

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/228742007>

# Die Integration von dynamischen und statischen Analysemethoden in einer intelligenten Lern- und Problemlöseumgebung

Conference Paper · September 1997

DOI: 10.13140/RG.2.1.3052.9684

CITATIONS

2

READS

41

3 authors, including:



Claus Möbus

Carl von Ossietzky Universität Oldenburg

209 PUBLICATIONS 500 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



modelling with probabilistic programming languages (WebPPL, OpenBugs, etc) [View project](#)

# **Die Integration von dynamischen und statischen Analysemethoden in einer intelligenten Lern- und Problemlöseumgebung**

Janine Willms, Hermann Göhler, Claus Möbus

Fachbereich Informatik, Abteilung Lehr- und Lernsysteme,  
C.v.O. Universität Oldenburg, D-26111 Oldenburg, und  
OFFIS Institut, D-26121 Oldenburg,

E-Mail: {Janine.Willms,Hermann.Goehler,Claus.Moebus}@informatik.uni-oldenburg.de

Der Hypothesentest-Ansatz ist ein zentraler Aspekt der ISP-DL-Theorie. Er ermöglicht individuelles Feedback auf Basis des Lösungsentwurfes des Lernenden, ohne ein explizites Schülermodell aufbauen zu müssen. Die Umsetzung dieser kognitionswissenschaftlichen Theorie in reale Problemlöseumgebungen (IPSE = Intelligent Problem Solving Environment) wurde bereits früher an Beispielen unterschiedlichster Domänen gezeigt (Möbus 1995, Möbus 1996). Dabei wurde bisher die Prüfung der dynamischen Aspekte eines Systems in den Vordergrund gestellt, da diese eine vollständige Methode zum Testen beliebiger Schülerlösungen bietet. Ein System, das auf dynamischem Model-Checking beruht, kann dadurch zu jedem Designentwurf eines Schülers korrektes Feedback geben. Allerdings bietet das dynamische Model-Checking sehr fein granulierte Basis-Informationen, die in frühen Entwicklungsphasen des Modells, d.h. Abwäge- und Planungsphasen, nicht sinnvoll genutzt werden können. Eine zweite, auf der Statik eines Entwurfs beruhende, konzeptbasierte Analyse soll helfen, den Lernenden in diesen frühen Entwicklungsphasen zu unterstützen. In diesem Beitrag wird PULSE vorgestellt, eine IPSE zur Umsetzung pneumatischer Schaltpläne aus vorgegebenen Spezifikationen, in der beide Analysemethoden realisiert wurden, so daß die Vorteile beider Methoden nutzbar sind.

## 1 Statische and Dynamische Analyse

Es gibt verschiedene Arten, einen Lernenden durch ein computergestütztes System zu unterstützen. Unabhängig davon, ob diese nun intelligentes Tutorssystem, explorative Lernumgebung oder intelligente Designumgebung genannt werden, ist die Analyse des Entwurfs eines Schülers essentiell, um dessen Ziele zu erkennen und Fehler zu beseitigen. Systeme wie SPADE (Miller & Goldstein, 1976) oder PROUST (Johnson, 1986) verwenden eine statische, planbasierte Analyse mit korrekten Plänen und Fehlerplänen, um den Entwurf des Lernenden zu untersuchen und zu bewerten. Korrektur- und Ergänzungsvorschläge werden auf Basis der dem System bekannten Pläne generiert. Andere Systeme, wie z.B. das System PETRI-HELP (Möbus, Pitschke, Schröder, 1992), bewerten den dynamischen Aspekt eines Entwurfes. Nur selten wird die Kombination von dynamischer und statische Analyse zur Bewertung von Schülerlösungen herangezogen. In dem intelligenten Tutorssystem SYPROS (Henning, 1994; Herzog, 1996) wird der statische Ansatz momentan durch eine dynamische Analysekomponente erweitert. Aus unseren Erfahrungen mit PETRI-HELP wissen wir die Vorzüge einer dynamischen Analyse, nämlich deren Vollständigkeit und Korrektheit, zu schätzen. Vollständigkeit bedeutet hierbei, daß der gesamte Problemraum durch die Analysemethode abgedeckt wird. Korrektheit bezieht sich auf das Ergebnis der Analyse. Das dynamische Model-Checking liefert immer korrektes Feedback. Für frühe Problemlösephasen, wie z.B. der Abwäge- oder Planungsphase sind die Rückmeldungen einer dynamischen Analyse jedoch nur eingeschränkt geeignet. Teilentwürfe können oft nicht auf Basis einer dynamischen Analyse getestet werden. Auch sind gerade in frühen Entwurfsphasen höhere Konzepte notwendig, um einen generellen Entwurfsablauf zu planen. Die fein granulierten Rückmeldungen einer dynamischen Analyse sind in dieser Planungsphase auf einer zu niedrigen Entwurfsebene. Das System PULSE vereint beide Analyse-Methoden. Es handelt sich bei PULSE um eine IPSE (Möbus, 1996), die auf der ISP-DL-Theorie ("Impasse-Success-Problem-Solving-Driven-Learning", Möbus, 1995, 1996), einer kognitiven Theorie des Wissenserwerbs beruht.

## 2 Die ISP-DL-Theorie

Die ISP-DL-Theorie verbindet verschiedene Formen menschlicher Wissensakquisition zu einer zusammenhängenden kognitiven Theorie. Dabei wird von einem induktiven Erwerb neuen Wissens nach Überwindung einer Stocksituation (Impasse) und einer deduktiven Optimierung bisher bereits vorhandenen Wissens nach der erfolgreichen Lösung eines Problems ausgegangen. Der Problemlöseprozeß umfaßt vier verschiedene Phasen: die Abwäge-, Planungs-, Aktions- und Bewertungsphase. Um intelligente Problemlöseumgebungen oder intelligente Designumgebungen zu realisieren, die effektiv 'Learning by doing' und die Akquisition neuen Wissens *während des Problemlösens* unterstützen, wurden aus der ISP-DL-Theorie Design-Kriterien für solche Systeme abgeleitet:

- Das System sollte einen Lernenden niemals unterbrechen. Neues Wissen wird nur akzeptiert, wenn sich der Schüler in einer selbstverschuldeten Stocksituation befindet. Eine Unterbrechung des Problemlöseprozesses durch das System würde störend wirken.
- Der Lernende soll sein Vorwissen in den Problemlöseprozeß so weit als möglich einbringen können. Das System muß daher in der Lage sein, die Aktionen des Schülers zu interpretieren und zielgerichtete Hilfen und Erklärungen anzubieten. Diese sollten auf den bisherigen Entwurf des Lernenden abgestimmt sein.

- Der Problemlöseprozeß des Lernenden sollte nicht durch das System eingeschränkt werden. Aktives Lernen, uneingeschränktes Arbeiten und exploratives Erforschen der Domäne sollte dem Lernenden ermöglicht werden.
- Das System sollte auf Anfrage passende Informationen und Hilfestellungen in Form von Korrektur- und Ergänzungsvorschlägen liefern können. Diese sollten **in jeder Phase** des Problemlöseprozesses sinnvoll nutzbar sein.

Insbesondere die letzte Anforderung wird erst durch die Verbindung von dynamischer und statischer Analysemechanismen realisiert.

### 3 Die intelligente Problemlöseumgebung PULSE

PULSE (pneumatic learning and simulation environment) ist eine wissensbasierte Design- und Problemlöseumgebung, welche die Konstruktion pneumatischer Schaltungen unterstützt. PULSE wurde 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 entwickelt. Die Problemlöseumgebung PULSE soll in der Ausbildung zum Industriemeister "Metall" eingesetzt werden. Den Auszubildenden steht eine Sequenz von 30 Aufgaben der Prüfungs-, Aufgaben- und Lernmittelstelle (PAL) zur Verfügung. Sie 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.

PULSE beinhaltet die folgenden Komponenten:

- Eine *Sequenz von Pneumatik-Aufgaben* mit aufsteigendem Schwierigkeitsgrad. Aufgaben sind durch textuelle Beschreibungen und Funktionsdiagramme spezifiziert. Ein Funktionsdiagramm beschreibt dabei nur das Verhalten von elementaren Aktoren (z.B. Zylinder), Stellgliedern und Schaltern oder Tastern (z.B. manuell betätigte Ventile).
- Ein *CAD-Editor* erlaubt die Konstruktion und Simulation von Pneumatikschaltungen. Wichtige CAD-Funktionen wie Verschieben, Ausrichten, Gummibandeffekt usw. stehen dabei als Werkzeuge zur Verfügung.
- Lösungsentwürfe werden durch einen *dynamischen Modelchecker* geprüft. Er zeigt an, an welcher Stelle ein Lösungsentwurf das im Funktionsdiagramm dargestellte gewünschte Verhalten einer Pneumatikschaltung verletzt und wo korrektes Verhalten vorliegt.
- *Konzeptbasierte Hilfen und Heuristiken* unterstützen den Entwurfsprozeß in frühen Problemlösephasen.
- Eine *Simulationsumgebung* ermöglicht die Visualisierung des dynamischen Schaltungsverhaltens des realisierten Entwurfes.
- Ein *Dozentenmodus* ermöglicht die interaktive Erstellung neuer Aufgaben und die Einbindung in das System durch einen Dozenten.

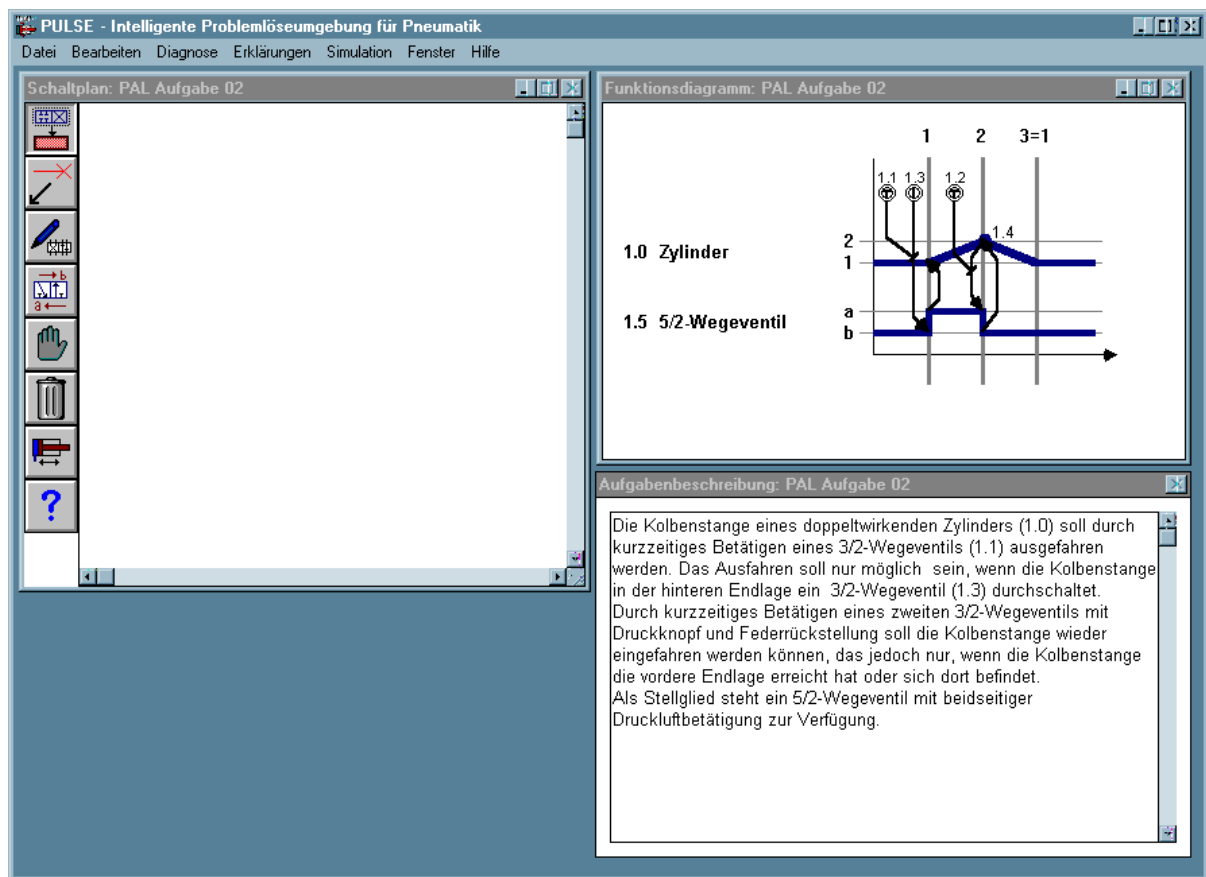


Abb. 1: Aufgabe PAL 02

### 3.1 Der Hypothesentest-Ansatz: Model-Checking und statische Analyse

Nach Auswahl einer Aufgabe präsentiert sich dem Lernenden die Aufgabenstellung in Form einer textuellen Beschreibung sowie eines Funktionsdiagramms. Ein leerer Editor steht zur Konstruktion des Lösungsentwurfes zur Verfügung (siehe Abb. 1). Pneumatische Bauglieder nach DIN-ISO 1219 können in beliebiger Weise auf der Entwurfsfläche plaziert und durch Leitungen miteinander verbunden werden. Durch Benennung der einzelnen Bauglieder mit den Bezeichnungen aus der Aufgabenstellung wird der Entwurf für das System überprüfbar.

Wenn der Lernende sich bereits in der Domäne auskennt, kann er, ohne durch das System unterbrochen zu werden, eine Lösung generieren und diese im Anschluß durch den in PULSE integrierten, dynamischen Model-Checker (Clarke, Emerson & Sistla, 1986; Damm, Döhmen, Gerstner, Josko, 1990; Josko 1990) verifizieren lassen. Eventuelle Vorkenntnisse des Lernenden werden optimal unterstützt. Die Dynamik eines Schaltplans kann *jederzeit* durch den dynamischen Model-Checker analysiert werden. Dafür markiert der Lernende im Funktionsdiagramm eine Testhypothese, von der er annimmt, daß sie bereits korrekt realisiert wurde. Er hat die Möglichkeit nur einzelne Bauteile oder bestimmte Taktschritte, zum Beispiel bis zu einer bestimmten Schaltungsstellung, prüfen zu lassen. Die Korrektheit der Hypothese wird durch Erzeugung eines Fallgraphen überprüft, der sämtliche von einem Initialzustand erreichbaren Zustände der pneumatischen Steuerung und deren Übergänge enthält. Aus der Testhypothese können temporallogische Formeln abgeleitet werden, die anhand des Fallgraphen auf Gültigkeit überprüft werden können. Fehler im Verhalten der Schaltung werden durch entsprechende Markierung im Funktionsdiagramm visualisiert.

Diese Analysemethode hat den Vorteil, zu jedem Entwurf und jeder Testhypothese eine korrekte Antwort generieren zu können. Daher kann sie von einem erfahrenen Lerner genutzt werden, um den Lösungsraum einer Aufgabenstellung explorativ zu erforschen, d.h. verschiedene Lösungsalternativen aufzubauen und schnelle Antworten über die Korrektheit seiner Entwurfsideen zu erhalten. Der dynamische Model-Checker bezieht sich bei seiner Rückmeldung jedoch immer nur auf die dynamischen Aspekte des Funktionsdiagramms. Bedingungen, die zum Beispiel aus dem Text der Aufgabenstellung hervorgehen, z.B. daß ein bestimmtes Ventil zur Lösung der Aufgabe genutzt werden soll, können nicht überprüft werden. Es muß dafür auf die statische Analyse zurückgegriffen werden.

Die Rückmeldungen basieren immer auf dem Funktionsdiagramm, daher sind die Informationen sehr fein-granuliert und beziehen sich immer auf einzelne Bauteile und Taktschritte. Die Erklärung von Fehlern ist auf Grundlage des dynamischen Model-Checkers entsprechend schwierig. Der Lernende muß sich selbst erklären, wie Abweichungen von dem erwarteten Verhalten entstanden sind. Ihm können keine abstrakteren Erklärungen auf einer höheren Konzeptebene angeboten werden, wie sie von Lehrern oder Lehrbüchern genutzt werden.

Es ist anzunehmen, daß ein unsicherer Lerner jedoch bereits in früheren Entwurfsphasen Feedback durch das System benötigt. So kann bereits das Lesen der Aufgabenstellung Stocksituationen auslösen, beispielsweise durch unbekannte Konzepte (3/2-Wegeventil, Stellglied), oder fehlende oder unvollständige Pläne zur Umsetzung der Aufgabenstellung.

Grundlegende Begriffsdefinitionen, auf die sich die Aufgabenstellung und die konzeptbasierten Erklärungen beziehen, werden deshalb durch PULSE *just in time* zur Verfügung gestellt. Es werden *Objektkonzepte* und *Funktionskonzepte* unterschieden. Bei Objektkonzepten handelt es sich um die Bauteile der Domäne, die sich in einer Hierarchie (siehe Abb. 2) organisieren lassen. Durch Verbindung der Bauteile mit Leitungen im Editor werden Funktionen realisiert. Auch können einzelne Bauteile innerhalb der Schaltung bestimmte Rollen übernehmen. Ein Ventil übernimmt beispielsweise die Rolle eines *Stellglieds*, wenn ein Zylinder direkt und nur durch dieses Ventil angesteuert wird. Diese Funktionskonzepte bilden ebenfalls eine Hierarchie, an dessen Spitze das Funktionskonzept der *Indirekten Steuerung* steht. Diese Hierarchien können direkt von dem Lernenden eingesehen werden und durch Anklicken des gewünschten Bauteils oder der gewünschten Funktion werden ihm detailliertere Informationen, teilweise durch Grafiken und Animationen unterstützt, präsentiert.

Der Lernende hat darüberhinaus die Möglichkeit, *aufgabenspezifische Erklärungen* anzufordern, die beispielsweise beinhalten, welche Objekte und Funktionen zur Lösung der speziellen Aufgabe zu realisieren sind. Diese Erklärungen sind nur von der Aufgabenstellung abhängig. Die dafür benötigten Daten werden beim Starten der Aufgabe durch PULSE aus der Aufgabenstellung mittels Heuristiken extrahiert. Daher ist PULSE in der Lage, konzeptbasierte Erklärungen zur Aufgabenstellung auch zu begründen. Ein eventuell bereits vorgenommener Lösungsansatz des Lernenden in Form eines Schaltplans wird dabei noch nicht berücksichtigt.

PULSE stellt weiterhin Erklärungen zur Verfügung, die sich auf die *Umsetzung der Aufgabenstellung* in einen Schaltplan beziehen. Dabei handelt es sich um domänenspezifische Regeln und Heuristiken, die dem Lernenden auf Anfrage zur Verfügung gestellt werden.

Zur Lösung einer Aufgabenstellung reichen aufgabenspezifische Erklärungen allein aber im Allgemeinen nicht aus. Normalerweise baut der Lernende einen eventuell unvollständigen Schaltplan auf und möchte diesen durch das System bewerten lassen. Eventuell fehlt ihm Wissen, wie er mit der Lösung der Aufgabe fortfahren soll. Er benötigt also zum einen

schaltplanbezogene Erklärungen, für die sein Entwurf analysiert und bewertet werden muß, und Erklärungen zur Korrektur bzw. Ergänzung seines Entwurfes.

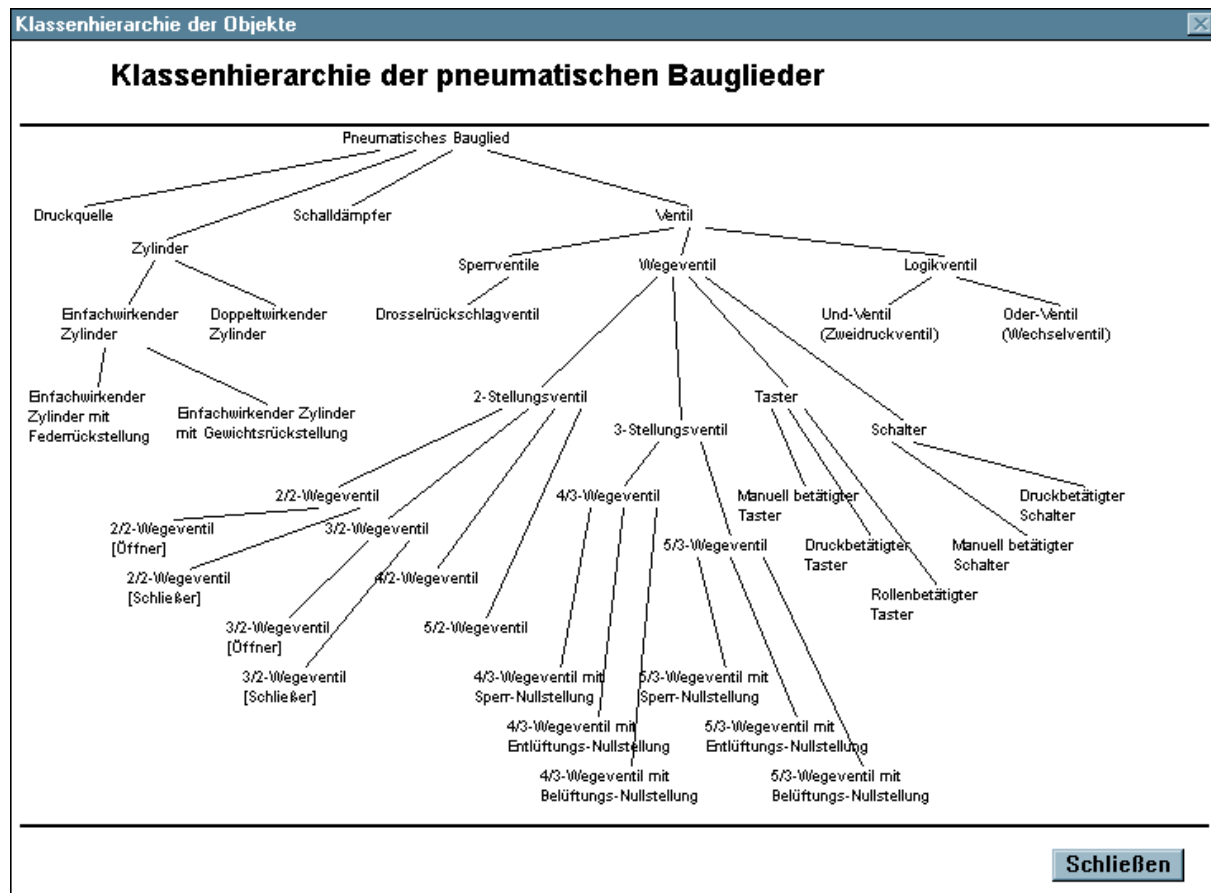


Abb. 2: Die Hierarchie der pneumatischen Bauteile

Bei der statischen Analyse des Schaltplans handelt es sich um eine planbasierte Analyse des Schaltplans mit allen Vor- und Nachteilen, die bereits in anderen Projekten identifiziert wurden. Die planbasierte Analyse ist eine unvollständige Analysemethode, da eventuell nicht der gesamte Problemraum durch die statische Analyse abgedeckt wird. Der Schüler kann möglicherweise Schaltungen entwerfen, die PULSE aufgrund der beschränkten Menge integrierter Pläne nicht identifizieren kann. Dadurch kann es zu Fehlinterpretationen durch das System und sogar zu fehlerhaften Rückmeldungen kommen. Im Gegensatz zum dynamischen Model-Checking ist die statische Analyse also nicht inhärent vollständig und korrekt. Die Qualität der statischen Analyse ist abhängig von der Vielfalt der dem System bekannten Pläne. Zu jedem Konzept der Aufgabenstellung muß PULSE mindestens einen Plan kennen, wie dieses Konzept auch im Schaltplan umgesetzt werden kann. Durch die Existenz verschiedener Pläne zur Realisierung eines Konzeptes wächst die Anzahl der erkannten korrekten Lösungen. Die statische Analyse ist erfolgreich, wenn sehr viele korrekte Pläne integriert sind. Es muß jedoch immer davon ausgegangen werden, daß nicht alle korrekten Pläne identifiziert wurden bzw. werden können. Auch steigt mit der Anzahl der Pläne natürlich auch die Dauer der Analyse, so daß zwischen Zeitbedarf und 'Erfolgsquote' abgewogen werden muß.

Es ist sinnvoll, sich den dynamischen Model-Checker zur Identifikation noch nicht bekannter Pläne nutzbar zu machen. Kann die statische Analyse einen Entwurf nicht identifizieren, weil beispielsweise ein komplexerer Plan fehlt, so wäre es mit Hilfe des dynamischen Model-

Checkers möglich, zu erkennen, ob sich der Schaltungsentwurf dennoch korrekt verhält. Ist das der Fall, so kann davon ausgegangen werden, daß ein vermeintlich fehlendes Konzept durch eine Ersatzschaltung realisiert wurde. Durch Analyse der relevanten Druckluftströmungen, die aus den Daten der Fallgrapherzeugung des dynamischen Model-Checkings entnommen werden können, kann ein Modul isoliert werden, das dasselbe Ein-/Ausgabeverhalten aufweist, wie das fehlende Konzept. Dieses Ersatzmodul kann dann als neue Realisationsmöglichkeit des Konzeptes in das System integriert werden.

*Schaltplanbezogene Erklärungen* können sich ebenfalls auf Objekt- und Funktionskonzepte beziehen. Der Entwurf des Lernenden wird zunächst unabhängig von der Aufgabenstellung analysiert. Dabei werden die Objekte untersucht und ihre spezifischen Eigenschaften festgehalten. Diese Daten werden dem Lernenden auf Wunsch zur Verfügung gestellt. Auch wird festgehalten, wie die Objekte zu Funktionen verknüpft worden sind.

Der Vergleich der Konzepte des Entwurfs mit denen der Spezifikation deckt Fehler im Entwurf auf. So können fehlerhafte, fehlende oder zusätzliche Objekte und Funktionen identifiziert und dem Lernenden als schaltplanbezogene Informationen zur Verfügung gestellt werden.

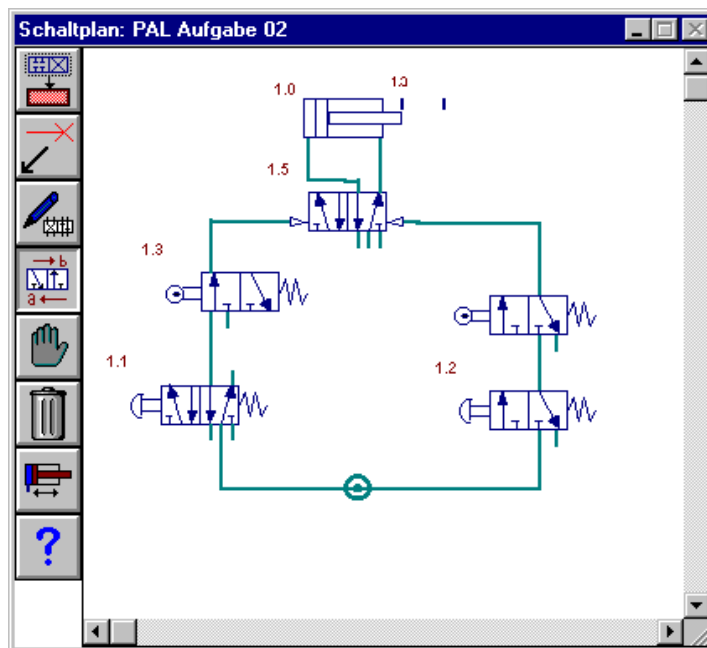


Abb. 3: Fehlerhafter Entwurf eines Lernenden

In Abb. 3 wird der Entwurf eines Lernenden zur Lösung von Aufgabe 2 (siehe Abb. 1) dargestellt. Der Aufruf der Diagnose bewirkt zunächst einen Test auf Basis der statischen Analyse, um fehlerhafte und unbenannte Objekte zu identifizieren und auf inkorrekte Ausgangsstellungen der Bauglieder hinzuweisen. Eine Meldung gibt dem Lernenden entsprechende Hinweise (siehe Abb. 4), damit er sich eventuelle Fehler in der Dynamik besser erklären kann. Genauere Details werden an dieser Stelle nicht zur Verfügung gestellt. Der Lernende kann nun den Test seines Entwurfs abbrechen und selbständig die Fehler identifizieren und korrigieren, oder sich durch die Erklärungskomponente gezielt Hinweise geben lassen. Fährt er mit dem Test seiner Lösung fort, so erhält er die in Abb. 5 dargestellte Rückmeldung auf Basis des Funktionsdiagramms.



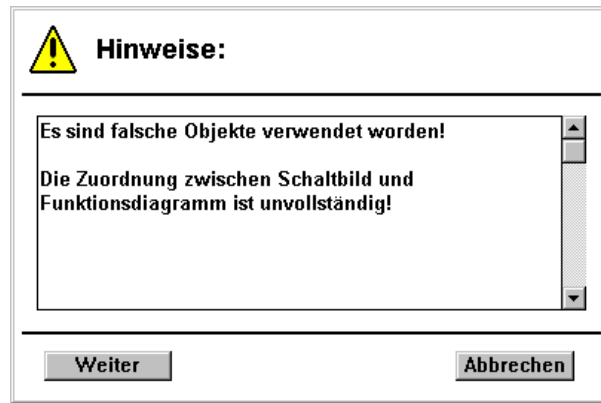


Abb. 4: Erste Rückmeldungen

Er steht nun vor der Aufgabe, die dynamischen Effekte seines Schaltungsentwurfes zu interpretieren und Fehler in der Dynamik durch Fehler in der Statik zu erklären. Diese Aufgabe ist für Novizen häufig nicht allein durchführbar. Ohne Hilfe durch das System befände sich der Lernende in einer Stocksituation, die er eventuell nicht selbständig überbrücken kann. Es ist zu vermuten, daß dies seine Motivation, mit dem System zu lernen, stark negativ beeinflussen würde. Die Erklärungskomponente auf Basis statischer Daten kann nun den Lernenden in seinem Problemlöseprozeß unterstützen. Er kann entweder die Daten der Analyse seines Schaltplanes betrachten, wobei ihm fehlerhafte, fehlende und unbenannte Bauteile und Funktionen genannt werden. Dabei würde ihm an dieser Stelle überlassen bleiben, wie er die konkreten Probleme lösen kann. Oder aber, wenn er sich nicht in der Lage fühlt, in seinem Entwurf fortzufahren, kann er sich Tips von der Erklärungskomponente geben lassen, die ihm auf Wunsch detailliert erläutert, was zu tun ist, und wie ein Entwurf zu einer korrekten Lösung transformiert werden kann.



Abb. 5: Feedback des dynamischen Model-Checkers

Für *Korrektur- und Ergänzungsvorschläge* durch das System mußte eine Synthesekomponente implementiert werden, die ausgehend von einem Entwurf eine korrekte Lösung generiert, die möglichst wenige Änderungen an dem bestehenden Entwurf notwendig macht. Der Lernende wird auf diese Weise nicht mit einer Musterlösung des Systems konfrontiert, die nichts mit seinem Entwurf gemein hat, sondern erkennt korrekte Teile seines Entwurfs wieder und wie sie in eine korrekte Lösung integriert werden können. Die Hinweise werden zunächst auf abstrakter

Ebene gegeben, d.h. es wird zum Beispiel zuerst die Reparatur des obersten Konzeptes gefordert, in dem ein Fehler vorliegt. Reicht diese Information nicht aus, so kann durch Nachfragen die Position des Fehlers immer mehr eingeschränkt werden. Auch bei Ergänzungsvorschlägen wird nach dieser *Methode der minimalen Information* vorgegangen. Sie soll bewirken, daß der Lernende sich selbst Gedanken über die Lösung macht und Fehler möglichst selbst entdecken und korrigieren kann. Es wird angenommen, daß Informationen, die sich der Lernende selbst erarbeitet hat, qualitativ hochwertiger sind, d.h., daß der Lernende sich länger daran erinnern kann und sie häufiger anwenden wird. PULSE ist ebenfalls in der Lage auf Wunsch auch konkrete Hinweise zu geben, die sich direkt auf das Ändern oder Hinzufügen einzelner Bauteile oder Leitungen beziehen.

In dem gegebenen Beispiel sind mehrere Fehler kombiniert aufgetreten, die dem Lernenden erklärt werden müssen. a) Bei dem gemeldeten falschen Objekt handelt es sich um das Ventil 1.1, das laut Aufgabenstellung ein 3/2-Wegeventil sein soll. Im Entwurf wurde statt dessen ein 5/2-Wegeventil gewählt, das durchaus das korrekte Verhalten aufweisen kann und daher im Feedback der dynamischen Analyse auch als korrekt dargestellt wurde.

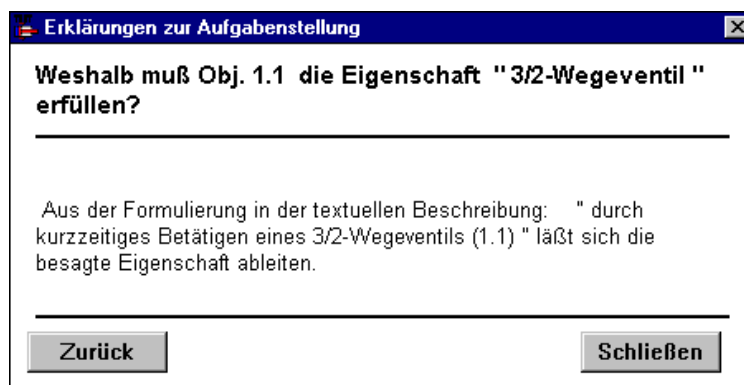


Abb. 6: Aufgabenspezifische Erklärung zu einem Bauteil

b) Eine fehlende Zuordnung bewirkt, daß die Dynamik des unbenannten Ventils durch den Model-Checker nicht geprüft werden kann. Das entsprechende Ventil 1.4 ist daher in der Rückmeldung als falsch markiert, ob wohl es, allerdings unbenannt, bereits korrekt realisiert wurde. c) Die Ursache dafür, daß sich der Zylinder nicht korrekt verhält, liegt in einer fehlenden Leitung beim Druckanschluß. Da Ventil 1.5 nicht an die Druckquelle angeschlossen ist, kann der Druck nicht zum Zylinder weitergegeben werden. Dies zeigt deutlich ein Manko der dynamischen Analyse. Obwohl der Schaltplan bereits korrekt aufgebaut sein kann, liefert sie zwar korrekte, aber unpassende Rückmeldungen, wenn die Schaltung nicht ablauffähig ist, zum Beispiel, wenn die notwendigen Anschlüsse an die Druckquelle fehlen. Dadurch ist der dynamische Model-Checker für das Testen von Teilentwürfen nur bedingt geeignet.

#### 4 Zusammenfassung

Der Hypothesentest-Ansatz ist eine Methode, einem Lernenden individuelles Feedback zu liefern, ohne die Probleme des Aufbaus eines Schülermodells (Korrektheit, Adäquatheit, Datenschutz, etc.) in Kauf nehmen zu müssen. Die statische Analyse ist eine wichtige Informationsquelle, um dem Lernenden auf Basis abstrakterer Informationen und höherer Konzepte Fehler und weitere Vorgehensweisen zu erläutern. Sie sind daher insbesondere in den

frühen Phasen eines Entwurfsprozesses, den Abwäge- und den Planungsphasen, nützlich. Solche Erklärungen stehen auf Basis des dynamischen Model-Checkings nicht zur Verfügung. Fehlende Pläne zur Realisierung von Konzepten können jedoch fehlerhafte Erklärungen generieren. Das dynamische Model-Checking bietet dafür eine vollständige und korrekte Methode zur Analyse beliebiger Schaltpläne. Eine Kombination beider Analysemethoden bietet dem Lernenden sowohl die fein-granulierten Erklärungen des dynamischen Model-Checkers als auch die höheren Konzepte, wie sie in früheren Entwurfsphasen benötigt werden. Fehlende Pläne der statischen Analysemethoden können mit Hilfe des dynamischen Model-Checkers ergänzt werden. Durch Integration beider Analysemethoden in dem System PULSE hoffen wir, die Vorteile der Methoden zu nutzen und die Nachteile ausgleichen zu können.

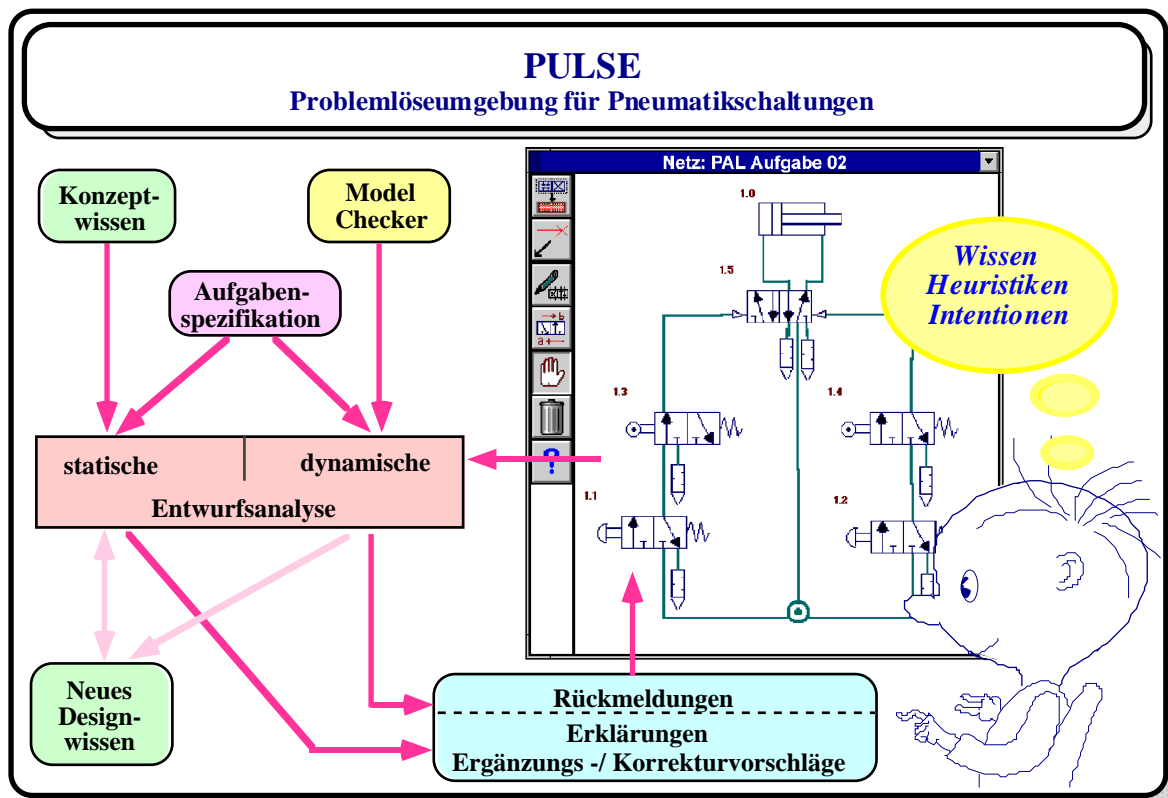


Abb. 7: Die IPSE PULSE im Überblick

## Literatur

- Clarke, E.M., Emerson, F.A. & Sistla, A.P. (1986). Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications. *ACM Transactions on Programming Languages and Systems*, Vol. 8, No. 2, 244 - 263
- Damm, W., Döhmen, G., Gerstner, V., Josko, B. (1990). 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*. Berlin: Springer, LNCS 430.
- Henning, U. (1994). Analyse eines Modells paralleler Prozesse für die Anwendung in einem intelligenten Lehrsystem, Dissertation, TU München, Institut für Informatik.
- Herzog, C. (1996). Syntaxisorientierte vs. ablauforientierte Diagnose in intelligenten Programmierumgebungen als Beispiel für den Einsatz konkurrierender Problemlöser. in: M. Thielscher, S.-E. Bornscheuer (eds.), *Fortschritte der Künstlichen Intelligenz (KI 96)*, Dresden: Dresden University Press.
- Johnson, W.L. (1986). *Intention-based Diagnosis of Novice Programming Errors*. Los Altos: Morgan Kaufmann.
- Josko, B. (1990). 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*. Berlin: Springer, LNCS 430, 386-400.
- Miller, M.L., Goldstein, I.P. (1976). SPADE: a grammar bases editor for planning and debugging programs. AI Lab Memo 386. Massachusetts Institute of Technology, Cambridge, Massachusetts.
- Möbus, C., Pitschke, K., Schröder, O. (1992). 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, 294 - 301
- Möbus, C. (1995). Towards an Epistemology of Intelligent Problem Solving Environments: The Hypothesis Testing Approach, in: J. Greer (ed.), *Artificial Intelligence in Education, Proceedings of AI-ED 95*, Washington, D.C., August 16-19, 1995, Charlottesville: AACE, 138-145
- Möbus, C., (1996). Towards an Epistemology on Intelligent Problem Solving Environments: The Hypothesis Testing Approach, in: *Proceedings of EuroAIED 96*, Lisbon, Portugal, Sept. 30 - Oct. 2, 1996