1987) spezifiziert. (Abb. 1 rechts zeigt ein Beispiel. "[]" bedeutet "in jeder Welt gilt, daß ...". "◊" bedeutet "in irgendeiner zukünftigen Welt wird gelten, daß ...". "o" bedeutet "in jeder nächstmöglichen Welt wird gelten, daß ..."). Die Aufgabe des Benutzers besteht darin, ein Netz zu konstruieren, das diese gegebene Formelmenge erfüllt. Die temporallogische Spezifizierung der Aufgaben ermöglicht, daß Netzentwürfe vom System durch Model-Checking (Josko, 1990) untersucht werden können (Möbus et al., 1992b, c), indem die Formeln auf dem Fallgraphen interpretiert werden.

• einem *Netzeditor* zur Konstruktion und Simulation von Bedingungs-Ereignis-Netzen.

• einer *Hypothesentestkomponente*. In einem Fenster wählt der Benutzer aus der Menge der Aufgabenformeln die Formeln aus, die er durch den aktuellen Stand des Entwurfs für erfüllt hält (= Formulierung einer Prüfhypothese: Abb. 1 oben links). Aus dem Entwurf erzeugt das System den Fallgraphen und interpretiert auf ihm die vom Benutzer ausgewählten Formeln. Dem Benutzer werden die erfüllten und nicht erfüllten Formeln der Hypothese zurückgemeldet (Abb. 2).

• einer *Ergänzungs-* und *Korrekturkomponente*. Der Benutzer kann sich von dem System seinen aktuellen Entwurf ergänzen bzw. korrigieren lassen. Grundlage hierfür ist eine *Lernkomponente*, die aus Interaktionssequenzen von Benutzern Regeln generiert, die dann für spätere Benutzer zur Verfügung stehen. Es gibt zwei Arten von Regeln:

Netz-Netz-Regeln assoziieren zwei Teilnetze miteinander, wobei das zweite Netz eine echte Obermenge der vom ersten Netz erfüllten Formelmenge erfüllt. Wenn der Entwurf des Benutzers dem ersten Netz topologisch äquivalent ist, kann die Differenz der beiden Netze als Ergänzungsvorschlag angeboten werden. Die beiden Netze in einer Netz-Netz-Regel entsprechen zwei Stadien in der Entwicklung eines Netzentwurfs durch einen Benutzer.

Formel-Netz-Regeln assoziieren Formelmengen mit Teilnetzen. Auch hier wird ein Netz einer Regel, das eine Obermenge der durch den aktuellen Entwurf erfüllten Formeln erfüllt, mit dem aktuellen Entwurf verglichen. Als Ergebnis dieses Vergleichs werden Ergänzungs- und Korrekturvorschläge angeboten.

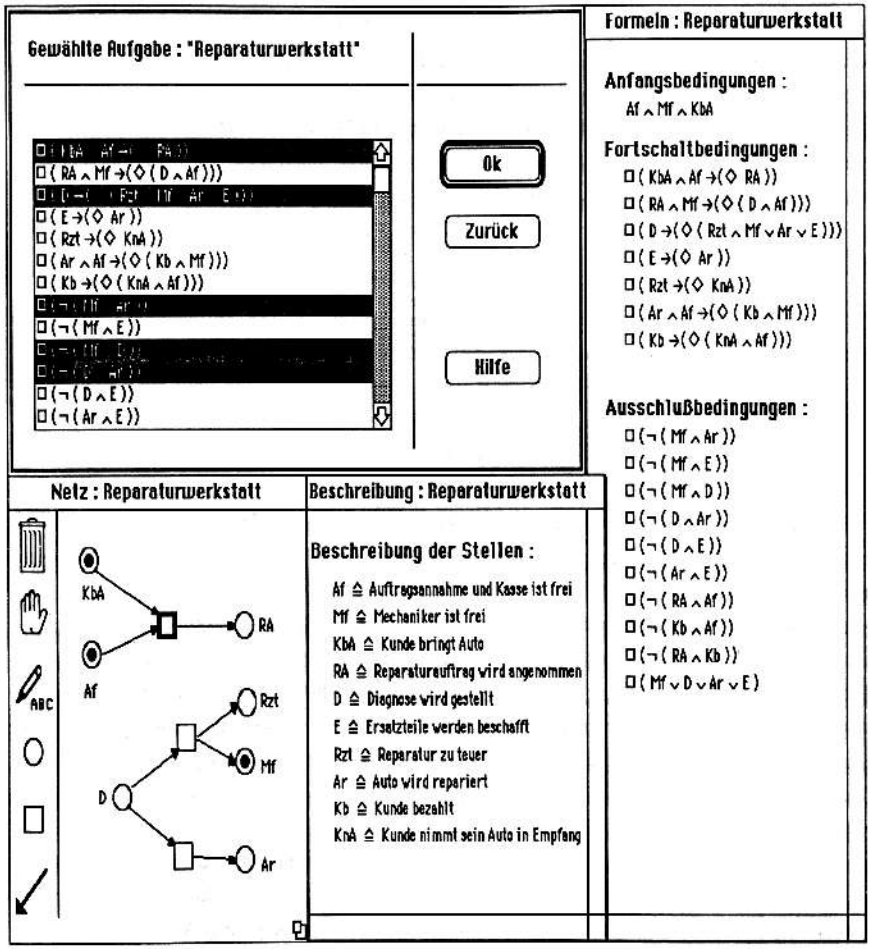

Abb. 1: Schnappschuß der Oberfläche von PETRI-HELP. Rechts: Aufgabe als Menge temporallogischer Formeln. Unten Mitte: Erläuterung der Abkürzungen. Unten links: Netzeditor mit in der Entwicklung befindlichem Lösungsentwurf. Oben links: Prüfhypothese: von dem Benutzer für erfüllt gehaltene (= schwarz markierte) Formeln

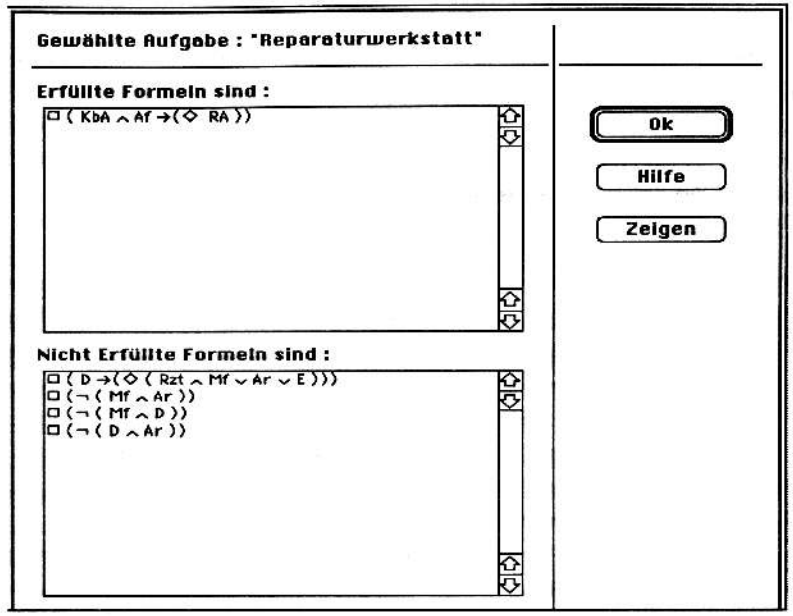

Abb. 2: Rückmeldung erfüllter und nicht erfüllter Formeln für die Hypothese und den Entwurf in Abb. 1

Die durch beide Arten von Regeln erzeugten Vorschläge stellen also sicher, daß die Menge der gerade erfüllten Formeln anwächst. Zur Generierung der Vorschläge werden Regeln so ausgewählt, daß eine möglichst kleine Obermenge der gerade erfüllten Formeln erfüllt wird. Dadurch, daß das System lernt, wird es zunehmend kleinere Informationsmengen pro Ergänzungsvorschlag anbieten. Somit bekommt der Benutzer möglichst nicht mehr Informationen, als zur Überwindung der aktuellen Stocksituation notwendig sind. Abb. 3 zeigt vom System generierte Ergänzungsvorschläge für den Netzentwurf aus Abb. 1. "D", "Mf" usw. stehen für die entsprechenden Stellen im Entwurf. "place#36" steht für eine neue, unbenannte Stelle.

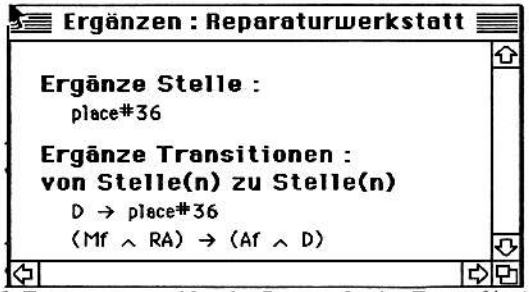

Abb. 3: Ergänzungsvorschlag des Systems für den Entwurf in Abb. 1

3. Erprobungen von PETRI-HELP

PETRI-HELP wurde im Rahmen eines Fortbildungskurses und einer Lehrveranstaltung empirisch erprobt. Wir wollen hier die Ergebnisse zu den folgenden Aspekten skizzieren:

a) Welche Strategien werden beim Netzentwurf eingesetzt?

b) Wie werden das Hypothesentesten und die Rückmeldungen erfüllter und nicht erfüllter Formeln von den Benutzern akzeptiert und genutzt?

c) Wie werden die Ergänzungs- und Korrekturvorschläge akzeptiert und genutzt?

Zu a): Eine häufig beobachtete und von den Benutzern auch in verbalen Berichten genannte Strategie bestand darin, daß die Fortschaltbedingungen der Formelmenge (vgl. Abb. 1 rechts) von oben nach unten mittels "Designheuristiken" (Möbus et al., 1992a) in Netzfragmente umgesetzt wurden. Die Designheuristiken sind hypothetische Regeln, die jeweils einer Formel ein Netzfragment zuordnen. Für das so konstruierte Netz wurde dann die Hypothese formuliert, daß es alle Formeln erfülle. Diese Vorgehensweise ist relativ schematisch und erfordert kaum wirkliches Problemlösen - außer in den Fällen, wo nach Umsetzung aller Fortschaltbedingungen noch unerfüllte Ausschlußbedingungen vorliegen.

Zu b): Das Hypothesentesten wurde positiv aufgenommen. Durch Rückmeldung erfüllter und unerfüllter Formeln wird die Aufmerksamkeit des Benutzers auf unerfüllte Teile der Aufgabenspezifikation gelenkt, ohne aber bereits Lösungsteile vorwegzunehmen.

Zu c): Die Ergänzuings- und Korrekturvorschläge wurden dagegen weniger positiv aufgenommen, und zwar aus drei Gründen:

- Indem konkrete Stellen, Transitionen und Kanten vorgeschlagen werden, werden Teile der Lösung vorweggenommen, statt z.B. nur einen Lösungsweg oder -ansatz vorzuschlagen.
- Es werden meist mehr Information vorgegeben, als der Benutzer aktuell benötigt. (Dies hängt allerdings vom Lernstand des Systems ab.)
- Die vorgeschlagenen Informationen werden nicht erklärt. Damit gibt PETRI-HELP tendenziell zu viele, unerklärte Informationen vor. Dies führt nach unserer Theorie günstigenfalls zu Selbsterklärungseffekten (z.B. Chi et al., 1989), im Regelfall aber wohl eher zu passiver Übernahme der angebotenen Information, evtl. auch zu Verärgerung (zur Ableitung dieser und weiterer Vorhersagen aus der ISPDL-Theorie siehe Möbus, Schröder, Thole, 1992). Diese Reaktionen konnten wir auch bei einigen Benutzern beobachten.

Zusammenfassend unterstützt PETRI-HELP gegenwärtig die Teilsapekte von Modellierungswissen, die in Tabelle 1 fett umrandet sind: Die Angabe erfüllter und nicht erfüllter Formeln nach Prüfhypothesen unterstützen die Auswahl und Sequenzierung von Implementationsoperatoren (Plazieren, Verschieben und Löschen von Stellen, Transitionen, Kanten, sowie Beschriften von Stellen). Die Implementation geschieht im Netzeditor. Die Ergänzungs- und Korrekturvorschläge sind direkte Implementationshilfen. Die Evaluation von Entwürfen wird durch Hypothesentesten sowie auch durch die Netzsimulation unterstützt.

PETRI-HELP unterstützt gegenwärtig also weder Spezifikations- noch Planungswissen. Hervorzuheben ist jedoch, daß aufgrund des Model-Checking-Ansatzes jeder Netzentwurf vom System untersucht und kommentiert werden kann. Dadurch wird freies, explorativ es Problemlösen gefördert.

4. Erweiterungen und Modifikationen von PETRI-HELP

Um weitere Aspekte von Modellierungswissen zu unterstützen und die von Benutzern geäußerten Kritikpunkte in Verbesserungen umzusetzen, muß PETRI-HELP in vier Aspekten weiterentwickelt bzw. modifiziert werden:

a) Unterstützung der Spezifikation von Problem-/Aufgabenstellungen. Der Benutzer soll Spezifikationen selbständig im Dialog mit dem System entwickeln können.

b) Unterstützung der schrittweisen Transformation von Spezifikationen über Zwischenspezifikationen und -ziele. Dies soll dem Benutzer die Möglichkeit geben, einen Lösungsplan zu entwickeln, ohne gleich auf der Ebene der Petrinetzkonstrukte zu denken.

c) Generierung von Erklärungen für die Rückmeldungen erfüllter und nicht erfüllter Formeln sowie für die Ergänzungs- und Korrekturvorschläge.

d) Generierung individualisierter Hilfen.

Die Aspekte c und d werden hier nicht diskutiert werden, weil sie u.a. von den Lösungen der Aspekte a und b abhängen.

Zu a): Es wird gegenwärtig untersucht, inwieweit die Entwicklung einer Problemspezifikation durch den Benutzer im Sinne eines sokratischen Dialogs (z.B. Collins, 1977) gestaltet werden könnte: Der Benutzer legt in Interaktion mit dem System fest, welche Beteiligten oder Agenten in dem System vorkommen sollen, welche Eigenschaften und Zustände die Beteiligten haben bzw. annehmen können, welche Aktionen die Beteiligten ausführen können, welche Voraussetzungen dafür erfüllt sein müssen, welche Aktionen sich gegenseitig ausschließen, welche Aktionen nur in bestimmten Situationen ausgeführt werden dürfen, und dergleichen. Das System kann diesen Explorationsprozeß durch entsprechende Fragen an den Benutzer (Collins, 1977) steuern. Die Entwicklung der Spezifikation und die Entwicklung einer Petrinetzlösung können dabei Hand in Hand gehen und sich gegenseitig beeinflussen. Der Netzentwurf kann dabei nach wie vor durch Model-Checking analysiert werden.

Zu b): Um die schrittweise Transformation einer Spezifikation in einen Netzentwurf unterstützen zu können, wurde aus Basis des von Olderog (1991) entwickelten Transformationsansatzes eine "Papier-und-Bleistift"-Anwendung für PETRI-HELP entwickelt und empirisch in einer Einzelfallstudie mit einer Probandin erprobt, die keine Vorkenntnisse mit Petrinetzen hatte.

Die Aufgabenstellung für die Probandin bestand in der Konstruktion eines Netzes, das die Tätigkeiten und Interaktionen zweier Agenten beschreibt, z.B. die Tätigkeiten von Koch und

Kellner in einem Restaurant. Der Ausgangszustand ist folgendermaßen definiert: Die letzte Handlung des Kochs bestand darin, daß er den Kellner informierte, daß ein Essen fertig ist (abgekürzt "Koch nach f"). Der Kellner hat gerade dem Gast ein Essen serviert: "Kellner nach e". Ferner waren die möglichen Handlungen und ihre Abfolgen vorgegeben. Tabelle 2 zeigt die Aufgabenstellung und die Vorgaben. Neben der verbalen Beschreibung wurde die Aufgabe als *Zielregion* vorgegeben (Abb. 4): Das zu konstruierende Netz, das der gegebenen Spezifikation genügen soll, wird durch ein "?" symbolisiert.

Konstruiere ein Netz, **T(Koch nach f) und T(Kellner nach e)**, das die Tätigkeiten und Interaktionen von Koch und Kellner in einem Restaurant beschreibt.

Vorgaben:

	Koch	Kellner
1. Tätigkeiten:	a (Auftrag von Kellner an Koch) b (bereitet Essen zu) f (Information von Koch an Kellner: Essen fertig)	s (schlafen) a (Auftrag von Kellner an Koch) f (Information von Koch an Kellner: Essen fertig) g (Bestellung vom Gast) e (Gast Essen servieren)
2. Reihenfolge:	a --> b b --> f f --> a	s --> f oder g f --> a e oder a --> s g --> a
3. Ausgangszustand	Koch nach f	Kellner nach e

Tabelle 2: Aufgabenstellung und Vorgaben für die Transformation in einen Netzentwurf

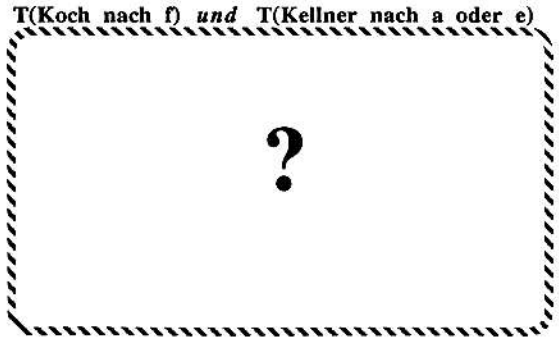

Abb. 4: Aufgabenstellung für die Transformation in einen Netzentwurf

Abb. 5 zeigt einige graphische Umsetzungen der von Olderog (1991, S. 171f) beschriebenen Transformationsregeln: Die Parallelitätsregel spaltet ein Problem, dessen Spezifikation aus einer Konjunktion besteht, in Teilprobleme auf, deren Lösungen (Teilnetze) später zu synchronisieren sind. Die Präfixregel zerlegt eine Spezifikation, nach der nur eine Aktion, y, als nächstes möglich ist (init(A) = {y}), in eine Stelle, eine Transition y und eine Restspezifikation. Die Netzsynchronisationsregel fügt Teilnetze an ihren gemeinsamen Transitionen zusammen.

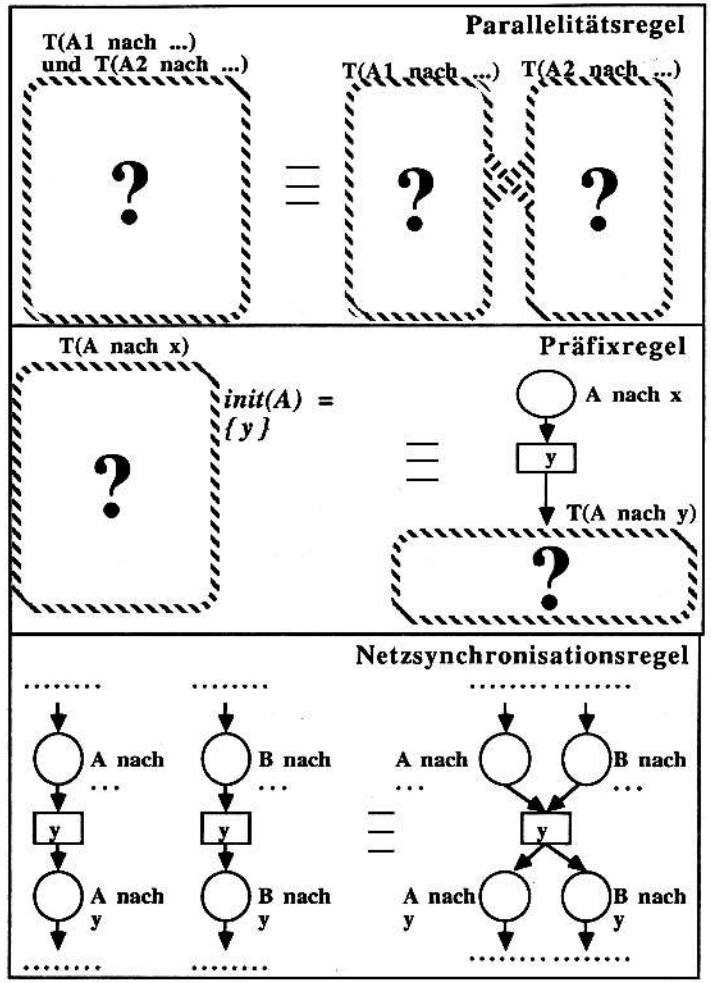

Abb. 5: Parallelitätsregel, Präfixregel und Netzsynchronisationsregel in Anlehnung an Olderog (1991)

Diese und weitere graphische Transformationsregeln standen der Probandin als Instruktions- und Hilfematerial zur Verfügung. Ferner wurden ihr Kärtchen mit Stellen, Transitionen,

Kanten und Zielregionen ausgehändigt, die sie beschriften und auf dem Tisch anordnen mußte. Außerdem erhielt die Person schriftliche Insruktionsmaterialien zum Versuchsablauf.
Mit den gegebenen Materialien konstruierte die Probandin ausgehend von Abb. 4 den in Abb. 6 dargestellten Netzentwurf. Die gesamte Untersuchung dauerte etwa zwei Stunden.

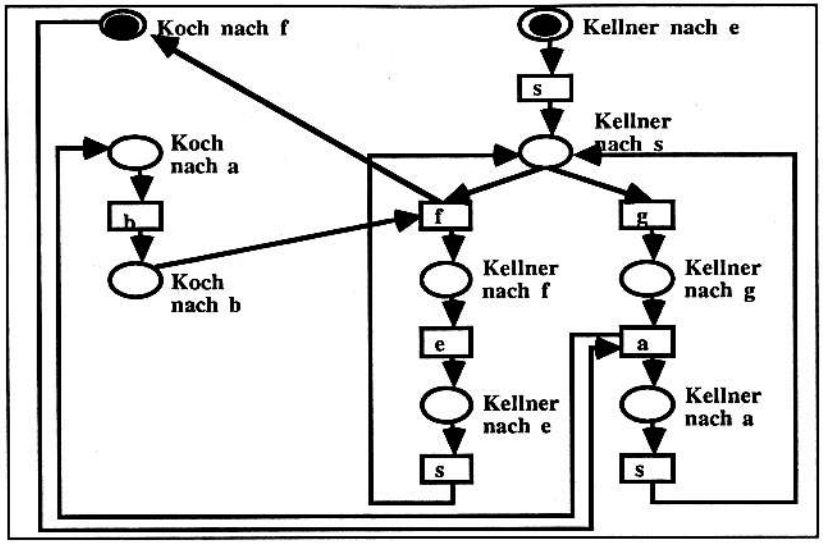

Abb. 6: Von der Probandin mit den Transformationsregeln konstruierter Netzentwurf

Diese empirische Untersuchung hat gezeigt, daß Instruktions- und Hilfematerial auf der Basis des Transformationsansatzes von eine ungeübten Person ohne längere Einarbeitung benutzt werden kann. Es waren sogar nach kurzer Zeit Lerneffekte beobachtbar, denn die Probandin begann, die Konstruktion der Zwischenspezifikationen bei der Anwendung der Präfixregel zu überspringen. Derartige Lerneffekte haben wir in ABSYNT durch Regelkomposition modelliert (Möbus, Schröder, Thole, 1992).
Da der Transformationsansatz das Arbeiten mit Teilspezifikationen bzw. "Zielregionen" erlaubt, könnte dieser Ansatz dahingehend weiterentwickelt werden, daß de Benutzer Hypothesen nicht nur auf Netzentwürfen, sondern auch mit unfertigen Teilplänen (den Zielregionen) formuliert und so in einem sehr frühen Planungsstadium Rückmeldung über den eingeschlagenen Lösungsansatz erhält. Auch Ergänzungs- und Korekturvorschläge könnten dann auf der Ebene von Teilspezifikationen erfolgen.

5. Ausblick

Die oben in den Abschnitten 3 und 4 beschriebenen empirischen Untersuchungen haben deutlich gemacht, daß das Wissen, das zur Umsetzung *vorliegender* Spezifikationen in (Bedingungs-Ereignis-) Petrinetze benötigt wird, auch Novizen in relativ kurzer Zeit und zum Teil überraschend schnell vermittelt werden kann. Ein stärkeres Gewicht muß demgegenüber auf die Unterstützung der *Entwicklung* von Spezifikationen gelegt werden. Dazu gehören Hilfen für die Akquisition und Integration von Wissen, das für die Spezifikation eines Problems benötigt wird. Auch ist es wichtig, in Stocksituationen die Akquisition von Wissen zu unterstützen, das der Benutzer benötigt, um Fehler selbständig suchen und korrigieren zu können sowie Entscheidungen für Planungsstrategien treffen zu können.

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

Josko, B., Verifying the Correctness of AADL Modules using Model Checking, in: de Bakker, de Roever, Rozenberg (eds), Proceedings REX-Workshop on Stepwise Refinement of Distributed Systems: Models, Formalisms, Correctness, Berlin: Springer LNCS 430, 1990, 387-400

Kröger, F., Temporal Logic of Programs, Berlin: Springer, 1987

Möbus, C., Wissenserwerb mit kooperativen Systemen, in V. Claus, P. Gorny (Hg), Informatik: Wege zur Vielfalt beim Lehren und Lernen, Informatik-Fachberichte, Band 292, Berlin: Springer, 1991, 288-298

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 (a)

Möbus, C., Pitschke, K., Schröder, O., Towards the Theory-Guided Design of Help Systems for Programming and Modelling Tasks, in C. Frasson, G. Gauthier, G.I. McCalla (eds), Intelligent Tutoring Systems, Proceedings of the Second International Conference ITS 92, Montreal, Berlin: Springer LNCS 608, 1992 (b)

Möbus, C., Pitschke, K., Schröder, O., Folckers, J., Göhler, H., Stand der Realisierung von PETRI-HELP, in M. Sonnenschein, U. Lichtblau (Hg), 4. Kolloquium der Arbeitsgruppe Informatik-Systeme, Bericht AIS-6, Universität Oldenburg, FB Informatik, April 1992 (c)

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

Olderog, E.R., Nets, Terms, and Formulas, New York: Cambridge (Cambridge Tracts in Theoretical Computer Science 23), 1991

Pitschke, K., Schröder, O., Möbus, C., Entwurf eines Hilfesystems für Petrinetzmodellierer, in V. Claus, P. Gorny (Hg), Informatik: Wege zur Vielfalt beim Lehren und Lernen, Informatik-Fachberichte, Band 292, Berlin: Springer, 1991, 299-305