

Evolution of the Hypotheses Testing Approach in Intelligent Problem Solving Environments

Janine Willms, Claus Möbus

Fachbereich Informatik, Abteilung Lehr-und Lernsysteme,
C.v.O. Universität Oldenburg, D-26111 Oldenburg, Germany
{Janine.Willms,Claus.Moebus}@Informatik.Uni-Oldenburg.de

Abstract. In this paper, we compare different realizations of the hypotheses testing approach in the IPSEs (Intelligent Problem Solving Environments) ABSYNT, PETRI-HELP and MEDICUS and introduce the changes necessary to transfer the hypotheses testing approach to the real world domain of patent applications. Patent-IT is the first IPSE to overcome the limiting aspects of fixed specifications and a black box oracle.

1 Hypothesis Testing and the IPSE Approach

The *hypotheses testing approach* is a core concept of what we call *intelligent problem solving environments* (IPSE) ([8], [9]) and also gives a key qualification having a beneficial influence on the student's knowledge acquisition process. The learner acquires knowledge by actively exploring a domain, creating solution proposals for problems, testing hypotheses about their correctness, during which the system analyzes the proposals and provides help and explanations, making use of an oracle or an expert knowledge base. The IPSEs we developed initially had some limiting aspects such as fixed specifications and a black box oracle. We now present the IPSE Patent-IT, which does not exhibit these limitations. This has become necessary due to the rather demanding domain of inventions and patents. The novice should learn how to transform an inventive idea into a legal patent.

The IPSE approach is psychologically based on the ISP-DL theory of knowledge acquisition and problem solving [8], which is influenced by theoretical assumptions of van Lehn [13], Newell [11], Anderson [1] and Gollwitzer [5]. It briefly states that new knowledge is acquired as a result of problem solving by applying weak heuristics in response to impasses. Furthermore, existing knowledge is optimized if applied successfully. The learner encounters four distinct problem solving phases namely deliberation, resulting in marking a goal as an intention, planning how to satisfy the intention, execution of the plan and evaluation of the result. Several design principles for IPSEs [8] could be drawn from the following assumptions:

- The system should not interrupt the learner but offer help on demand. According to the theory, the learner will look for and appreciate help at an impasse.
- Feedback and help information should be available on request at any time, taking the actual problem solving phase of the learner into account.

- Help should be tailored to the learner's pre-knowledge as much as possible. The best method to fulfill this requirement is to let the learner freely state hypotheses of the solution.

1.1 The Scope of our IPSEs so far and what is to come

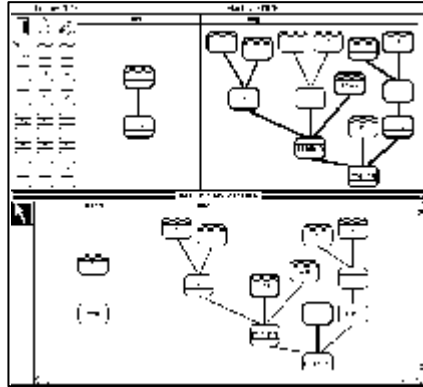
We have developed several IPSEs in variable application domains. The one common characteristic is that, the hypothesis testing process is fixed and hidden in a black box. The IPSEs ABSYNT and PETRI-HELP each define a closed world in which the learner explores a domain. MEDICUS defines the next step towards a non-closed world application, because real-world scenarios may act as a source for modeling. The modeling task is left to the learner, and MEDICUS is able to evaluate the equivalence of two different representations (specification, bayesian belief networks) of the real world scenario by an internal and fixed diagnostic process.

Patent-IT is an IPSE, which supports the application of a patent by assessing critical aspects. The learner needs *evaluation, judgement and argumentation skills* to perform this task. He has to construct a model of his invention and is supported by the IPSE in his critiquing process. In this connection, the domain of patents serves as a kind of *metadomain*, which *incorporates the hypothesis testing process*. Neither a task specification nor the model are fixed in Patent-IT. The patentability of an invention is dependent on the state of the art. It is necessary that the model is *not* a derivative from the state of the art. This differs from previous IPSEs, where the model was to be equivalent to a specification or its logical conclusion. The state of the art for an invention is a *real world concept*. It is defined by a research in real world patent and literature databases and consists of a set of documents. By changing the model of the invention the state of the art also changes. This dynamic behavior of the domain makes a diagnostic process quite difficult and is realized in Patent-IT as a dialogue based cooperation of the system and the learner, which makes the process transparent and understandable for him. The critical dialogue depends on the modeled invention, the domain ontology known to the system, the results of the search for the state of the art and the user's statements and arguments. By defending his invention, the learner may use his pre-knowledge as much as possible. In impasse situations where he needs support, Patent-IT is able to control the evaluation process, direct the learner and offer argumentative hints. Patent-IT is therefore the next consequent step on the evolutionary line of IPSEs.

1.2 ABSYNT ("Abstract Syntax Trees")

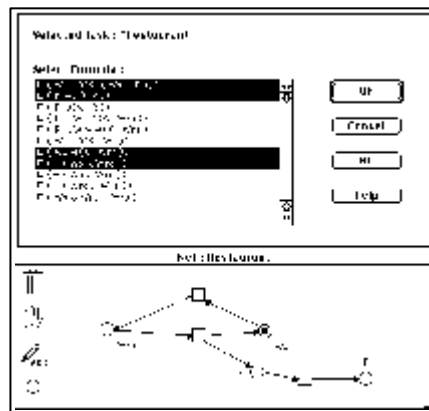
The ABSYNT [8] problem solving environment supports learners by offering help and proposals for functional programming in a graphical tree representation of pure LISP. The learner is given a fixed set of tasks. The programming task is internally represented as a symbolic goal, which triggers a set of transformation rules developed in the Munich CIP project ([2], [12]). In a diagnostic, hypotheses and help environment, the learner may visually state the hypothesis in a tree-like representation, that his solution proposal (or a boldly marked part of that proposal) to

a programming task is correct. The system then analyzes this part of the solution proposal. One reason for the hypotheses testing approach is that, in programs, bugs cannot be absolutely localized. Therefore, the decision, which parts of a buggy



solution proposal are to be kept, is left to the learner. This results in the system giving help and error feedback on the implementation level by synthesizing complete solutions, starting from the learner's hypothesis. If the hypothesis is embeddable within a complete solution, the learner may ask for completion proposals. The hypotheses testing is a hidden process, based on a set of hundreds of diagnostic rules, defining a "goals-means-relation", which analyzes and synthesizes several millions of solution proposals for 40 programming tasks. The rules, which are not shown to the user, generate complete solutions, recognize and complete incomplete proposals. The learner works in a closed world, defined by the tasks and the internal rules of the IPSE.

1.3 PETRI-HELP



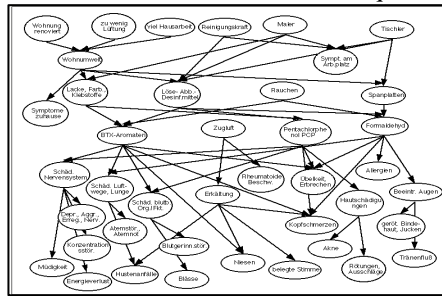
PETRI-HELP [8] supports novices in learning to model dynamic systems with condition-event petri nets. Tasks are stated as a set of temporal logic formulas, which describe the behaviour of a dynamic system. The learner is asked to construct a petri-net model that fulfills the given formulas. He may test hypotheses based on his solution by selecting those formulas he believes to be fulfilled by his model. The system checks the hypotheses through model checking ([3],[7]) by interpreting the temporal logic formulas on the case graph of the learner's petri net. This model

checking mechanism is a domain-independent method for dynamic systems. We adapted it for an IPSE in the domain of pneumatic (PULSE) [14] and electronic circuits (MSAFE). The model checking process is normally hidden from the learner.

1.4 MEDICUS

MEDICUS [4], [9] is designed as a problem solving tool for modeling uncertain domains. Diagnosis in domains of complex, fragile and uncertain knowledge is quite a difficult reasoning and problem solving task. Therefore, the training of diagnostic

strategies will be supported by the system qualitatively (i.e., what information is necessary to support or differentiate between given hypotheses?) as well as quantitatively (i.e., how does information gathered affect my diagnostic hypotheses? Which information should be acquired next?). The learner may state hypothetical



conclusions of a given situation and the resulting strategic actions.

The learner may construct Bayesian Belief Networks (BBN) as explanatory models, evaluate their consequences qualitatively and quantitatively and revise the models with support from the system. The learner asserts the dependence or independence of multiple variables in a diagnostic dialog. He then states the hypothesis that his BBN

is consistent with this information. The system analyzes this hypothesis using the d-separation criterion. If the dependence and independence assertions are not given by the modeled BBN, a new BBN is constructed internally from the dependence and independence assertions and compared to the modeller's graph. The hypotheses testing method is a mathematical method, external to the "learn domain". It always leads to a correct answer and allows explanations for a redesign of the modeled BBN. This may lead to the result that edges have to be removed from and/or added to the graph in order to be consistent with the dependencies and independencies.

2 Elaboration of a Taxonomy of Different Hypotheses Used in our Prior IPSEs

On a high level viewpoint all those IPSEs share the characteristic, which enables the learner explore a domain, state hypotheses after which the system assesses the correctness of the solution proposal by a domain-independent diagnostic process. Even in those systems, different types of hypotheses can be found depending on task formulation and solution proposal. On taking a closer look, at least two different dimensions of hypotheses are used in the initial IPSEs. In ABSYNT, a hypothesis is actually (a part of) the *solution proposal*, whereas in PETRI-HELP, the hypothesis is a selection of the *task specification*. For many domains it is possible to offer those two dimensions of hypotheses.

MEDICUS is slightly different, because no specific task is given. Instead, the learner himself models two different representations (dependence/independence statements and BBNs) of the same domain with the hypothesis that both representations are equivalent. The independence statements play the role of a specification and the BBNs, the role of the solution model.

We will show that hypotheses testing in Patent-IT still allows another approach. Like in MEDICUS, there is no task given, but several criteria mentioned in the patent law must be fulfilled by the learner's invention.

The learner constructs a model of his invention. This model may be divided into several parts and each combination out of these may serve as a hypothesis. This is similar to hypotheses testing in ABSYNT. It is also possible to state different hypotheses according to the criteria of the patent law. Some of these criteria are independent, hence the possibility of checking them separately. Others depend on each other, e.g. it is impossible for an invention to be inventive if it is not novel. The opportunity to state hypotheses according to the criteria which have to be proven is similar to the approach followed in PETRI-HELP.

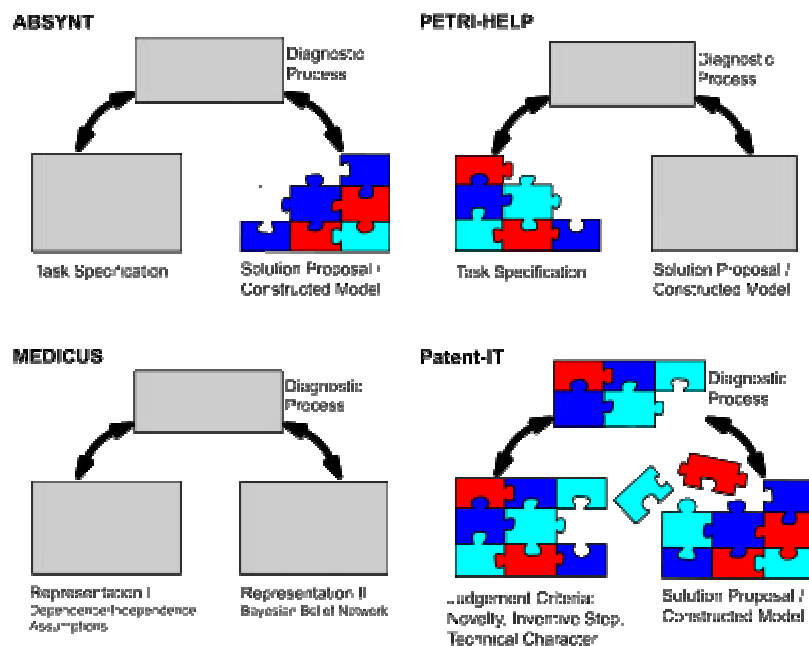


Fig. 1. Overview of different hypothesis selection opportunities

Patent-IT offers a further opportunity of stating a hypothesis. The diagnostic process of an IPSE is embedded in the *metadomain* of patent application examination. This examination process was elicited by empirical studies by the first author at the German Patent Office (DPMA)¹ and the European Patent Office (EPA) and integrated into Patent-IT. The process is made transparent to the inventor in order to familiarize him with the patent law and the processes in the patent offices. In comparison to prior diagnostic processes in our IPSEs, the analysis method in Patent-IT is neither complete nor correct. The process heavily depends on heuristics and ontological knowledge of the invention domain. The diagnostic process is therefore carried out as a sequential process in cooperation with the system and the learner. It is possible to enter this process at different entry points, i.e. under different assumptions. Everything before the entry point forms a theoretical basis, which may or may not be

¹ Grateful acknowledgements are due to the DMPA and the EPA for having given us the opportunity to participate in the work carried-out in these patent organizations.

true. This is a hypothesis testing approach, which was not available in the initial IPSEs, due to the diagnostic process being hidden in a black box.

We can see that the hypotheses testing approach may be applied in many different domains leading to different forms of realization (Fig. 1). In the next section, we will describe the IPSE Patent-IT in greater detail.

3 Patent-IT

Patent-IT is designed to help an inventor to evaluate the acceptability of his invention as a patent. The goals of Patent-IT are to:

- explain basic concepts of the patent domain “just in time”,
- introduce the judgement criteria and judgement processes which are applied by the patent researchers and patent examiners,
- lower the inhibition level of applying for a patent by offering simulated trial applications,
- gain more detailed information before an expensive patent application process is started,
- stimulate the systematic structuring of the invention,
- discover a strategic combination of invention features,
- help the inventor search for the state of the art,
- offer an argumentation testing ground to defend the patentability of his invention,
- find and emphasize the critical aspects of his patent application.

In this case, the evaluation process is difficult because many undefined factors are influencing the result. Usually, the inventor is unable to tell if his invention is patentable or not. Moreover, the details of the current state of the art are unknown to him. One of the most important steps in the evaluation process is therefore to find the relevant state of the art. The inventor does not even know the best level of abstraction to describe his invention and which linguistic form he should use. The use of *natural language texts* to describe an invention is another factor of uncertainty. Usually, words, terms or phrases can be understood differently. Many words have two or more meanings and misunderstandings are normal. Even if two people agree on the same meaning of a concept, they may have a *different ontology*, leading to the fact that, one says that the invention is not novel compared to a document of the state of the art whereas the other rejects this point of view.

We designed a new cooperative form of the hypotheses testing method as a result of these uncertainties. The user, in this case the inventor, is included in the evaluation process. The evaluation process itself is not domain independent any more but a simulation of the processes in the patent organization.

The examination process can be divided into several subtasks (Fig. 2). Usually, the inventor prepares the application for patent with or without the help of a patent attorney, after which the application is examined by the patent organization. The initial step involves the classification of the invention according to the International Patent Classification (IPC). The application is then transferred to a research department, where the search for the state of the art is performed resulting in a search report which is transferred to the patent applicant.

The search report is the foundation for further examination. A patent examiner compares the documents in the state of the art with the invention. At the EPA he

decides the novelty and inventiveness of the patent application in a process called the “Problem-Solution-Approach”.

According to the patent law a condition to be fulfilled by the invention is its technical character. This is difficult to decide especially if the application refers to a software-related invention.

Patent-IT is designed to assess the patentability of an invention *before* the application process is started and *before* the actual features to be claimed are determined by writing down the application for a patent. This means that the whole process of examination must be simulated by the system. The system acts as classifier, researcher and examiner and represents the public by attacking the user’s invention but at the same time it supports the user when necessary. Unfortunately, the method to evaluate the users inventive idea has become domain-specific and uncertain.

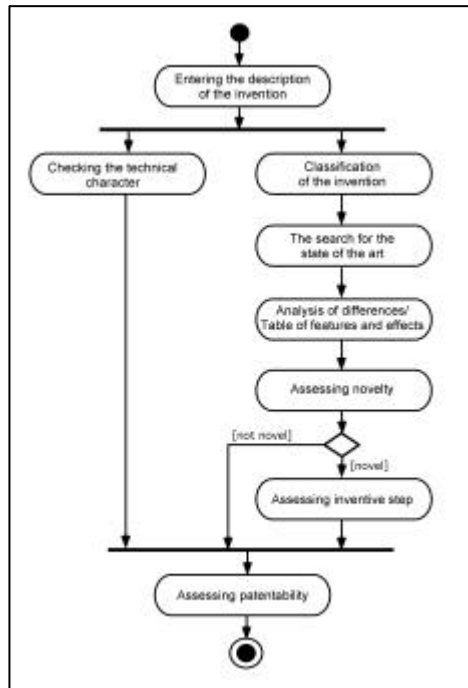


Fig. 2. A model of the evaluation process

We will now describe the *cooperative process of evaluation* between the inventor and the IPSE Patent-IT. As stated earlier, the evaluation process heavily depends on environmental factors, heuristics and ontological knowledge.

When Patent-IT is started, the user is asked in a html-based dialogue to describe his invention, i.e. the topic, a title, features and special characteristics of features, effects and advantages as well as disadvantages of the invention. These phrases are analyzed to extract keywords to be searched for, and stopwords which are neglected. The system is designed to allow easy access to a common but limited ontological network called Wordnet [15] in order to expand each keyword to a set of synonyms, super- and sub-terms. The search is performed in the IBM-Patentserver [6] and the results are presented to the user. Now it is the user’s task to select the most relevant documents. These documents define the state of the art for this special invention. The next step is to generate the table of differences. Usually, the inventor is convinced that his invention is novel and inventive. It is now important to assess the similarities of the invention and the state of the art in the most objective way. In a critical dialogue, the system tries to find a justification for the belief that the feature in focus is already given in the state of the art. The inventor has to defend his invention. This dialogue game results in a filled table of differences, in which an indication is made for each feature or effect of the invention found in each of the documents of the state of the art. This table is used by the system to trigger empirical rules about novelty and inventiveness. In spite of the inventiveness of the user’s idea, it could be rejected

because of the lack of technical character. It is therefore necessary to assess the technical character of the invention. Patent-IT uses a database of arguments on the technical character of several inventions taken from legal proceedings in order to attack the user's invention or to give him advice on how to present his case.

As already stated, the user engages in a cooperative process with the system, both having distinct tasks and skills. Two parts are needed in order to assess the patentability of the invention in this dialogue based evaluation process: a) The user's knowledge about the domain of the invention, his ability to understand the content of documents and to generate arguments and b) the system's knowledge about how to perform searches and how to decide whether an invention defined by a set of features and effects is inventive or not.

3.1 Hypothesis Testing in Patent-IT

Hypothesis testing is made possible by choosing a *set of features* as defining elements for the invention. If the result of the evaluation process poses a question of the novelty and inventiveness of the application, the user may adjust his hypothesis by adding more detailed features and therefore reducing the claimed scope of the protection.

Furthermore, it is possible to state a *hypothesis based on certain judgement criteria*. For example, the user may state that his invention is patentable according to the criteria novelty and inventiveness under the assumption that the technical character of the invention is given.

Due to the stepwise process of evaluation, a third kind of hypothesis is possible. The user is able to define an *entry point and a stopping point for the evaluation process*. This allows partial tests of the invention, which are not directly related to the judgement criteria. One example may be the evaluation of an invention given a set of predefined documents as the state of the art.

The methods used by Patent-IT to perform the evaluation processes are based on a cooperative dialogue between the system and the user. Patent-IT makes use of multiple knowledge bases in order to find justifications for a critical dialogue. The user is however always able to state that the system's conclusions are wrong in this special case. The knowledge bases are adapted according to the user's statements. New synonyms and ontological relationships are saved in a separate user-wordnet. They may be transferred to the original wordnet after review. New legal proceedings may be integrated into the case base in order to introduce new arguments. Incorrect results of the hypothesis testing are possible, because a user is able to tell lies to the system or to define meaningless inventions. But Patent-IT is designed as a system to support the applicant by providing a testing ground for a real patent application. Therefore telling lies to the system would not make sense.

Even though the dialogue based evaluation process may still produce incorrect results because Patent-IT is an open system, and the evaluation process depends on the chosen state of the art. In reality, even if two different examiners examine the same application resulting in a different state of the art, the overall result concerning the patentability of the application is mainly the same. We are therefore confident that the

systematic process simulated by the system will be able to at least give hints about critical aspects concerning the patent application.

Patent-IT is designed as a distributed web-based system. The main actions are controlled by a production-system architecture, which includes rule bases for the different parts of the evaluation process. Some of these rules cause the establishment of a connection to a patent server in the internet in order to find relevant documents, while others retrieve ontological related terms to a keyword from an online ontology. Patent-IT is still under development, but main parts have already been realized. Further work includes the integration of the case based argumentation framework and the production system as well as the evaluation of the whole system with sample patent applications.

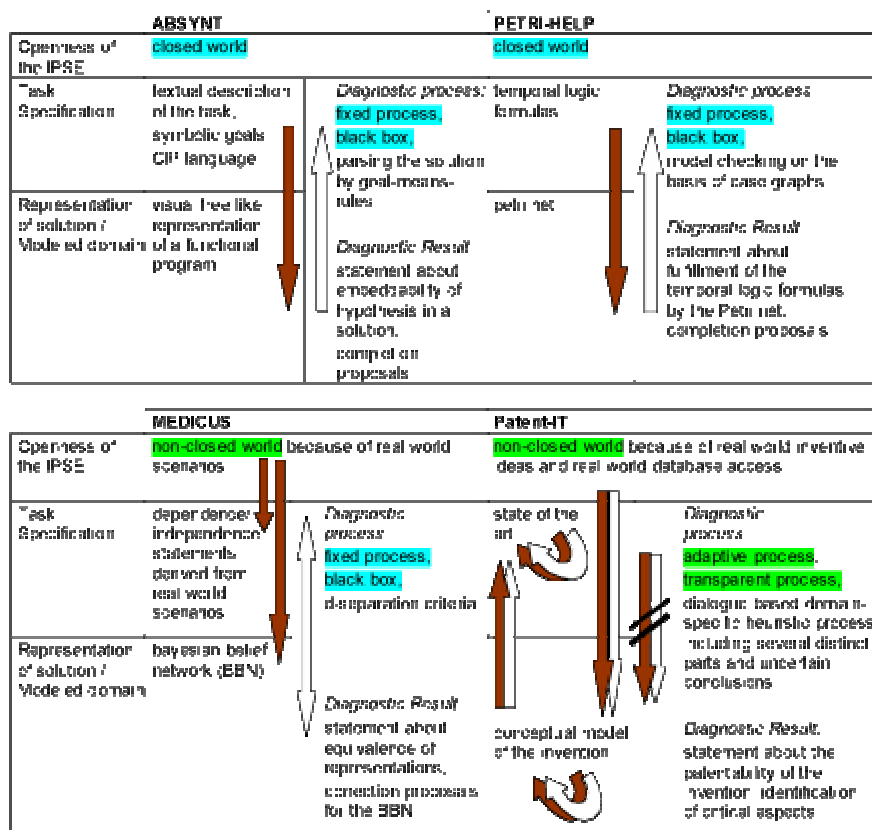


Fig. 3. The evolution of the hypotheses testing approach

4 Summary of the Hypotheses Testing Approach in Different IPSEs

Different domains have led to different realizations of the hypotheses testing approach. Fig. 3 summarizes the differences between IPSEs realized in our working group. The dark arrows indicate the users actions, the white ones, processes of the

system. The hypothesis testing approach chosen for Patent-IT is the result of the desire to improve evaluation skills in a real world domain with all its uncertainties. Supported by the IPSE, the learner develops a model of his invention and searches for the relevant state of the art. The next step is for the learner and the system to engage in a critical dialogue in order to cooperatively find clues that the invention was derivable from the state of the art and therefore is not patentable. This conclusion would make it necessary to adjust the model and to restart the process.

References

1. Anderson, J.R. (1993). *Rules of Mind*. Hillsdale: Erlbaum.
2. Bauer, F.L., Ehler, H., Horsch, A., Möller, B., Partsch, H., Paukner, O., Pepper, P. (1987). *The Munich Project CIP, Vol. II: The Program Transformation System CIP-S*. Berlin: Springer (LNCS 292).
3. 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
4. Folkers, J., Möbus, C., Schröder, O., Thole, H.J., (1996). An Intelligent Problem Solving Environment for Designing Explanation Models and for Diagnostic Reasoning in Probabilistic Domains, in: C. Frasson, G. Gauthier, A. Lesgold, *Intelligent Tutoring Systems, Proceedings of ITS'96*, Montreal, Berlin: Springer, LNCS 1086.
5. Gollwitzer, P.M. (1990). Action Phases and Mind-Sets. In E.T. Higgins, R.M. Sorrentino (eds): *Handbook of Motivation and Cognition: Foundations of Social Behavior*, 2, 53-92.
6. IBM-Patentserver: <http://patent.womplex.ibm.com/>
7. 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.
8. Möbus, C. (1995). Towards an Epistemology of Intelligent Problem Solving Environments: The Hypothesis Testing Approach I, in: J. Greer (ed.), *Artificial Intelligence in Education, Proceedings of AI-ED 95*, Washington, D.C., August 16-19, 1995, Charlottesville: AACE, 138-145.
9. Möbus, C., (1996). Towards an Epistemology on Intelligent Problem Solving Environments: The Hypothesis Testing Approach II, in: *Proceedings of EuroAIED 96*, Lisbon, Portugal, Sept. 30 - Oct. 2.
10. Möbus, C., Schröder, O. (1997). Building Domain Models by Novices in Stochastics: Towards the Probabilistic Semantics of Verbalized Stochastic Relations, in: B. du Boulay, R. Mizoguchi, *Artificial Intelligence in Education: Knowledge and Media in Learning Systems*, Amsterdam: IOS Press, pp. 394-401.
11. Newell, A. (1990). *Unified Theories of Cognition*, Cambridge: Harvard Press.
12. Partsch, H.A. (1990). *Specification and Transformation of Programs: A Formal Approach to Software Development*. Berlin: Springer.
13. Van Lehn, K. (1988). Toward a Theory of Impasse-Driven Learning. In H. Mandl, A. Lesgold (eds): *Learning Issues for Intelligent Tutoring Systems*. Springer, 19-41.
14. Willms, J., Göhler, H., Möbus, C., (1997). Testing Hypotheses in an Engineering Domain: Combining Static and Dynamic Analysis of Pneumatic Circuits, in: B. Boulay, R. Mizoguchi (eds.): *Artificial Intelligence in Education*, Amsterdam: IOS-Press, 680-682.
15. Wordnet: <http://www.cogsci.princeton.edu/~wn>