

Planning: An Intermediate Solution to the Problems in Design

J. Bravo, M. Ortega, M.A. Redondo, and C. Bravo

Castilla-La Mancha University, Paseo de la Universidad, 4
10071 – Ciudad Real - Spain
{Jose.Bravo,Manuel.Ortega,Miguel.Redondo,
Crescencio.Bravo}@uclm.es

Abstract. A great variety of simulation environments aim at supporting education through a modelling process. However, there are certain problems associated to modelling environments. We intend to build the solution model to every problem through plans based on intermediate languages outlined by the students. These learner plans are abstract solutions to the design problems. Our proposal consists in a tool, Plan Editor, to help the students design a domestic environment, following the intermediate language approach as a first step in the resolution of the problem. In the end, the student will be able to more efficiently simulate the plan proposed.

1 Introduction

A great variety of simulation environments aim at supporting education. There are obviously multiple advantages in them. But there is not an evident conclusion on their effectiveness since, while some authors show its excellences [20] [10], others do not believe in it [7][17]. However, everybody agrees that these environments motivate the learner [1], since the achievement of the goals produces their satisfaction [5] and the knowledge acquired through the simulation is more intuitive and efficient [2].

Simulation environments can be used to promote discovery learning. The concepts of exploration and discovery greatly motivate the learner who has the initiative. However, they bear the responsibility in the learning process [8]. Here, the teacher is just an adviser in those situations in which the learners doubt about their own performance [16].

There are certain problems associated to discovery environments. To mention some, we can start with the incorrect execution of the exploratory processes of learning. De Jong finds out that only 50% of the processes by discovery are fulfilled in an efficient way. Another problem is the incorrect relation between hypothesis and experiments [13]. Wason [22] demonstrated by means of a series of experiments that students do not often follow logical thought. Finally another problem we found in this type of environments is that the students' knowledge of the model is not precise enough. Besides, the degree of freedom in knowledge acquisition in this kind of learning does not suggest any ordered mechanism or guidelines that the students must follow. So, responsibility together with freedom of action can, in some cases, frustrate the learner, preventing the training goals from being successfully achieved.

There are certain simulation environments that require the previous modelling of the scenario to be simulated. At this stage we should notice the role of the computer as a student, a metaphor mentioned by Paper [15]. Here we must tell the computer what to do in order to construct the scenario where the simulation will take place. Unlike simple simulation environments, in these cases the student can modify not only input data to the system but also the model itself.

A crucial element of the constructivist learning environments is the specification and use of complex activities during the learning process [11]. It is necessary for the learner to solve subproblems of less complexity before trying the real problem [9]. Schank proposes Project-Based Learning and Goal-Based Scenarios [18][19]. These approaches present the context, the motivation and the necessary material for the learner to build specific projects in the same system.

To organise the solution to complex problems, a common approach is to establish a series of schemes or stages represented in different intermediate languages of growing precision. This produces an optimum outcome, or at least a better solution than other techniques which do not use this approach. This technique is based upon the supposition that the important thing is not only the reached solution, but also the path taken until the final product is achieved. One of the main conclusions taken from Brown's article [6], is the necessary change from the formal learning based on textbooks to the more informal "learning by doing" or to the "learning while doing" methods. The other conclusions, which are equally important, are centred on the emphasis proposed on the learning of independent abilities of the domain (for example, the generic solution of problems or the search for information) as well as of the metacognitive abilities (those related to learning how to learn, for example, drawing conclusions from our own mistakes).

The introduction of intermediate languages with the objective of finding solutions to proposed problems can be found in [21] [3]. Both authors present tutors for the learning of programming, although they lack debugging mechanisms.

In [21] a series of methods of plan composition are described. Sequencing, looping, mixing or tailoring are the ways in which each created subproblem can meet. The use of dynamics opposite to the static design implied in the use of these techniques is also recommended. Consequently, the dynamic perspective has to do with the simulation of the created assumptions.

The ideas of Soloway are developed in the work of Bonar and Cunningham [3]. These authors suggest an intelligent tutoring system (called Bridge), with the purpose of teaching how to program. This tutoring system not only finds out and informs about the mistakes committed by the student, but it partially understands the designs carried out by the student and fills the developing programs.

The intermediate languages are considered a helping step between the goals to develop and the final product as these are initially quite distinct. These languages help us write the goals and their later refinement leading to the final product. In the case of this article, the code in Pascal.

We intend to build the solution model to every problem through plans based on intermediate languages outlined by the students. These learner plans will be abstract solutions to the design problems. The planning requires a strong knowledge of the domain to study [14], so we think that the system assistance is positive during the creation of the plan. The system guides the student when he/she needs it, relying on a

case memory [12]. This case memory is made up with the plan proposals that the teacher includes in the system. These cases must have certain degrees of freedom to choose among several solutions to a particular plan.

Every plan is a sequence of instructions and actions compatible with the goals [23]. In this way, the learner mistakes in the elaboration of the model and the failure in the use of this kind of educative tools in complex domains are reduced.

In this paper we present a proposal of intermediate solution. This proposal consists in a tool called Plan Editor to help the students design a domotic environment. In the next point we present the matching between the plan proposed in the first stage and the final design made at the subsequent stage. In section 4 we focus the process on matching with the simulation environment. Finally, we give the test results of an experience made in a real learning situation.

2 The Plan as a Proposal of Intermediate Solution

Our core aim is that the student learn to plan the actions of modelling. To do this, it is necessary to have a previous knowledge in the domain to study related to the domain objects, the objects characteristics, the relations between the objects and the actions to carry out over object.

Once this knowledge has been acquired by the student, he/she will be able to build the sequence of instructions that will constitute his/her plan. These instructions are obtained through combinations of objects and actions on them. His/her plan must be in accordance with the proposal that the teacher outlined about the problem to study.

The contribution of our proposal is that the instructions of the plan are abstract actions whose aim is to allow the learner to obtain a first approximation to a modelling problem solution. Each instruction can match with several actions in the modelling phase. The solution to the problem proposed by the teacher should be flexible enough to allow the student to choose among several options.

Our solution consists of educational environments composed of Simulator and Plan Editor [4]. The Plan Editor is the tool that will facilitate the intermediate solution as a previous step towards the final solution. We should mention the permanent matching between both tools. Thus, at the end of the design the student will have the information about the possible inconsistencies between plan and design.

In this learning environment the students have a collection of problems classified in different levels of complexity. They also have a structure of the domain so that they can make use of different instruction levels to compose the steps of their plan. The plan monitoring will be carried out by means of the comparison with the plans of the expert. The revision of their incorrect actions will be recorded in the plan trace, and, finally, the plan and the scenario design will match.

3 The Plan Editor

Our study domain is Domotics, this is, the integral automation of buildings and housing. We can define it as: "A set of elements that, when installed, interconnected

and controlled automatically in a building, save the users worrying about routine everyday actions, providing improvement in their comfort, in energy consumption, in security and in communications as well".

Previous to the simulation stage, a domotic simulation environment requires a phase of design of the scenario where it will be carried out. Before this phase of design we expose our monitored planning tool so that the student can cope with this intermediate solution leading to an acceptable design of the scenario to simulate.

Speaking about domotic design implies the management of several objects. These objects are classified into the ones which are strictly objects in domotics (such as radiators, the thermostat, sensors of fire, alarms, etc.), and those backgrounds where these will be inserted.

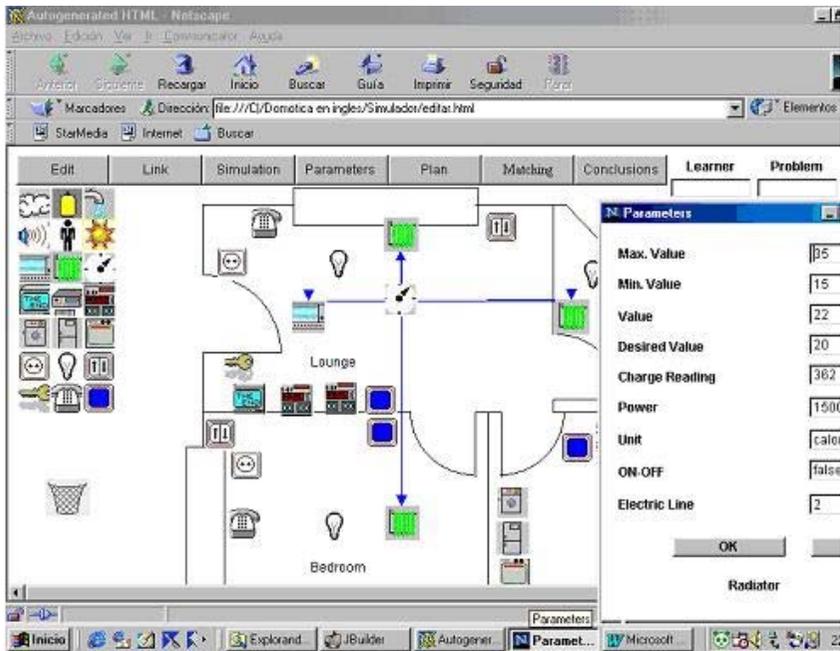


Fig. 1. Basic actions of the domotic design.

The domotic objects are grouped into the so-called management areas. There are different areas such as thermic comfort, lighting, security and the energy control. These areas include three kinds of domotic objects regarding their behaviour: the receivers that capture information, the actuators activated or inhibited when they receive a command, and the systems, taking the information from the receivers and sending commands to the actuators. This is the basic behaviour of a management area.

We have studied the actions of design as shown in Figure 1, which shows the simulation interface. In this figure we can see how the student should compose the scenario to simulate.

These actions refer to aspects such as:

- Selecting the background among a given set at the student's disposal.
- Extracting items from the tool bar and inserting them into the selected background.
- Relating them by means of visual connecting elements with compatible direction.
- Parameterisation of the domotic objects (providing the variables with their values) and, on the other hand, parameterisation of the environment.
- Showing the simulation behaviour.

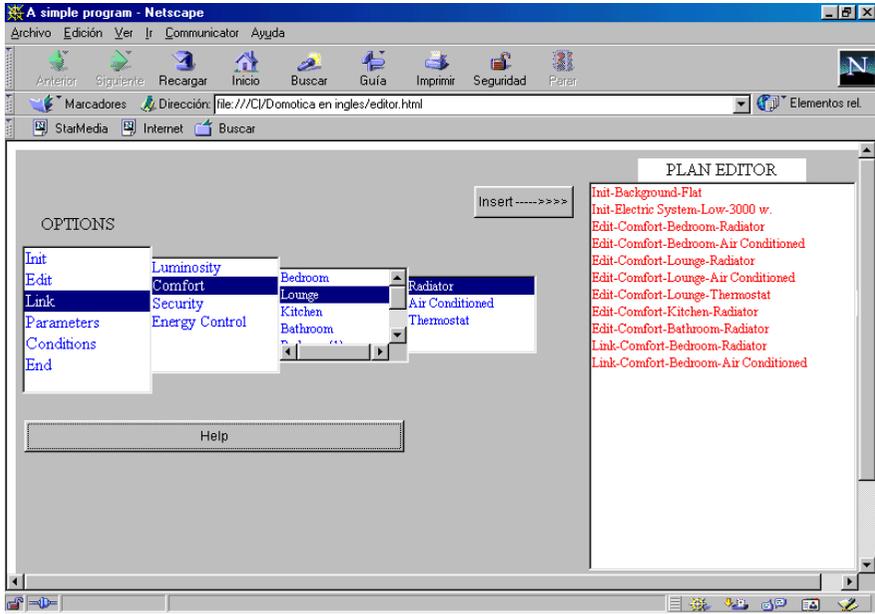


Fig. 2. The Student's Plan Editor.

Figure 2 shows the planning tool, called the Plan Editor. Here we can see the way in which the commands are included in the Student's Plan. As we can see, there is a series of instructional levels assembled in several menus. The first menu called OPTIONS contains the General Plan of Action, while the ones following can contain management areas, background items, domotic objects or variables of these objects or variables of the environment. Each menu is compounded according to the criteria mentioned before and is displayed gradually when the student selects each of the actions that it contains. Once completely displayed, the system, by means of certain mechanisms of comparison with the Expert's Plan, will accept or refuse each of the commands in the Student's Plan. In each case it will be able to request the appropriate help and, as we can also see in the figure, the system will offer some advice to the student.

Table 1 shows the Expert's Plan. The column on the left shows the sequencing of the General Plan of Action with the basic actions of design over the different parts of

the background object study case, or over the variables of the environment. The column on the right shows the domotic objects with the aforementioned characteristics of mandatory (M) or optional (O).

Table 1. Expert’s Plan (Study of the Bedroom Background Item Comfort and Energy Control).

Action[Subsystem] [Variables]	[Set of Objects][Set of Variables]
Initialise – Select Background	Bedroom
Edit – Comfort – Bedroom	Heater(M)–AirCond. (M)–Thermostat(M)
Edit – Energy Control– Bedroom	TV (O)-Video (O) - Hi-fi (O) -Plug(M)
Link – Comfort – Bedroom	Heater(M) – Air Conditioned (M)
Link – Energy Control– Bedroom	Plug(M)
Parameterise – Desired value	Thermostat (M)
Parameterise - Power/Consumption	TV (O)- Hi-fi(O)
Conditions	Exterior Temperature (M)-Load. Lines (O)

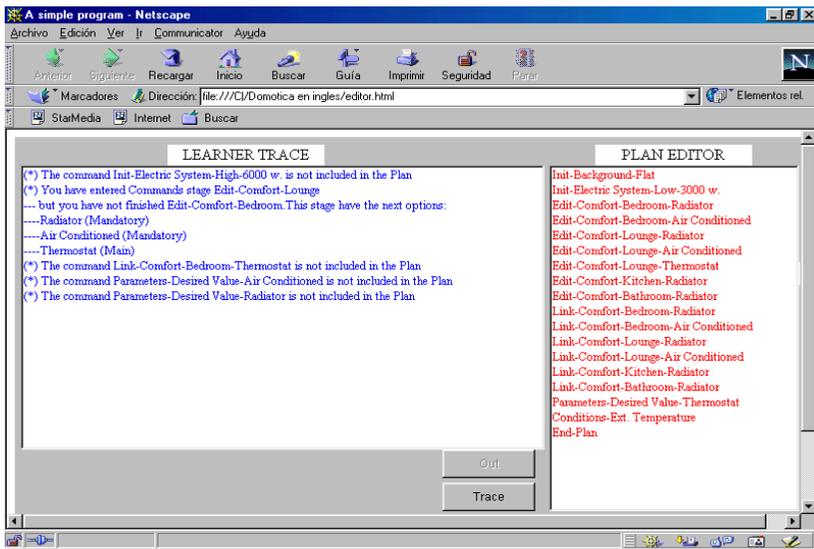


Fig. 3. Plan Trace

As an aid, the system offers the students the trace of the plan (Fig. 3), in which they are provided with the aspects of the plan where they should improve their knowledge of the domain. These aspects include warnings about the erroneous instructions they tried to include in their plan, incorrect sequencing, or some advice when including the appropriate domotic objects according to the problem to study.

In the same way (Fig. 4), in order to help the student at the design stage, the system automatically elaborates a scheme of the plan carried out. In it, the student will be able to observe the sequencing and its flexibility when inserting an item in a management area. This flexibility leads to an unordered insertion but it should be completely fulfilled.

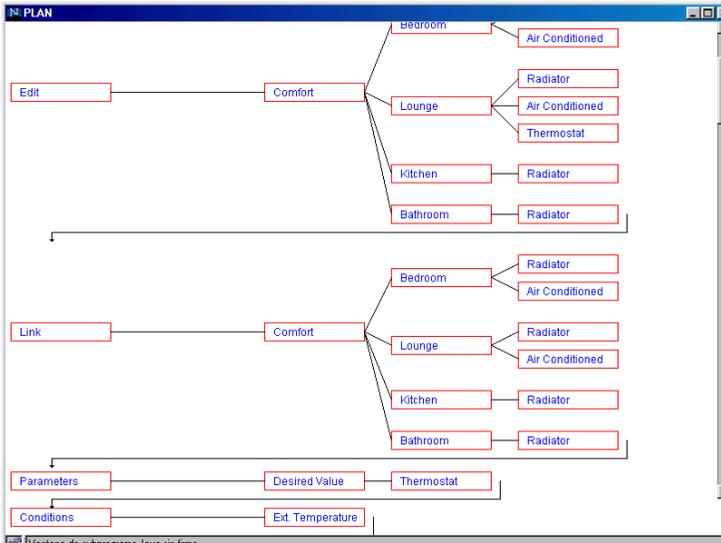


Fig. 4. Plan Outline

4 Matching with the Simulation Environment

At this point the students will have partially solved their problem. This abstract solution will make them capable of approaching the definite solution to their problem of design more confidently. Once the first approach to the design solution has been resolved, the students will be finally ready to carry out the design of the scenario to simulate in the free environment seen in fig. 1.

Therefore, the students will be completely free when undertaking the design of the scenario to simulate, although all their actions will be registered in the trace of the simulation. They will always have free access to their plan in order to proceed along the design. The matching between the planned actions and the freely fulfilled ones in the simulation environment is received by the students as a conclusion comparing the plan and the design.

Once the design has been completed in the simulation environment, the student plan should also be integrated in the working process. Therefore, their actions of free design in the simulation environment will match the ones previously planned.

As a result we propose a matching between both solutions in order to improve the knowledge of the domain in those actions with a higher level of detail, previously seen in a more abstract way in the plan. So, feedback is provided when the student considers re-planning the initial plan. Figure 5 shows the different user interfaces of the Plan Editor (up and left), the simulation environment (right) with a previously designed domotic scenario and finally the conclusions offered by the system (below) combining the design and the plan.

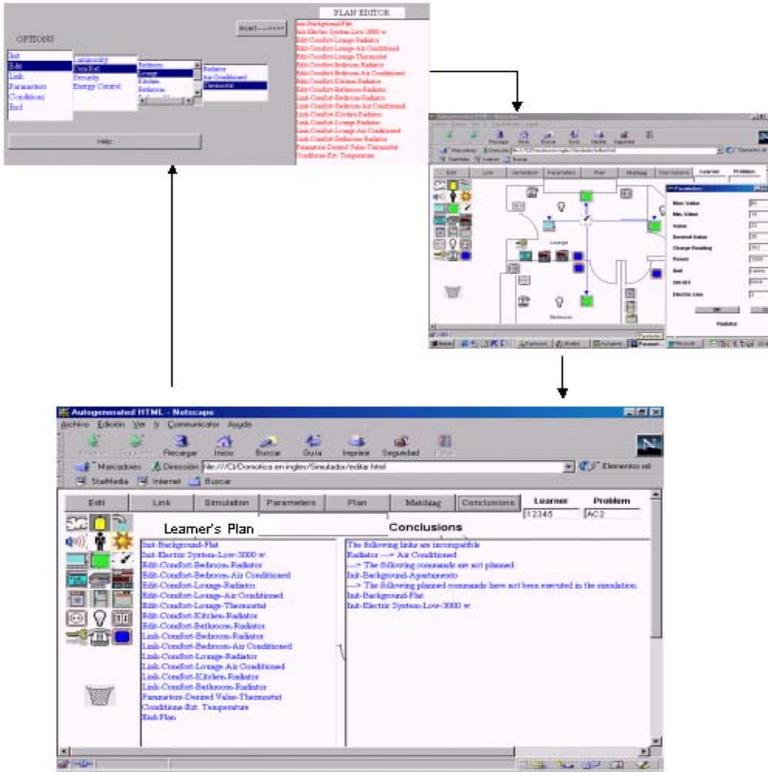


Fig. 5. Plan - Simulation Matching

The system has been designed to work on the Web. This will enable the student-teacher collaboration by including the student’s actions in both traces (the plan trace and the design trace), the student’s plan and its identification in a database available for that purpose. This database has a range of options (training levels), sets of problems, domotic items, compatibility tables, backgrounds, alternatives to the plan, etc.

5 Evaluation and Conclusions

In order to evaluate DOMOSIM-TP we used a database in which, together with all the data managing the teaching environment, the students' experiences have been recorded within that environment. So, there are tables containing data on their performance regarding these two tools, the Plan Editor and the Simulation Environment.

Regarding the exercises done by the students, we have taken two aspects into consideration. On the one hand, we have evaluated the impact of the new tool, its monitoring and consequently its acceptance. On the other hand, we have carried out a study of the contrast between the freedom of action that the Simulation Environment

offers and the planning + simulation approach to the result. The data related to this test are shown in Table 2. In it, we can see that a great percentage of students finished their plan satisfactorily, and only a small percentage did not.

Table 2. Acceptance of the Plan Editor

Students	Plans	Complete plans	Incomplete plans
87	152	122 (80%)	30 (20%)

The students have done a first exercise in the Simulation Environment without knowing about the tool used to plan their solution. Later they have carried out a second different exercise, but this time using the Plan Editor previously. The results are reflected in Table 3. With the second exercise we try to contrast the efficiency of the Plan Editor and the Simulation Environment. We can observe a considerable reduction of the mistakes appearing when using the Plan Editor previous to the Simulation Environment (79% incomplete solution without using the Plan Editor in contrast to 32 % using it). We have selected the aspects of monitoring the Student Plan and the matching between plan and design. In this sense we can see that in the aforementioned incomplete solution example, the failure in the resolution of each problem is reduced in more than a half. Other more precise aspects of this monitoring, such as the problem mistakes and sequence and repetition errors are also considerably reduced.

Table 3. Efficiency of use of the model

Simulation Environment				Plan Editor + Simulation Environment			
Incomplete Solution	Problem Errors	Sequence Errors	Repetition Errors	Incomplete Solution	Problem Errors	Sequence Errors	Repetition Errors
19 (79%)	11 (49%)	8 (33%)	9 (38%)	10 (32%)	3 (10%)	5 (16%)	3 (10%)

We think that building a plan, i.e., an intermediate solution, helps the student to break the complexity of the modelling task and make learning through this kind of simulation environments much more effective. The students first build a more abstract, and consequently easier, solution and then, with an adequate design plan, they proceed to build a model. This model is the final solution to be tested by simulation.

Our proposal for the future includes peer collaboration for planning as much as for simulation. Members of our team are working to deal with this approach so that groups of students can work together in solving both the intermediate solution (plan) and the final solution (model). We think this collaborative planning and simulation can improve the final solution due to the positive interaction that the classmates can produce through collaboration and negotiation.

References

1. Ajewole, G.A. (1991), "Effects of discovery and expository instructional methods on the attitude of students to biology". *Journal of Research in Science Teaching*, 28, 301–409.
2. Berry, D.C. & Broadbent, D.E. (1984), "Explanation and verbalization in a computer-assisted search task". *The Quarterly Journal of Experimental Psychology*, 386, 209–231.

3. Bonar, J. G. & Cunningham, R. (1988), "Intelligent Tutoring with Intermediate Representations" ITS-88 Montreal.
4. Bravo Rodríguez, J. (1999), "Design Planning in Simulation Environment for Distance Learning". Doctoral Dissertation, Madrid.
5. Brennenstuhl, D.C. (1975), "Cognitive versus effective gains in computer simulations". *Simulation & Games*, 6, 303–311.
6. Brown, J.S., (1983), "Process vs product: a perspective on tools for communal and informal electronic learning", report from "The learning Lab: Education in the electronic age".
7. Carlsen, D.D. & Andre, T. (1992), "Use of a microcomputer simulation and conceptual change text to overcome students preconceptions about electric circuits". *Journal of Computer-Based Instructions*, 19, 105–109.
8. de Jong, T., van Joolingen, W., Pieters, J. & van der Hulst, Anja. (1993), "Why is discovery learning so difficult? and what can we do about it?". EARLI conference. Aix-en-Provence.
9. Duffy, T., Jonassen, D. (1992), "Constructivism and the Technology of instruction". Lawrence Erlbaum Associates, Hillsdale, New Jersey.
10. Faryniarz, J.V. & Lockwood, L.G. (1992), "Effectiveness of microcomputer simulations in stimulating environmental problem solving by community college students". *Journal of Research in Science*.
11. Fishman, B.J., Honebein, P.C., Duffy, T.M. (1991), "Constructivism and the design of learning environments: Context and authentic activities for learning". NATO Advanced Workshop on the design of Constructivism Learning.
12. Hammond, K.J. (1990), "Case-Based Planning". In *Cognitive Science*, vol 14, pp. 385–443.
13. Lavoie, D.R. & Good, R. (1988), "The nature and use of predictions skills in a biological computer simulation". *Journal of Research in Science Teaching*, 25(5), pag. 335–360.
14. Minton, S. & Zweben (1993), "Learning, Planning and Scheduling: An Overview". *Machine Learning Methods for Planning*. Minton (Ed.) Morgan Kaufmann.
15. Paper, S. (1980), "Mindstorms: Children, Computer and Powerful Ideas". Basic Books Inc., New York.
16. Paper, S. (1987), "Computers in Education: Conceptual Issues". Saphiro, S., *Encyclopaedia of Artificial Intelligence*, Edit. Willey, New York.
17. Rivers, R.H. & Vockell, E. (1987), "Computer simulations to stimulate scientific problem solving". *Journal of Research in Science Teaching*, 24, 403–415.
18. Schank, R. Clearcy, C., (1994), "Engines for Education". Lawrence Erlbaum Associates, Hillsdale, New Jersey. [http:// www.ils.nwu.edu/~e_for_e](http://www.ils.nwu.edu/~e_for_e)
19. Schank, R., Kass, A., "A Goal-Based Scenario for Higher School Students". *Comm. of the ACM*, 39(4), 28 (April-1996).
20. Shute, V.J. (1990), "A comparison of inductive and deductive learning environments: Which is better for whom and why?". Paper presented at the American Educational Research Association (AERA) Annual Meeting, Boston. USA.
21. Soloway, E. (1986), "Learning to Program = Learning to Construct Mechanisms and Explanations". *Communications of the ACM*.
22. Wason, P.C. (1966), "Reasoning". In Foss, B.M. (Ed.). *Nes Horizons in Psychology*. Harmondsworth, England: Penguin.
23. Wasson, B. (1990), "Determining de Focus if Instruction: Content Planning for Intelligent Tutoring System" Doctoral Thesis, Dep. Computational Science, University of Saskatchewan.