

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

**1. Kolloquium  
der Arbeitsgruppe Informatik-Systeme**

**Volker Claus, Ulrike Lichtblau (Hrsg.)**

**Bericht Nr. AIS-1 - März 1991**

**Arbeitsgruppe Informatik-Systeme**

**FB Informatik - Universität Oldenburg**

Herausgeber der Berichtsreihe:

Die Professoren des Fachbereichs Informatik

Anschrift des Autors / der Autoren:

Fachbereich Informatik  
Universität Oldenburg  
Postfach 2503  
2900 Oldenburg

© Der Autor / die Autoren 1991

## Vorwort

Am 1.7.1990 wurde die Arbeitsgruppe Informatik-Systeme gegründet mit dem Ziel, die verschiedenen Arbeitsbereiche des Fachbereichs Informatik stärker zueinander zu bringen. Diese Zusammenarbeit soll unter anderem der Vorbereitung des geplanten Oldenburger Forschungs- und Entwicklungsinstituts für Informatik-Werkzeuge und -Systeme (OFFIS) dienen.

Die Arbeitsgruppe erarbeitet Prinzipien und Methoden zur Entwicklung von Software-Werkzeugen, mit deren Hilfe komplexe Programm-Systeme und höhere Informatik-Werkzeuge rascher und sicherer erstellt werden können. Dieser Untersuchungsbe-  
reich ist von wissenschaftlicher und wirtschaftlicher Bedeutung, da zum einen die Komplexität der zu erstellenden Systeme nur noch mit Werkzeug-Unterstützung bewältigt und zum anderen eine Auslagerung der Informationstechnologie in Billig-  
länder auf diesem Wege verhindert werden kann.

Konkret werden folgende Projekte bearbeitet:

- Erforschung und Entwicklung eines Planungssystems für Entwurf, Analyse und Simulation von Vorgängen, vorwiegend für Anwendungen im Büro (Prof. Claus)
- Benutzungsoberflächen und Fragen der Visualisierung im Bereich von Planungssystemen (Prof. Appelrath)
- Wissensstandsbezogenes Design einer adaptiven Systemoberfläche, um Benutzern eine optimale Arbeitsumgebung zur Verfügung zu stellen, die vom Zeitpunkt und vom Wissensstand abhängt (Prof. Möbus)
- Rechnerunterstützte Entwicklung von verteilten Systemen, vor allem von Multi-  
prozessorarchitekturen (Prof. Damm)
- Anschluß öffentlicher Kommunikationsdienste, auch mit dem Ziel, Bild, Text und Sprache zu integrieren (Prof. Jensch).

Die einzelnen Teilvorhaben hängen zum einen eng thematisch zusammen, da sie sich mit Planungssystemen und deren Umfeld befassen, zum anderen sind sie methodisch sehr nahe verwandt, da die Darstellungs-, Spezifikations- und Entwurfsmethodiken in allen Bereichen auf ähnlichen Konzepten aufbauen.

Daneben werden weitere Fragestellungen unter dem Aspekt einer möglichen Einbet-  
tung in das Gesamtvorhaben untersucht :

- Benutzerschnittstellen-Entwicklung (Prof. Gorny)
- Entwicklungs-System für verteilte Programme (Prof. Spies)
- Konnektionistische Expertensysteme (Prof. Scheerer, Megnet)
- Spezifikation offener kommunizierender Systeme (Prof. Olderog)
- Aufbau des Oldenburger Rechnernetzlabors (Prof. Kowalk)
- Einrichtung eines CAD-Trainingszentrums (Priv.Do. Reich).

Aus Mitteln der Stiftung Volkswagenwerk (Az. 210-70631/9-13-14/89) stehen dem Fachbereich Informatik vom 1.7.1990 bis zum 31.12.1994 für die Arbeitsgruppe Informatik-Systeme insgesamt mehr als drei Millionen DM zur Verfügung. Hieraus werden 7,5 wissenschaftliche Mitarbeiter, Hilfskräfte, Sachmittel, Mieten und Investitionen finanziert.

- **Kooperatives Arbeiten:** Die Bilddaten befinden sich an einem der an der Kommunikation beteiligten Arbeitsplätze und werden bei Bedarf zum anderen Arbeitsplatz übertragen.
- **Koordiniertes Arbeiten:** Die Bilddaten werden insgesamt vor der gemeinsamen Bearbeitung an den Kommunikationspartner übertragen. Während der Bearbeitung werden dann nur noch Kommandos zur Annotation ausgetauscht, die synchron an beiden Arbeitsplätzen ausgeführt werden. Dadurch wird das Kommunikationsmedium insgesamt weniger belastet.

## 6 Technische Realisierung

Entwicklungsziel für DIMMI ist weitgehende Unabhängigkeit von der verwendeten Hardware. Um dieses Ziel zu erreichen, werden gezielt Standardwerkzeuge zur Softwareentwicklung eingesetzt. Als Betriebssystem wird UNIX eingesetzt. Die Benutzeroberflächen sind mit OSF/Motif bzw. NeXT-Step realisiert und für die Kommunikationsfunktionalität wird das TCP/IP-Protokoll benutzt. Dadurch kann der Entwicklungsaufwand bei der Portierung auf unterschiedliche Rechner klein gehalten werden. Lediglich für bestimmte Teilfunktionalitäten (z.B. Aufzeichnung einer Bildsequenz auf Video) sind speziellere Hard- und Softwareanpassungen erforderlich.

## 7 Anwendungsfelder

- **Operationsplanung:** Verschiedene Bildmodalitäten können überlagert und in einer Bildsequenz aneinandergesetzt werden, die dann von einer Reihe von Spezialisten begutachtet und annotiert wird. Dem Operationsteam steht die so erstellte Szene bei der Operation zur Verfügung.
- **Ferndiagnose:** Durch die Möglichkeit des kooperativen Arbeitens kann bei Bedarf ein räumlich weit entfernter Spezialist in die Diagnose mit einbezogen werden. Besonders nützlich kann dies in Regionen, in denen die diagnostischen Möglichkeiten auf verschiedene Kliniken verteilt sind, sein.
- **Archivierung:** Bildsequenzen können zusammen mit Annotationen als multimediales Dokument archiviert werden. Durch die fortschreitende Technologie besonders im Bereich der magneto-optischen Speichermedien gewinnt die elektronische Archivierung medizinischen Bildmaterials zunehmend an Attraktivität. Für die medizinische Diagnostik ergibt sich der Vorteil, daß zu einem gerade untersuchten Fall Bildmaterial vergleichbarer Fälle schnell wiedergefunden und nutzbar gemacht werden kann.
- **Ausbildung:** Bildmaterial medizinischer interessanter Fälle kann vollständig annotiert in speziellen Archiven abgelegt und zur Aus- und Fortbildung verwendet werden. Durch die Möglichkeit der Vernetzung mit ISDN können in medizinischen Zentren Archive angelegt werden, die auch von praktischen Ärzten genutzt werden können. Annotierte Bildsequenzen können auch auf herkömmliche Videokassetten überspielt und auf diese Weise verteilt werden.
- **Mobile Information:** Es ist anzunehmen, daß neue Telefonanschlüsse zunehmend in ISDN-Technologie realisiert werden. Durch einen speziell konfigurierten Laptop-Computer können daher an jedem Ort mit ISDN-Anschluß Bilddaten über vorhergehende Untersuchungen eines Patienten abgefragt werden. Da der ISDN-Standard auch die drahtlose Kommunikation vorsieht, ist eine technische Unterstützung eines behandelnden Arztes an nahezu jedem Ort möglich. Besondere Bedeutung kann dies in der Notfallmedizin gewinnen, wo wichtige Patientendaten (soweit vorhanden) in kürzester Zeit zur Verfügung stehen müssen.

**Projekt:** "Ein wissensstandsbezogenes Hilfesystem für Petrinetz-Modellierer"

**Projektleiter:** Prof. Dr. Claus Möbus, FB 10, Angewandte Informatik, Abt. Lehr-Lernsysteme  
**Mitarbeiter:** Knut Pitschke, Olaf Schröder

**Projektziel:** Ziel des Projekts ist die Entwicklung eines Hilfesystems, das Personen bei der Modellierung mit Petrinetzen auf verschiedenen Entwurfsebenen mit wissensstandsbezogenen Planungshilfen unterstützt. Das Hilfesystem soll Benutzern mit unterschiedlichem Erfahrungshintergrund zu jedem Zeitpunkt eine optimale, dem aktuellen Wissensstand des jeweiligen Benutzers angepaßte Arbeitsumgebung zur Verfügung stellen. Unerfahrenen Benutzern soll es möglich sein, sich durch die Interaktion mit dem System in die Modellierung mit Petrinetzen einzuarbeiten und sich schrittweise zu einem "Experten" zu entwickeln.

Dieser Prozeß soll in *drei Phasen* erfolgen. Zum einen soll es den Benutzern möglich sein, durch *"freies", ungeleitetes Problemlösen* eigenständige Lösungsentwürfe zu vorgegebenen Aufgaben zu entwickeln und sich dabei von dem System individualisierte Hilfen geben zu lassen. Zum anderen soll die Möglichkeit zu *regelgeleitetem, durch adaptive Planungshilfen unterstütztem Problemlösen* bestehen. Schließlich ist die Bearbeitung "offener", selbst gestellter bzw. nicht oder nur unvollständig spezifizierter Planungsprobleme vorgesehen. Bei der Modellierung natürlicher Systeme (Organisationen, Produktionsabläufe, Bürokommunikation etc.) dürften fast immer *"offene" Planungsprobleme* vorliegen.

Die Entwicklung wissensstandsbezogener Hilfen setzt detaillierte Annahmen über das aktuelle Wissen des Benutzers (Benutzermodell) voraus. Grundlage für diese Annahmen ist die über den Benutzer rechnerseitig erfaßbare Information und ihre Interpretation mit Hilfe differenzierter Hypothesen zum Wissenserwerbsprozeß. Solche Hypothesen wollen wir auf Basis der IDL-SDL-Theorie ("impasse-driven learning - success-driven learning" [van Lehn, 1988; 1990; Wolff, 1987]) sowie empirischer Einzeluntersuchungen gewinnen.

Vorarbeiten zur Analyse von Lösungsentwürfen, zur Generierung von Hilfen und zur Modellierung hilfegeleiteter Wissenserwerbsprozesse wurden in unserem DFG-Projekt ABSYNT für die Domäne des funktionalen Programmierens durchgeführt (s.a. unseren OFFIS-Antrag vom 29.11.89). Das Hilfesystem für Petrinetzmodellierer wird in enger Zusammenarbeit mit dem OFFIS-Projekt MOBY (Prof. Claus) entwickelt. Ferner sollen für die Entwicklung der Wissensdiagnostikbausteine Transformations- und Verifikationskomponenten übernommen werden; hierzu laufen Kooperationsgespräche mit Prof. Damm und Prof. Olderog.

Unser Meilensteinplan sieht grob wie folgt aus: Bis Ende 1991 sollen für eine begrenzte Sequenz einfacher Aufgaben ein Expertenmodell und eine erste Version eines Benutzermodells entwickelt werden. Der Benutzer soll für diese Aufgaben Petrinetze entwerfen und vom System untersuchen lassen können und Ergänzungsvorschläge erhalten. 1992 sollen empirische Einzeluntersuchungen von Petrinetzmodellierern folgen, um die abgelaufenen Wissenserwerbsprozesse im Sinne der IDL-

SDL-Theorie zu modellieren. In Zusammenarbeit mit einem der kooperierenden Projekte soll ferner ein Verifikationsalgorithmus in das System integriert werden. Die für die Wissensdiagnostik verwendeten Transformationsregeln werden in visuelle Planungshilfen umgesetzt. Für 1993/94 sind empirische Untersuchungen mit den Planungshilfen, die Weiterentwicklung des Benutzermodells anhand der Ergebnisse der Modellierungsarbeiten, der Ausbau des Hilfesystems für offene Planungsprobleme und eine Evaluationsstudie geplant.

#### Zwischenbericht für den Zeitraum 1.7.90 - 31.12.90

##### Konzeption des geplanten Hilfesystems

Das vom Benutzer zu erwerbende *Planungswissen* zur Modellierung mit Petrinetzen besteht aus *Transformationswissen* (Wissen zur Überführung einer deklarativen Aufgabenstellung in Konstrukte der Zielsprache, hier Netzteile) und *Kontrollwissen* (Heuristiken zur Auswahl bei verschiedenen Transformationsmöglichkeiten). Das Transformationswissen besteht weiter aus *Operationalisierungswissen* (Wissen zur Überführung der Aufgabe in eine implementierbare Zwischenrepräsentation) und *Implementationswissen* (Wissen um die Realisation der Zwischenrepräsentation in der Zielsprache).

Der hilfegeleitete Wissenserwerbsprozeß des Benutzers soll in drei Phasen verlaufen. In der 1. Phase entwickelt der Benutzer "freie" Netzentwürfe zu gegebenen, einfachen Aufgaben. Der Benutzer kann die Entwürfe oder Ausschnitte davon vom System untersuchen lassen und erhält ggf. wissensstandsbezogene Hilfen. In dieser Phase wird der Lernende mit den Netzkonstrukten vertraut gemacht. Er/sie erwirbt implizit Transformationswissen. Das System entwickelt hierbei bereits (über das Benutzermodell) Hypothesen über dieses vom Benutzer erworbene und verwendete Wissen. In der 2. Phase erhält der Problemlöser Hilfen zur schrittweisen Transformation der gestellten Aufgabe in ein Netz. Durch dieses regelgeleitete Problemlösen können die Hypothesen über das Transformationswissen des Benutzers überprüft werden. Der Lernende erwirbt in dieser Phase explizit Transformationswissen und implizit Kontrollwissen. In der 3. Phase werden dem Problemlöser offene Planungsprobleme zur Bearbeitung vorgelegt (z.B. in Form einer fiktiven oder empirischen Zeitreihe von Beobachtungsdaten, sowie ggf. Kriterien für das zu entwerfende Netz zur Reproduktion dieser Zeitreihe). Von Seiten des Benutzers ist hier primär das heuristische Wissen in Form von Kontrollwissen zur Steuerung der Transformationsschritte gefordert. Hier können die Hypothesen über das in der 2. Phase aufgebaute Kontrollwissen des Benutzers überprüft werden.

Das Hilfesystem soll für die Unterstützung der 1. und 2. Problemlösephase aus folgenden Komponenten bestehen:

- ein *Expertenmodell*, das aus Transformationsregeln besteht und einen möglichst großen Lösungsraum aufspannt.

- ein *Benutzermodell* zur Repräsentation des jeweils aktuellen hypothetischen Wissensstands des Benutzers. Seine Bestandteile können sein: eine Teilmenge des Expertenmodells, fehlerhafte Transformationsregeln (Malrules), optimierte Regeln (Komposita) und Kontrollwissen. Das Benutzermodell wird anhand der Transformationsschritte des Benutzers kontinuierlich aktualisiert.

- eine *Verifikationskomponente*, mit der Netzentwürfe im Hinblick auf die vorliegende Aufgabe überprüft werden können.

- ein *Editor*, in dem Aufgaben präsentiert, Netze sowie Zwischenrepräsentationen konstruiert werden, und in dem Prüfhypothesen formuliert werden können.

- eine *Rückmeldungs-* und *Hilfekomponente* zur Präsentation der Systemantworten nach Prüfhypothesen.

- eine Bibliothek wissensstandsangepaßter *Transformationshilfen*.

Diese Komponenten arbeiten wie folgt zusammen: In der 1. Phase entwickelt der Benutzer mit dem Netzeditor Lösungsentwürfe für die vorgelegten Aufgaben. Der Lernende kann die Entwürfe oder Teile davon vom System untersuchen lassen. Dazu übergibt er den gewünschten Entwurfsausschnitt dem System als Prüfhypothese. Dieser Entwurfsausschnitt wird dann mit den Transformationsregeln des Benutzermodells untersucht. Wenn nötig, werden zusätzlich Regeln des Expertenmodells zur Analyse herangezogen. Kann der Entwurfsausschnitt mit diesen Regeln erkannt werden, gibt es zwei Möglichkeiten. Enthält die zum Parsen benutzte Regelmenge eine Malrule, so ist der Entwurf als inkorrekt erkannt. Andernfalls ist er korrekt.

Nun kann der Benutzer Hilfen z.B. in Form von Vervollständigungsvorschlägen von dem System anfordern. Diese Hilfen sind insoweit wissensstandsbezogen, als sie auf dem Benutzermodell beruhen. Außerdem wird das Benutzermodell aktualisiert: Die für die Analyse des Entwurfs benutzten Expertenmodell-Regeln werden in das Benutzermodell aufgenommen, und die benutzten Regeln des Benutzermodells werden durch Komposition optimiert.

Kann der Entwurfsausschnitt dagegen nicht erkannt werden, kann er mit der Verifikationskomponente weiter überprüft werden, wenn der Benutzer für seinen Entwurfsausschnitt ein Teilziel im Sinne der Aufgabenstellung angibt. Im positiven Fall wird der Entwurfsausschnitt als neues Kompositum dem Benutzermodell hinzugefügt, und es werden weitere Regeln abgeleitet, mit denen Benutzer- und Expertenmodell erweitert werden. Im negativen Fall wird der Entwurfsausschnitt als neue Malrule in das Benutzermodell aufgenommen.

In der 2. Wissenserwerbsphase transformiert der Problemlöser unter Anleitung der Planungshilfen die vom System geeignet präsentierten Aufgaben schrittweise in Netze. Die Planungshilfen sind visuelle Repräsentationen der Transformationsregeln und Komposita des Benutzermodells.

## Bisherige Vorarbeiten und Überlegungen zur Realisation des Hilfesystems

Die *Realisation des Hilfesystems* umfaßt die Entwicklung von Transformationsregeln (Expertenmodell) zur Diagnose von Lösungsentwürfen, die Entwicklung des Benutzermodells, die Implementierung der verschiedenen Oberflächenkomponenten einschließlich Rückmeldungskomponente und visueller wissensstandsbezogener Transformationshilfen und die Integration der Verifikationskomponente. Weiterhin sind für die Entwicklung wissensstandsbezogener Hilfen *empirische Untersuchungen* vorgesehen. Eine *Evaluationsstudie* und ggf. die Entwicklung einer Optimierungskomponente sollen sich anschließen.

**Expertenmodell:** Ausgangspunkt hierzu ist die in unserem Projekt ABSYNT entwickelte Diagnosekomponente, die für vorgegebene Aufgabenstellungen funktionale Programmentwürfe erkennt und generiert [Möbus, Thole, 1990; Möbus, 1991]. Sie basiert auf einer Ziel-Mittel-Relation (ZMR), die einen UND-ODER-Baum definiert. Sie läßt sich, wie [Hölldobler, Schneeberger, 1990] gezeigt haben, auch zur Lösung allgemeiner Planungsprobleme verwenden. Durch die ZMR wird das Aufgabenziel in Subziele (UND-Knoten) zerlegt, die Subziele werden weiter ausdifferenziert usw. bis auf die Ebene der funktionalen Sprachkonstrukte. Zu jeder Zieldifferenzierung bzw. -realisierung gibt es viele verschiedene Alternativen (ODER-Knoten). Für 22 ABSYNT-Programmieraufgaben können selbst bei einer auf sechs beschränkten Programmbaumhöhe mehrere Millionen Lösungen erkannt oder generiert werden.

Mit Hilfe der ZMR können Lösungen generiert (*Synthese*), Entwürfe erkannt (*Analyse*) und Teilentwürfe vervollständigt (*Hypothesentest*) werden. Diese drei Leistungen sollen auch im Hilfesystem für Petrinetzmodellierer realisiert werden. Wir haben dazu alternativ zwei Möglichkeiten untersucht:

- (a) Die Entwicklung einer ZMR für verschiedene Petrinetz-Beispiele aus der MOBY-Projektgruppe sowie aus den einschlägigen Lehrbüchern [Reisig, 1986; Baumgarten, 1990].
- (b) Die Prüfung der Eignung logischer Grammatiken und hier speziell der Definiten Klauselgrammatik (DCG) von [Pereira, Warren, 1980] zur Analyse von Petrinetzen. Es konnte gezeigt werden, daß das Hypothesenprüfen als Constraint Satisfaction Problem (CSP) unter Verwendung des DCG-Ansatzes formulierbar ist. Damit können mit der DCG Netze erkannt, generiert und Teilnetze vervollständigt werden. Ein ähnlicher Ansatz wurde unabhängig davon jüngst in [Tanaka, 1991] publiziert.

Wie sich ferner zeigte, lassen sich beide Ansätze (a) und (b) ineinander überführen.

Für das geplante Hilfesystem wollen wir diese Ansätze jedoch verallgemeinern. Wir wollen die bisherigen Ziele der ZMR bzw. die Nichtterminale der DCG durch prädikative Beschreibungen ersetzen. Damit können aufgabenübergreifende Transformationsregeln entwickelt werden, mit denen die als prädikative Beschreibungen präsentierten Aufgaben - ggf. über Zwischenrepräsentationen - in Lösungsentwürfe transformiert werden können [Olderog, 1989; Partsch, 1990].

**Benutzermodell:** Wie schon eingangs geschildert, soll das Hilfesystem für Petrinetzmodellierer mehrere Phasen des Problemlösens unterstützen:

Beim freien Problemlösen ist der Benutzer in keiner Weise eingeschränkt in Bezug auf die Konzeption seiner Lösung. Mit Hilfe der im Expertenmodell abgelegten Parsingregeln wird eine vom Schüler angebotene (Teil-) Lösung auf ihre Korrektheit überprüft. Benutzte Parsingregeln gehen in das Schülermodell ein. Wird z.B. eine Teilhypothese vom Benutzer formuliert und als korrekt ergänzbar erkannt, wird das Benutzermodell um das zum Parsen benutzte Operationalisierungswissen und Implementationswissen erweitert. Zusätzlich befähigen hieraus komponierte Regeln [Lewis, 1987] das System einerseits, zukünftige, ähnliche Benutzerlösungen schneller zu erkennen, andererseits repräsentieren sie den Lernfortschritt des Benutzers.

Ist eine angebotene Benutzerlösung nicht im Expertenmodell enthalten, wird versucht, diese zu verifizieren [Josko, 1989]. Ist der Entwurf inkorrekt, und hatte das System bis dato schon Annahmen (Defaults) über richtiges Problemlöseverhalten des Schülers getroffen, so müssen diese, einschließlich aller daraus gezogenen Schlüsse revidiert werden (belief revision [Besnard, 1989], [Huang, 1990], [Reiter, 1978]). Außerdem werden Malrules [Sleeman, 1984] angelegt. Diese Malrules sollen neben der Unterstützung der Fehlererklärung das System befähigen, ähnliche Fehler künftig schneller zu erkennen.

Beim regelgeleiteten Problemlösen erwirbt der Benutzer explizites Transformationswissen. Das System bietet ihm verschiedene ausführbare Transformationsregeln an. Die konkreten, vom Schüler getroffenen Auswahlentscheidungen gehen in das Benutzermodell ein.

Durch die Notwendigkeit, Vermutungen über den Effekt von Planungsschritten anzustellen und Entscheidungen zwischen verschiedenen Lösungswegen zu treffen, erwirbt der Benutzer außerdem implizit Kontrollwissen. Die Auswertung der getroffenen Transformationsentscheidungen (und damit des Benutzermodells) gibt Hinweise auf das Kontrollwissen. So lassen sich Präferenzen oder Aversionen des Benutzers herausarbeiten (z.B. die Vorliebe für die allgemeinste, die speziellste oder grundsätzlich die erste angebotene Regel). Bei der nächsten dargebotenen Auswahl von Regeln kann diesen Erwartungen Rechnung getragen werden.

**Oberflächenkomponenten:** Ausgangspunkt hierfür ist der in Smalltalk implementierte, auf einem Macintosh-Rechner zur Verfügung stehende MOBY-Petrinetzsimulator, auf dem u.a. Netze entwickelt werden können. Die Oberflächenkomponenten (Hypothesenbildung, Rückmeldung, Planungshilfen, Zwischenrepräsentationen bei der Transformation) sollen in Zusammenarbeit mit dem MOBY-Projekt entwickelt werden.

**Verifikationskomponente:** Hierfür wollen wir auf einen bereits vorhandenen Ansatz (z.B. den Model checking Algorithmus, [Josko 1990; Damm, Döhmen, Gerstner, Josko, 1990]) zurückgreifen.

**Empirische Untersuchungen zur Entwicklung wissensstandsbezogener Hilfen:** Um die Fortentwicklung eines Novizen zu einem Experten in einer Domäne möglichst effektiv unterstützen zu können, sind detaillierte Hypothesen und Daten über diesen Wissenserwerbsprozeß erforderlich.

Wissenserwerbsprozesse können u.E. als Wechselspiel aus *impasse-gesteuertem Lernen (IDL)* und *erfolgsgesteuertem Lernen (SDL)* beschrieben werden. Demnach ist der Lernende bevorzugt in Impasse- oder Stocksituationen zur Informationsaufnahme oder zu aktiver Informationssuche bereit. Als Ergebnis wird neues Wissen erworben und die Stocksituation überwunden (IDL). Wird dagegen bereits erworbenes Wissen erfolgreich genutzt, so wird es z.B. im Sinne von Regelkomposition optimiert (SDL). Nach der IDL-SDL-Theorie ist als Hilfe gedachte Information nur dann hilfreich, wenn

- der Lernende sich in einer *Stocksituation* befindet
- die Information *wissensstandsangepasst*, d.h. weder trivial noch unverständlich ist
- die Information dem gegenwärtigen *Problemlösekontext* entspricht, d.h. mit der vom Lernenden gegenwärtig verwendeten Kontroll-/Problemlösestrategie verträglich ist.

Die Bereitstellung solcher Hilfen setzt also die *online-Diagnose* von Stocksituationen, von aktuellem Wissensstand und von der aktuell benutzten Kontrollstrategie voraus. Zur Gewinnung differenzierter Hypothesen sowie von rechnerseitig registrierbaren Indikatoren für diese Prozesse sind *Wissenserwerbsmodelle* geplant, mit denen wir real abgelaufene Wissenserwerbsprozesse detailliert retrognostizieren und in weiteren empirischen Untersuchungen überprüfen wollen. Dabei soll u.a. vorhergesagt werden, in welchen Problemlösesituationen der Lernende auf welche Hilfen zugreift. Im Zusammenhang mit IDL sind auch emotionale Komponenten zu berücksichtigen, wie die Entstehung von Ärger, Streß oder Befriedigung und ihre Rückwirkung auf den Planungsprozeß [z.B. Spies, Hesse, 1986]. Die gewonnenen Hypothesen und Indikatoren werden dann zur *Verbesserung des Benutzermodells* und der *Hilfengenerierung* eingesetzt. Vorläufer solcher Wissenserwerbsmodelle haben wir bereits im Projekt ABSYNT entwickelt.

**Evaluation des Hilfesystems:** Wir wollen das Hilfesystem unter zwei Gesichtspunkten empirisch evaluieren:

1. Sind *Petrinetze* geeignete Planungswerkzeuge für Modellierer offener Planungsprobleme? Wieweit sie von den Benutzern akzeptiert? Für welche Einsatzbereiche? Fällt es Novizen leicht, sich einzuarbeiten? Gibt es diesbezüglich Unterschiede beim freien vs. regelgeleiteten Problemlösen? Diese Fragen sollen entweder durch den Einsatz des Hilfesystems im Feld oder durch einen experimentellen Vergleich mit anderen Planungswerkzeugen (SADT, SIMSCRIPT/SIMFACTORY, ...) untersucht werden.
2. Welche Bedeutung haben *adaptive, wissensstandsbezogene* vs. *nichtadaptive* Hilfen in Bezug auf Lernfortschritt und Akzeptanz durch den Benutzer? Auch hier wollen wir detaillierte Vorhersagen machen: Werden z.B. die Hilfen, die mit dem aktuellen Wissensstand übereinstimmen, häufiger benutzt als andere? Werden die Hilfen benutzt, die mit der aktuellen Kontrollstrategie übereinstimmen? Bewirken nichtadaptive Hilfen einen Strategienwechsel, wie z.B. vermehrte Analogienutzung?

**Optimierungskomponente:** Bei der Optimierungskomponente wird daran gedacht, den Benutzer zu *zielgerichteterem, effizienterem Problemlösen* anzuleiten. Dabei wird nach einer korrekten,

regelgeleitet ausgeführten Problemlösung der Weg des Benutzers über die gewählten Zwischenzustände zum Ziel untersucht. Gibt es eine Regel, die mehrere Zwischenzustände überspringt, wird dem Schüler diese Regel zusammen mit der von ihm gewählten, bedeutungsgleichen Sequenz vorgestellt. Er kann dann entscheiden, ob diese "Abkürzung" für ihn erstrebenswerte Vorteile bringt und er sein Problemlöseverhalten in diesem Punkt in Zukunft verändern wird.

#### Literatur:

- Baumgarten, B., Petri-Netze - Grundlagen und Anwendungen. BI Wissenschaftsverlag, 1990
- Besnard, Ph., An Introduction to Default Logic. Berlin: Springer 1989
- Damm, W., Döhmen, G., Gerstner, V., Josko, B., Modular Verification of Petri Nets. The Temporal Logic Approach. In de Bakker, J.W., de Roever, W.P., Rozenberg (eds), Proceedings REX-Workshop on Stepwise Refinement of Distributed Systems: Models, Formalisms, Correctness. Springer: Lecture Notes in Computer Science 430, 1990
- Hölldobler, S., Schneeberger, J., A New Deductive Approach to Planning. New Generation Computing, 8, 1990, 225-244
- Huang, X., Student Model Revision: Evolution and Revolution. ARIES Laboratory, Dept. of Computer Science, University of Saskatchewan, Saskatoon, Canada, 1990
- Josko, B., Verifying the Correctness of AADL Modules using Model Checking. In de Bakker, J.W., de Roever, W.P., Rozenberg (eds), Proceedings REX-Workshop on Stepwise Refinement of Distributed Systems: Models, Formalisms, Correctness. Springer: Lecture Notes in Computer Science 430, 1990, 387-400
- Lewis, C., Composition of Productions. In: Klahr, D., Langley, P., Neches, R. (eds), Production System Models of Learning and Development. Cambridge: MIT Press, 1987, 329-358
- Möbus, C., The Relevance of Computational Models of Knowledge Acquisition for the Design of Helps in the Problem Solving Monitor ABSYNT. In: Lewis, R., Otsuki, S. (eds), Advanced Research on Computers in Education. Elsevier North Holland, 1991, 137-144
- Möbus, C., Thole, H.-J., Interactive Support for Planning Visual Programs in the Problem Solving Monitor ABSYNT: Giving Feedback to User Hypotheses on the Language Level. In: Norrie, D.H., Six, H.W., Computer Assisted Learning. Proceedings of the 3rd ICCAL, 1990, 36-49
- Olderog, E.-R., Nets, Terms, and Formulas: Three Views of Concurrent Processes and Their Relationship. Universität Oldenburg, FB Informatik, 1989
- Partsch, H.A., Specification and Transformation of Programs: A Formal Approach to Software Development. Berlin, Springer 1990
- Pereira, F.C.N., Warren, D.H.D., Definite Clause Grammars for Language Analysis - A Survey of the Formalism and a Comparison with Transition Networks. Artificial Intelligence, 13, 1980, 231-278
- Reisig, W., Petrinetze - eine Einführung. Berlin: Springer, 1986
- Reiter, R., On Reasoning by Default. Proc. on Theoretical Issues in Natural Language Processing. Urbana, 1978
- Schröder, O., Kohnert, K., Toward a Model of Instruction-Based Knowledge Acquisition: The Operational Knowledge for a Functional, Visual Programming Language. Journal of Artificial Intelligence in Education, 1, 1989/90, 105-128
- Sleeman, D., An Attempt to Understand Students' Understanding of Basic Algebra. Cognitive Science, 8, 1984, 387-412
- Spies, K., Hesse, F.W., Interaktion von Emotion und Kognition. Psychologische Rundschau, 37, 1986, 75-90
- Tanaka, T., Definite-Clause Set Grammars: A Formalism for Problem Solving, Journal of Logic Programming, 1991, 10, 1-17
- van Lehn, K., Toward a theory of Impasse-Driven Learning. In: Mandl, H., Lesgold, A. (eds), Learning Issues for Intelligent Tutoring Systems. New York, Springer, 1988, 19-41
- van Lehn, K., Mind Bugs: The Origins of Procedural Misconceptions. MIT Press, 1990
- Wolff, J.G., Cognitive Development as Optimization. In: Bolc, L. (ed), Computational Models of Learning. Berlin, Springer, 1987, 161-205