

PULSE¹ - Eine Intelligente Problemlöseumgebung im Bereich der Pneumatik

Hermann Göhler, Claus Möbus, Janine Willms

Universität Oldenburg
Fachbereich Informatik
Abteilung Lehr- Lernsysteme
Postfach 2503
D 26111 Oldenburg

eMail: claus.moebus@informatik.uni-oldenburg.de

Zusammenfassung

PULSE (pneumatic learning and simulation environment) ist eine wissensbasierte Design- und Problemlöseumgebung, welche die Konstruktion pneumatischer Schaltungen unterstützt. In Zusammenarbeit mit der DIHT-Gesellschaft für Berufliche Bildung, Organisation zur Förderung der IHK-Weiterbildung mbH (*Deutscher Industrie- und Handelstag in Bonn*) sowie den regionalen Industrie- und Handelskammern wurde PULSE basierend auf der Konzeption von PETRI HELP entwickelt.

Die Problemlöseumgebung PULSE soll in der Ausbildung zum Industriemeister "Metall" eingesetzt werden. Die Auszubildenden müssen hier u.a. eine Sequenz von 30 Aufgaben der *Prüfungs-, Aufgaben- und Lernmittelstelle* (PAL) lösen.

Die theoretische Grundlage bildet der ISPD-ANSATZ (s.a. Möbus et al., 1993, *impassé success problem solving-driven learning*), aus dem sich folgende Designprinzipien für Problemlöseumgebungen (IPSE = *Intelligent Problem Solving Environment*, Möbus 1995) ableiten lassen: Formulierung und Überprüfung von Hypothesen, situationsbezogene Erklärungen als Feedback und freies Problemlösen. PULSE orientiert sich an diesen Kriterien.

Auszubildende können in PULSE mit einem CAD-Editor pneumatische Schaltungen *frei konstruieren*. In jeder Phase des Entwerfens ist es ihnen dabei möglich, ihren Lösungsentwurf als Hypothese an das System zu formulieren. Das System untersucht diese Hypothese und gibt dem Auszubildenden differenzierte Fehlerrückmeldungen und Erklärungen.

Die kognitionstheoretische Grundlage von IPSE's: Der ISPD-ANSATZ

Aufgrund eigener empirischer Untersuchungen nehmen wir an, daß es nützlich ist, Lernen als ein Zusammenspiel von Lernen durch Stocksituationen und Lernen durch Erfolg (s.a. Möbus et al., 1993), zu sehen. Dazu entwickelten wir ein Phasenmodell, welches den Strom von Aktionen und Verbalisationen eines Probanden während einer Problemlösesequenz differenziert beschreibt (Möbus, Schröder & Thole, 1994; Möbus 1995).

Zur Beschreibung kognitiver Prozesse bei Problemlösen und Wissenserwerb sieht der ISPD-ANSATZ drei Aspekte vor:

- die Unterscheidung bestimmter Problemlösephasen
- den stocksituationengesteuerten Erwerb von neuem Wissen
- die Optimierung vorhanden Wissens durch Erfolgssituationen

Erfolgsituationen führen zum effektiveren Einsatz bereits vorhandenen Wissens. Der Prozeß der Optimierung vorhandenen Wissens ist *deduktiv*, wenn angenommen wird, daß dieses *optimierte Wissen* die logische Konsequenz von *altem Wissen* (Hintergrundwissen) sowie Evidenz (Beobachtungen und Erfahrungen) ist:

Hintergrundwissen U Evidenz | = Optimiertes Wissen

Der Erwerb neuen Wissens tritt auf, wenn Lösungen durch den Einsatz schwacher Heuristiken gefunden wurden. Schwache Heuristiken werden vom Problemlöser als Reaktion auf Stocksituationen eingesetzt. Der Erwerb neuen Wissens ist induktiv:

Hintergrundwissen U Neues Wissen | = Evidenz

¹ gefördert von der Otto Wolff von Amerongen Stiftung, der Stiftung Volkswagenwerk und dem Institut OFFIS

Der Erwerb neuen Wissens nach erfolgreichem Heuristikeinsatz kann dabei mit obiger induktiver Inferenzregel beschrieben werden.

Aus dem ISPDL-Ansatz lassen sich folgende Kriterien für das Design von IPSE's ableiten:

- [1] "Unterbrich den Lernenden nie - biete statt dessen Hilfe an."
Dieses Prinzip stellt einen Kontrast zu Systemen mit aktiver Hilfe und sofortigem Feedback dar. Wir nehmen an, daß es sinnvoller ist, daß der Lernende seine eigenen Lösungsideen zunächst ungestört entwickelt. Der Lernende wird in Stocksituationen von sich aus als eine Form schwacher Heuristik Hilfen anfordern.
- [2] "Stelle detaillierte Informationen für die Problemlösephasen bereit, in denen Stocksituationen auftreten können: *planen*, *ausführen* und *bewerten*". Diese Forderung ist notwendig, um dem Lernenden jederzeit phasen-spezifisches detailliertes Feedback und Informationen geben zu können.
- [3] "Der Lernende soll solange wie möglich sein Vorwissen anwenden können, bis er nach Hilfe fragt."
Der Lernende soll frei explorativ arbeiten und sein Vorwissen möglichst weitgehend einsetzen können.
- [4] "Biete Informationen an, die den Wissenstand des Lernenden berücksichtigen."
Insbesondere wichtig ist dabei die Körnung der Information, denn eine zu feine oder zu grobe Körnung der Information oder nicht mit dem Wissensdefizit synchronisierte Information kann dazu führen, daß der Lerner neue Information filtern oder generieren muß, was unerwünschte Effekte auf den Lernfortschritt des Lerners haben kann (sekundäre Stocksituationen).

In unseren Systemen sind diese Anforderungen durch den Hypothesentestansatz realisiert: Der Lernende formuliert zu frei entwickelten Entwürfen Prüfungshypothesen. Das System untersucht diese Hypothesen und gibt auf Anforderung Rückmeldungen, Ergänzungs- und Korrekturvorschläge sowie bei weiterem Bedarf Erklärungen.

PETRI-HELP

PETRI-HELP ist eine Problemlöseumgebung, die den Benutzer beim Erlernen und bei der Modellierung mit Bedingungs-Ereignis Petri-Netzen unterstützt. Während der Problemlösung kann der Lernende seine Lösungsvorschläge vom System überprüfen lassen und Ergänzungsvorschläge anfordern. Die Bewertung von Lösungsentwürfen setzt sowohl eine formale Spezifikation der zu bearbeitenden Aufgabe als auch ein Orakel voraus, das den Lösungsentwurf bezüglich dieser Spezifikation bewertet. Eine Aufgabe wird dabei durch temporallogische Formeln spezifiziert.

Diese Form der Spezifikation erlaubt es, Lösungsentwürfe oder Teile davon gegen die Spezifikation mittels Modelchecking (Damm et. al., 1990) zu prüfen. Als Rückmeldung des Systems werden erfüllte oder nicht erfüllte temporallogische Formeln der Spezifikation zurückgegeben.

Das derzeitige System beinhaltet die folgenden Komponenten:

- Eine *Sequenz von Modellierungsaufgaben* mit aufsteigendem Schwierigkeitsgrad. Jede Aufgabe ist durch *temporallogische Formeln* spezifiziert. Lösungsentwürfe werden durch einen Modelchecker geprüft. Die zugrundeliegende Petrinetzverifikation wurde bereits für den Einsatz von Petri-Netzen zur Prüfung von Hardwarebeschreibungen (Josko 1990) eingesetzt.
- Ein *Netzeditor* erlaubt die Konstruktion und die schrittweise Simulation von Petrinetzen.

- Eine *Hypothesentestumgebung* gibt Informationen über den Stand erfüllter und nicht erfüllter Formeln eines Lösungsentwurfes. Die Rückmeldung erfolgt durch als richtig oder falsch markierte Formeln.
- Eine *Ergänzungs- und Korrekturkomponente*, welche auf Nachfrage mitteilt, wie eine korrekte Lösung ausgehend von dem vorhandenem Petri-Netz konstruiert werden kann. Diese Komponente basiert auf Regeln, die vom System anhand der Lösungen anderer Benutzer gelernt werden.

Überblick: PETRI HELP vs. PULSE

Inwieweit ist das PULSE-System mit PETRI HELP vergleichbar ?

Kriterium	PETRI-HELP	PULSE
Aufgaben-spezifikation	- temporallogische Formeln beschreiben das gewünschte Verhalten des zu konstruierenden BE-Netzes	- Funktionsdiagramme, die sich <i>diskret</i> durch temporallogische Formeln beschreiben lassen - textuelle Aufgabenbeschreibungen, denen weitere Lösungsinformationen entnommen werden können
Verifikations-mechanismus	- Erzeugung eines Fallgraphen - Prüfung von temporallogischen Formeln auf dem Fallgraphen mittels eines Modelcheckers	- Erzeugung eines Fallgraphen - Prüfung von temporallogischen Formeln auf dem Fallgraphen mittels eines Modelcheckers
Hypothesen formulierung	- markierte Spezifikationsteile (Formelmengen) dienen als Hypothese	- markierte Spezifikationsteile (Bereiche des Funktionsdiagramms) dienen als Hypothese
Systemrück-meldungen	- es werden erfüllte oder nicht erfüllte Formeln zurückgegeben	- erfüllte Bereiche der Markierung im Funktionsdiagramm werden als „richtig“ und nicht erfüllte als „falsch“ zurückgegeben
Ergänzungs- und Korrekturvorschläge	- mittels einer wissensbasierten Lernkomponente werden Ergänzungs- und Korrekturhinweise gegeben	- derzeit in der Entwicklung
Konzeptbasierte Diagnose / Planungsziele	- zu realisierende Ziele sind durch Konzeptnamen ausgedrückt	- derzeit in der Entwicklung (s. Ausblick)
Simulation	- Simulation des Schaltnetzverhaltens - Simulation bis zum Auftreten einer Fehlersituation	- Simulation des Verhaltens der Pneumatik-schaltung wird gegenwärtig entwickelt
Erklärungen	- textuelle Erklärung mit Bezug zur temporallogischen Spezifikation	- Konzeptbasierte Erklärungen sind in der Entwicklung

Abb 1: Vergleich PETRI HELP vs. PULSE

PULSE

PULSE ist eine Problemlöseumgebung, die den Benutzer beim Erlernen und bei der Konstruktion und Modellierung *pneumatischer Schaltungen* unterstützt. Schon während des Problemlösens kann der Lernende seine Lösungsvorschläge vom System überprüfen lassen. Die Bewertung von Lösungsentwürfen setzt wie auch in PETRI HELP eine formale Spezifikation voraus. Während in PETRI-HELP temporallogische Formeln das gewünschte Verhalten eines Netzes spezifizieren, müssen vorhandene Spezifikationen pneumatischer Schaltungen in PULSE erst diskretisiert werden. Eine gegebene quantitative Spezifikation (z.B. Diagramm mit exakten Zeiten) muß erst in eine abstrakte, qualitative Spezifikation umgewandelt werden, welche dann intern durch temporallogische Formeln repräsentiert werden kann.

Das derzeitige System beinhaltet die folgenden Komponenten:

- Eine *Sequenz von Pneumatik- Aufgaben* mit aufsteigendem Schwierigkeitsgrad. Aufgaben sind durch *textuelle Beschreibungen* und *Funktionsdiagramme* spezifiziert, die sich in *temporallogische Formeln* übersetzen lassen. Ein Funktionsdiagramm beschreibt dabei nur das Verhalten von elementaren Aktoren (z.B. Zylinder), Stellgliedern und Schaltern oder Tastern (z.B. manuell betätigte Ventile). Lösungsentwürfe werden durch einen Modelchecker geprüft.
- Ein *CAD-Editor* erlaubt die Konstruktion und Simulation von Pneumatikschaltungen. Wichtige CAD-Funktionen wie Verschieben, Ausrichten, Gummibandeffekt usw. stehen dabei als Werkzeuge zur Verfügung.
- Eine *Hypothesentestumgebung* zeigt an, an welcher Stelle ein Lösungsentwurf das im Funktionsdiagramm dargestellte gewünschte Verhalten einer Pneumatikschaltung verletzt und wo korrektes Verhalten vorliegt.
- *Hilfen und Heuristiken* sind derzeit durch Dialoge realisiert. Eine konzeptbasierte Hilfe ist in der Entwicklung (s.a. Ausblick).

Hypothesentesten in PULSE

Während der Konstruktion einer Schaltung kann der Benutzer zu jedem Zeitpunkt Hypothesen darüber formulieren, daß sein (Teil-) Entwurf im Hinblick auf die vorgegebene Aufgabenstellung korrekt ist. Dazu kann er im Funktionsdiagramm diejenigen Schaltungsverläufe markieren, von denen er annimmt, daß seine (Teil-) Schaltung das durch sie spezifizierte Verhalten erfüllt. Das System überprüft nun die Hypothese. Wenn beispielsweise aus dem Funktionsdiagramm die Verlaufskurve eines bestimmten Bauglieds ausgewählt wurde, dann wird das Verhalten dieses Bauglieds über alle Zeitpunkte geprüft.

Links in Abb. 2 hat der Benutzer aus einer Druckquelle, einem 3/2-Wegeventil und einem einfachwirkenden Zylinder einen Lösungsentwurf für die Aufgabe "Hebebühne" konstruiert. Anschließend formuliert er eine Hypothese über das gesamte Funktionsdiagramm, also die Hypothese, daß sich diese Schaltung so verhält wie vom Funktionsdiagramm gefordert (Abb. 2, rechts).

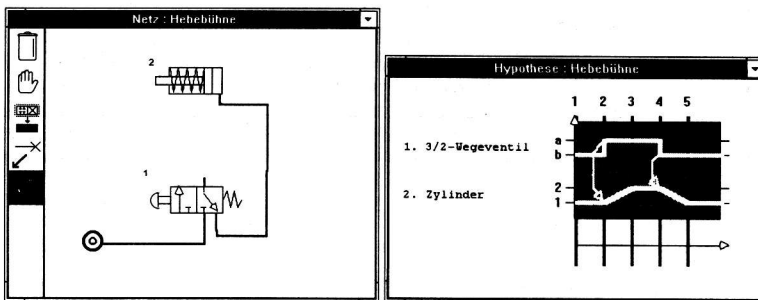


Abb. 2: Schaltbild (links) und Hypothese (rechts)

Wissensbasierte Diagnosekomponente

Das System überprüft die Hypothese des Anwenders mit einer *Diagnosekomponente*. Um beliebige Entwürfe des Anwenders überprüfen zu können, muß die Diagnosekomponente über Wissen aus der Domäne der Pneumatik verfügen. Das dazu erforderliche Grundlagenwissen über das dynamische und statische Verhalten der Pneumatik-Bauelemente und deren Vernetzung ist in die Diagnosekomponente integriert.

Wenn die Korrektheit eines Entwurfs überprüft wird, berechnet das System das komplette mögliche Verhalten der Schaltung in Form eines *Fallgraphen* und vergleicht dessen Struktur mit den Anforderungen der Spezifikation, also des Funktionsdiagramms.

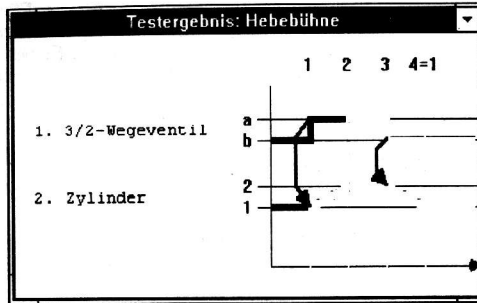


Abb. 3: Ergebnis des Hypothesentestens: Der Zylinder fährt nicht aus.

Das Ergebnis der Hypothesenprüfung wird zurückgemeldet wie in Abb. 3 dargestellt. Grün markierte Schritte bedeuten, daß alle geprüften Bauteile sich in diesen Schritten der Spezifikation entsprechend verhalten, rot markierte (in Abbildung heller dargestellt) sind zum Teil oder komplett inkorrekt. Entsprechendes gilt für die einzelnen Kurvenssegmente für jedes Bauteil. Abb. 3 zeigt die Rückmeldung für den fehlerhaften Lösungsentwurf der Aufgabe "Hebebühne" (vgl. Abb. 2, links).

Ausblick

PULSE wird gegenwärtig in folgenden Richtungen weiterentwickelt:

- *Hypothesen über den Entwürfen.* Gegenwärtig werden Prüfhypothesen lediglich über den Funktionsdiagrammen formuliert. In der Zukunft soll es auch möglich sein, gezielt Teile des Entwurfs auf Korrektheit testen zu können (Netzhypothesen).

- *Erklärungen:* Der Anwender soll zu fehlerhaften Entwürfen (wie in Abb. 2, links) Erklärungen sowie *Ergänzungs- und Korrekturvorschläge* erhalten können. Diese Erklärungen werden auf der Grundlage des in der wissensbasierten Diagnosekomponente enthaltenen Wissens über pneumatische Schaltungen gebildet. Eine konzeptbasierte Erklärungs-komponente ist in der Entwicklung.

Konzepte liegen dabei hierarchisch organisiert als *Objekt- oder Funktionskonzepte* vor. Eine Komponente des Systems untersucht dazu das Funktionsdiagramm und die textuelle Beschreibung auf zu realisierende Konzepte (*Spezifikationsanalyse*). Mittels einer Analyse des vorliegenden Lösungsentwurfs (*Netzanalyse*) werden Fehler in der Wahl von Bauteilen und dem Aufbau einer Schaltung erkannt. Eine *Synthesekomponente* soll durch prototypische Instantiierung fehlender Konzepte und Korrektur von Fehlern eine *Referenzlösung* generieren, die es ermöglicht, dem Benutzer *Ergänzungs- und Korrekturvorschläge* anzubieten. Die Korrektheit der Referenzlösung läßt sich dabei durch Modelchecking validieren.

Ein exemplarisches Beispiel für eine konzeptbasierte Erklärung ist in Abb. 4 dargestellt.

Bei der links in Abb. 4 zu sehenden Pneumatikschaltung fehlt das Konzept „Aktivierung des Zylinders 2 durch das Ventil 1“. Wie rechts in Abb. 4 zu sehen, ist die *Kausalität* der Aktivierung, die durch den umrandeten Pfeil dargestellt wird, nicht erfüllt. Ein konzeptbasierter *Ergänzungsvorschlag* könnte jetzt lauten: „Verbinden Sie den Anschluß 2 des Ventils 1 mit dem Anschluß 1 des Zylinders 2.“ Die auf diese Art konstruierte Lösung ist korrekt.

- *Differenziertere Fehlerrückmeldungen.* Bisher werden nur Fehler in der Dynamik der Schaltung erkannt und zurückgemeldet. In der textuellen Beschreibung und im Funktionsdiagramm sind jedoch auch Vorschriften zur Auswahl bestimmter Bauglieder enthalten. Die Extrahierung solcher Vorschriften aus der Aufgabenspezifikation mittels

Heuristiken kann konzeptbasiert geschehen und ermöglicht auch Fehlerrückmeldungen bezüglich der Auswahl bestimmter Bauglieder.

- Eine interaktive sowie automatische Simulationsumgebung zur Visualisierung des Verhaltens von Schaltungen (diskrete Simulation).
- Komponente zur Spezifikation von Aufgaben als Dozentenunterstützung. Für den Einsatz des Systems im Rahmen von Kursen und Schulungen soll der Dozent selbst die Aufgabensammlung durch Spezifikation von Aufgabentexten und Funktionsdiagrammen erweitern können. Der Entwurf einer neuen Aufgabe soll dabei durch direkte Manipulation der Einträge im Funktionsdiagramm geschehen bzw. durch einen Satzeditor erleichtert werden.

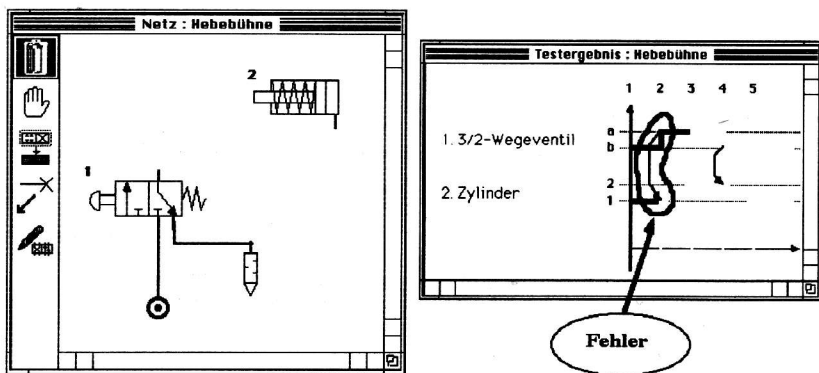


Abb 4: Fehlendes Konzept „Aktivierung des Zylinder durch das Ventil“

Literaturverzeichnis

- Damm, W., Döhmen, G., Gerstner, V., Josko, B. . Modular verification of Petri nets, The temporal logic approach, in J.W. de Bakker, W.P. de Roever, G.Rozenberg (eds), Proceedings REX-Workshop on stepwise refinement of distributed systems: models, formalisms, correctness., Springer, Verlag, Berlin 1990, LNCS 430
- Josko, B., Verifying the correctness of AADL modules using model checking. In J.W. de Bakker, W.P. de Roever, G. Rozenberg (eds): Proceedings REX-Workshop on stepwise refinement of distributed systems: models, formalisms, correctness., Springer Verlag, Berlin 1990, LNCS 430, 386-400.
- Möbus, C., Pitschke, K., Schröder, O., Towards the Theory-Guided Design of Help Systems for Programming and Modelling Tasks, in C. Frasson, Gauthier, G., G.I. McCalla (eds), Intelligent Tutoring Systems, Proceedings of the Second International Conference ITS 92, Montreal, Berlin: Springer (LNCS 608), 1992, S. 294 - 301
- Möbus, C., Pitschke, K., Schröder, O., Folckers, J., Göhler, H., Intelligent Support for Modelling Time-Discrete and Distributed Systems: PETRI-HELP, Universität Oldenburg 1993
- Möbus, C., Towards an Epistemology of Intelligent Problem Solving Environments: The Hypothesis Testing Approach, Proceedings of AI-ED, Washington, DC, 1995, S. 138-145
- Pitschke, K., Schröder, O., Möbus, C., Erklärungsgenerierung mit Petri-Help, in F. Huber-Wäschle, H. Schauer, P. Widmayer (Hg.), GISI95, 25. GI-Jahrestagung und 3. Schweizer Informatikertag, Proceedings, Berlin: Springer, 1995, S. 304-313

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

10. Kolloquium der Arbeitsgruppe
Informatik-Systeme

Michael Sonnenschein,
Hans Fleischhack (Hrsg.):

Bericht Nr. AIS-24- Juli 1995

Arbeitsgruppe Informatik-Systeme

FB Informatik - Universität Oldenburg

Vorwort

Der vorliegende Bericht stellt die Ergebnisse dar, die beim zehnten Kolloquium der Arbeitsgruppe Informatik-Systeme am 24. 11. 1995 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