

PLANNING IN PROBLEM SOLVING: A CASE STUDY IN DOMOTICS

José Bravo, Manuel Ortega¹ and Felisa Verdejo²

Abstract - Simulation environments requiring the creation of a model previous to the simulation stage raise a challenge for the student. In these environments the realization of the model itself, which is the basis for most part of the richness of the acquired knowledge, is a difficult step. This is the reason why in some cases the learning goal is not achieved. We consider of great importance to cope with this problem so that better support in model building could improve the mere free performance of the student. In this sense, we propose to include, before beginning the construction of the model, a tool allowing the expression of a first approach to the solution at a more abstract level and, consequently, easier to obtain. Finally, it is important to point out the necessity of collaboration of both the teacher and the student. In order to facilitate this, we will define data structures containing, among other aspects, the student's performance. This performance could be revised by the teacher, who will advise students where to reinforce their knowledge on the modeling process. This kind of collaboration can also provide an effective support for distance learning situations.

INTRODUCTION

There are a great number of authors that propose simulation environments as effective tools for learning. Many of these environments require building the model first, so that the student can check the laws of the domain to study by means of simulation. Students do not only interact through the variables of the model but they can also change the model itself. This approach is a change from the traditional learning based on textbooks to "learning by doing" or to the "learning while doing" methods. But modeling is often a difficult task, and without instructional support, the students are frequently unable to reach the goal.

Some authors have proposed to overcome this problem by using intermediate representations for problem solving. It is the case in [1,8] for the learning of programming. Both articles show a problem solving approach where a solution is built through several abstraction levels, using intermediate languages of growing precision until the result, a computer program, is obtained. This technique is based upon the assumption that the important thing is not only the final

solution, but also the steps taken until the result is achieved [3]. The theory underlying these intermediate representations (problem solution schemas) comes from the field of Cognitive Psychology: knowledge is represented as schemas and the reasoning techniques using these schemas can be formulated in terms of goals and plans.

In this paper we present a planning tool as an intermediate solution to help students to design a domotic environment. This tool mainly consists of a Plan Editor for the Student Plan that should match the solution proposed by the expert. Finally, we give the test results. It has been proved efficient as a successful resource in the learning process.

A PROPOSAL FOR INTERMEDIATE SOLUTION

We will outline the different aspects we have considered in order to build a tool for supporting students along the successive problem solving stages. A key idea has been to monitor students actions along the process. In this way their intermediate solutions can be examined, and therefore more focused help can be provided.

These aspects define the scope of the tool, and are discussed below:

The task. We are dealing with a kind of modeling, the scenario design. Therefore the idea of an intermediate solution will be a plan of design. This plan at a level of abstraction will be simple but useful enough to support the student in the next step: to carry out this plan using the basic modeling operations offered by the simulator.

The application domain.-

The following aspects of a domain should be taken into account for simulation purposes:

1. *Objects of the Domain* .- Relevant knowledge about the different types of objects in the domain, as well as their structure.
2. *Parameterization of the Object Variables* .- Knowledge about relevant variables of the domain objects. Changes in these variables will affect the properties of the model.
3. *Relationships between objects*.- The type of relations, as well as their constraints. Relationships between the objects will generally be responsible for the modeling behavior.

¹ Grupo CHICO; Universidad de Castilla La Mancha – Spain, mortega@uclm.es

² Universidad Nacional de Educación a Distancia - Spain

There is a range of applications to study. First of all, we will mention Domotics. As we will see, this is our case study, but similar ones can be mentioned such as the modeling of electronic circuits, digital circuits, etc.

Actions for modeling (design) .- These actions will be the ones that the student carries out when building the model. Previous to the modeling phase, students have to know about the domain objects and their relationships.

A description of the generic actions for modeling (Fig.1) follows:

1. **Initialization** .- It is the action where model requirements (for some basic parameters) are defined.

2. **Selection**.- An object (from a variety of objects) is selected to be part of the model.

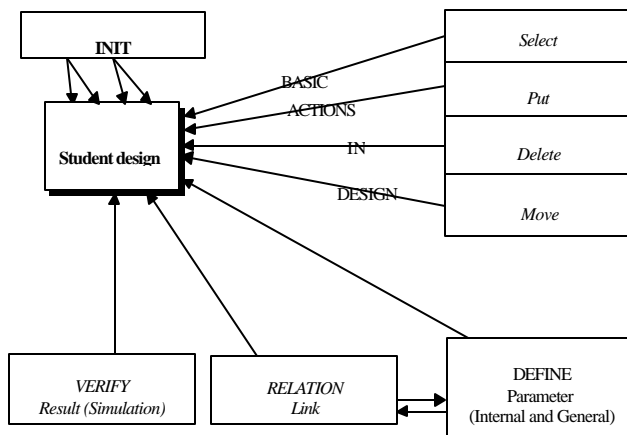


Fig. 1.- Basic actions of design

3. **Insertion**.- The object selected is inserted in the scenario where the model is carried out and in which finally the behavior will be observed by simulation.

4. **Relation**.- It will consist of the actions of linking compatible elements. These relationships are the base of the model behavior.

5. **Parameterization**.- Values are defined for the variables of an object. This point is also important for the model behavior.

6. **Verification**.- The confirmation that the model has been carried out correctly.

The planning tool

A problem can be solved in different ways, so the planning tool should be flexible and permit certain degrees of freedom to the student or, alternatively, there must be different solutions among which the students could choose the one that seems more appropriate to them.

Insertion of the necessary elements according to the model under construction.- Several choices must be supplied to enable the diverse elements to form a good solution from the different ones possible.

Sub-task Completion.- It is advisable, if there are groupings of the domain objects, to try to complete the modeling of these before going on to another task.

Set of models (problems) .- A collection of problems, structured in levels of growing complexity will be proposed to the student. This collection will support the progressive learning of the domain to study.

The collection includes different solutions for each problem. Based on this knowledge, a matching mechanism capable of comparing intermediate solutions can be used for monitoring purposes. More focused help can be provided along the problem solving process.

Structuring of the domain .- By combining the actions of modeling on the different domain objects we can reach the different instructional levels in order to construct the model of the problem to study.

The student's performance .- A performance trace would allow some automatic feedback. The teacher will also have access to this data in order to provide students with some advice for further improvement.

Continuity between intermediate and final solutions .- The planning tool should be integrated in the simulation environment in order to link the intermediate solution or plan with the modeling actions in the simulator. It will be essential to match the abstract actions of the plan with the corresponding ones of modeling at a greater level of detail.

From the correspondence between both solutions, intermediate and final, conclusions can be established so that the student can check the efficiency of his/her plan and take re-planning, if necessary, into consideration.

THE APPLICATION DOMAIN: DOMOTICS

Our application domain is Domotics, i.e. 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 to build the model to be simulated. Our approach establishes two phases: building a design plan first, and then proceeding to build the complete model following that plan. For the phase of designing a plan we provide an interactive tool so that the student can build a plan of design incrementally. The tool is not only an editor; it also includes certain facilities to support the student and give some advice on the planning process. Building the model is performed afterwards. The students have to deploy and refine their plan using the modeling interface (shown in fig 2) in order to create a complete scenario. The scenario, fully defined, will be the solution to be finally checked by the simulator.

Domotic modeling involves the management of several objects. These objects are classified into the ones that are strictly domotic (such as radiators, thermostats, sensors of fire, alarms, etc.), and the layouts of the rooms where these objects will be inserted.

The actions for building a model, as shown in Figure 2, are provided by the simulation environment. We can see the options (edit, link, parameters) on the upper part, and the selection of object icons on the left. An example of model can be seen in the middle of the figure. In this example the problem to be solved includes the layouts of different rooms and several domotic objects linked by arrows. The deployed form on the right shows the values that have been defined for the parameters of one of the objects.

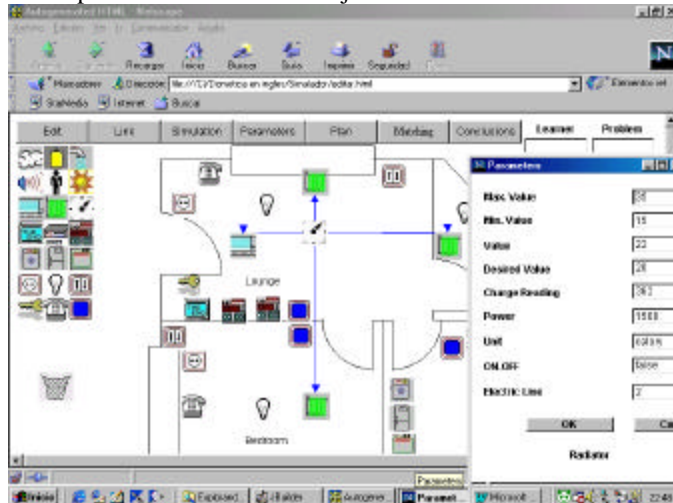


Fig. 2.- The interface for building a model in the simulation environment.

With the options provided by the environment, the student can perform the following actions:

1. Selection of a layout, from a set of possible choices. This set depends on the kind of problem to be solved.
2. Selection of objects from the icons and their insertion in the model. The objects have to be located in a position in one of the room layouts .
3. Linking the objects in the model, to express their relationships. Links are defined with a direction and further constraints, and this information will determine the model behavior in the simulation process.
4. Parameterization of the domotic objects (assigning values to variables) and, additionally, parameterization of the physical environment. The values will depend on the problem to solve and they deserve a level of detail that is not dealt with in the planning phase but refined at the modeling stage.
5. Relationship of the domotic objects with the physical environment. These relationships express constraints on the internal object variables due to the external environment variables and these interactions define also the model behavior for the simulation process.

DOMOSIM-TP: THE LEARNING ENVIRONMENT

The learning environment consists of the Student's Plan Editor, the tool for building a domotic design plan, and the

simulator, a free modeling environment where the student will carry out such a design according to a previous plan.

The diagram of the system is shown in fig. 3. Here we can see the two independent parts: on the left, the plan editor, and on the right the simulation tool. However, they are related, and this relation is represented in the figure by a link with the label *matching* .

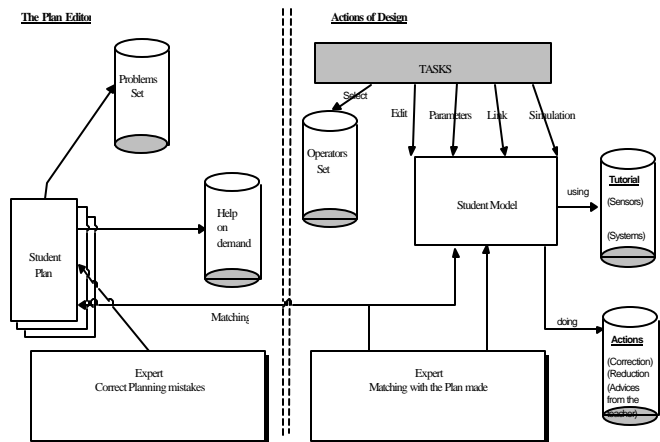


Fig. 3.- Structure of the teaching environment (DOMOSIM-TPC)

The planning tool enables the student to design a plan in an incremental way. The system includes a collection of problems. For each problem there is a statement describing the problem as well as the solution proposed by an expert. This solution is not unique, so the students have certain degree of freedom in planning their solution. The planning tool includes a supervisor module allowing or not the insertion of actions in the plan. This supervisor provides some advice as well to perform the planning process.

In figure 3, on the right, the simulator components (free modeling environment) are represented. On one side the actions for building the model (edit, link, parameters, and test the model (simulation)). On the other, an advisor module to check whether the student is following or not his/her design plan, and finally the tutorial component providing further help on demand.

THE PLAN EDITOR: A TOOL FOR AN INTERMEDIATE SOLUTION

In this section, we describe in detail the tool for planning, i.e., the Student's Plan Editor.

The students are identified so that a trace of their performance can be recorded in a database. In this way the teacher, even remotely, looking at this information, can give advice to students on the aspects which require further knowledge reinforcement.

The system provides a collection of problems to be solved. These problems are organized in levels of complexity. The student has first to select a level. These two aspects can be seen in figure 4.

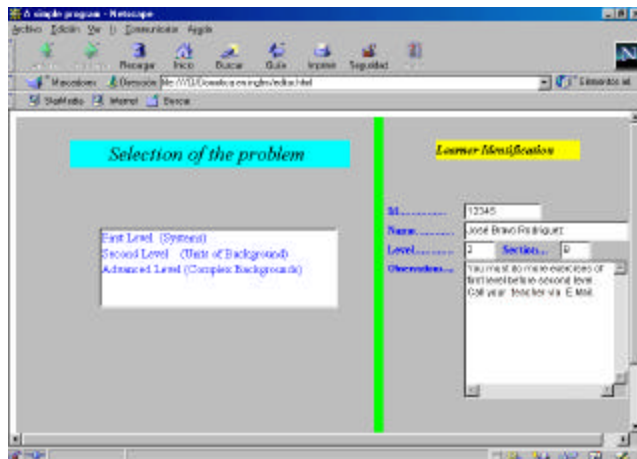


Fig. 4.- Identification of the student and selection of the level

The system allows adding more levels by including them and their associate problems in the Database.

In order to handle the complexity of the problem solving task for the application domain, we have organized the collection of problems into three levels. The first level, the most basic one, is devoted to the study of the domotic objects and their relationships. From these relationships we can extract their behavior when they are inserted in their corresponding layout.

In the second level both, layouts and domotic objects, come into play. Each component of the layout (the lounge, a bedroom, the kitchen, etc) is considered as an independent unit. Finally, the third level takes into account these units as a whole, studying the complete layout, to establish the global domotic solution.

Once the student has been identified and the level of the problem selected, the system offers multiple combinations of different groupings of domotic elements (layouts and management areas) generating a variety of problems, and proposing one of them to the student.

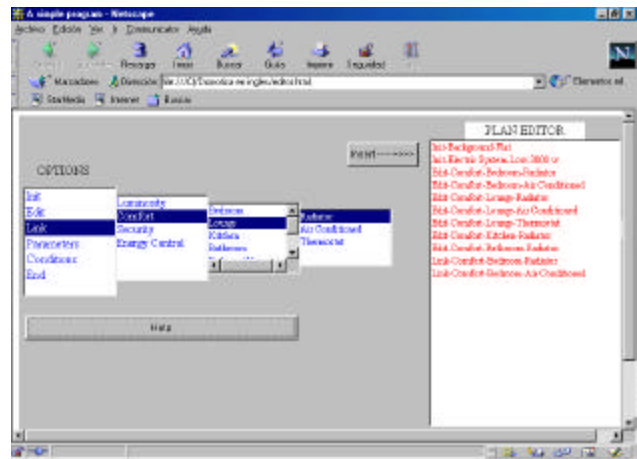


Fig. 5.- The Student's Plan Editor

Figure 5 shows the Plan Editor Interface. Here we can see the way in which the plan can be built incrementally. On the left the possible actions to include in a plan can be selected. The first menu called OPTIONS contains the different general actions. When one is selected, the other menus deploy further options to select: a management area to consider (such as luminosity, comfort, security and energy control), a kind of layout (such as bedroom, kitchen, ...) and domotic objects to be included or the conditions for external environment variables (such as temperature).

Each menu is displayed gradually when the student selects an option. Once the selection is completely finished, for instance *Edit comfort bedroom radiator* the system compares this action with the Expert's Plan, accepting or rejecting its inclusion in the Student's Plan. In each case the student can request for appropriate help and, as we can see in figure 6, the system will offer some advice to the student.

Table 1 shows an example of an Expert's Plan. The column on the left shows the sequencing of design actions over the different parts of the layout, or over the variables of the environment. The column on the right shows the domotic objects, and for each of them, the attribute of being mandatory (O) or optional (P) is indicated.

Table 1.- Expert's Plan (Study of the Bedroom Layout for Comfort and Energy Control).

Action [Subsystem] [Variables]	[Set of Objects] [Set of Variables]
Initialize – Select Layout	Bedroom
Edit – Comfort – Bedroom	Heater(O)–Air Conditioned(O)–Thermostat(O)
Edit – Energy Control– Bedroom	TV (P)-Video (P) - Hi-fi (P) -Plug(O)
Link – Comfort – Bedroom	Heater(O) – Air Conditioned (O)
Link – Energy Control– Bedroom	Plug(O)
Parameterize – Desired value	Thermostat (O)
Parameterize - Power/Consumption	TV (P)- Hi-fi(P)
Conditions	Exterior Temperature (O) - Loading Lines (P)

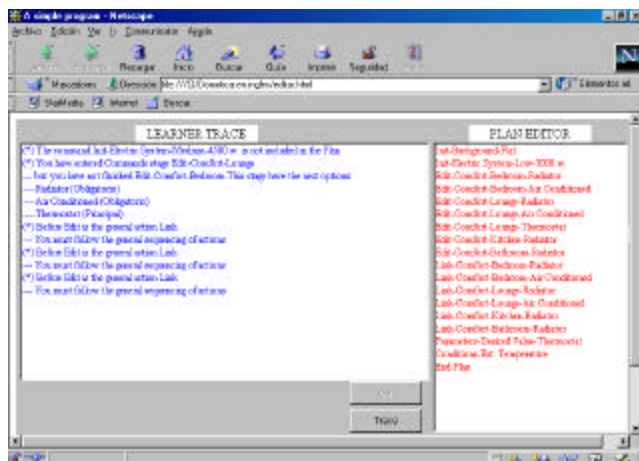


Fig. 6.- The Plan Trace

The system provides the students with comments on their performance (Fig. 6). These comments include warnings when they try to include wrong actions in their plan, or in the sequencing, as well as some advice for selecting the appropriate domotic objects according to the problem to be solved. In addition, it indicates when a proposed action does not belong to the expert plan.

As we can see in fig. 6, all these comments appear under the title learner trace, indicating the points where there has been some incompatibility with the Expert's Plan. The incompatibility here is related to the general sequencing of actions that the student should have followed. The sequencing obliges to plan the design in a correct order inserting the different domotic elements in each of the rooms of the house. It also includes a partial sequencing of the domotic management areas, that is to say, the insertion, if not ordered at least complete, of the elements assembled in such areas.

The learner trace is recorded in a Database that additionally contains all the instructional elements of the Plan Editor: menus, levels, set of problems, etc. In this way the teacher can asynchronously collaborate with the students indicating to them possible misconceptions.

In order to help the student at the design stage, the system automatically elaborates a scheme of the plan. In it, the student will be able to observe the sequencing and its flexibility at the moment of inserting an item in a management area. Parallel paths indicate that the order is not relevant, but all of them should be completely fulfilled.

The graphic representation of the plan offers a global view of the actions and their order. Sometimes it is a complete order such as in the figure for Edit, Link, Parameters, Conditions, etc., and in parallel, where the order is not important such as for Comfort in Bedroom, Lounge, Kitchen or Bathroom, but once one of them selected, Lounge for example, this level has to be completed, here inserting Radiator, Thermostat and Air Conditioned in any order. Then, the student should complete another parallel group

and so on until completing all of them and then proceed to the next point of the general plan, the definition of parameters in this case.

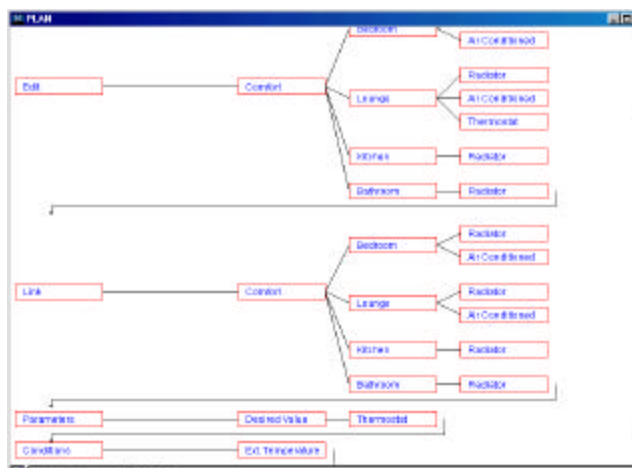


Fig. 7.- The Plan Outline

EVALUATION OF THE DOMOSIM-TP TEACHING ENVIRONMENT

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

The exercises that the students have done have two aspects. On the one hand the impact of the new tool has been studied, their monitoring and consequently its acceptance contrasting with the freedom of action that the Simulation Environment offers. 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 new tool

Students	Plans	Complete plans	Incomplete plans
82	141	118 (83%)	29 (17%)

With the second exercise we try to contrast the efficiency of the Plan Editor and the Simulation Environment. On the one hand the students have done a first exercise in the Simulation Environment without knowledge of the tool in order to plan their solution. Later they have carried out a second different exercise, but this time having used the Plan Editor previously. The results are reflected in Table 3.

Exercises	Students	Simulation Environment	Plan Editor + Simulation Editor
45 (32%)	32 (40%)	18 (40%)	27 (60%)

Simulation Environment				Plan Editor + Simulation Environment				
Incomplete Solution	Problem Errors	Sequence Errors	Repetition Errors	Incomplete Solution	Sequence Errors	Repetition Errors	Planned and not Simulated	Simulated and not Planned
15	15	24	19	9	6	2	6	0

Table 5.- Efficiency of the use of the new tool

From this, we can deduce that the use of the Plan Editor previous to the use of the Simulation Environment reduces the number of errors when completing the problem in a 60%; it reduces the errors in a 75% in the sequencing of actions and in a 90% in the repetition of commands. What is more, the errors made when including elements that do not correspond to our problem in our design (Problem Errors = 15) are reduced in a 60%.

CONCLUSIONS AND FUTURE WORK

We think that building a plan, i.e., an intermediate solution, helps the student to break the complexity of the modeling 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).

For a demonstration of DOMOSIM-TP you can connect to: <http://chico.inf-cr.uclm.es/jbravo/domosim/inicio.html>

REFERENCES

[1] Jeffrey G. Bonar & Robert Cunningham. (1988), *“Intelligent Tutoring with Intermediate Representations”* ITS-88 Montreal.

[2] Bravo Rodríguez, J. (1999), *“Planificación del diseño en entornos de simulación para el aprendizaje a distancia”*. Doctoral Dissertation, Madrid.

[3] 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”*.

[4] 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*.

[5] Paper, S. (1980), *“Mindstorms: Children, Computer and Powerful Ideas”*. Basic Books Inc., New York.

[6] Paper, S. (1987), “Computers in Education: Conceptual Issues”. Saphiro, S., *Encyclopedia of Artificial Intelligence*, Edit. Willey, New York.

[7] Rivers, R.H. & Vockell, E. (1987), “Computer simulations to stimulate scientific problem solving”. *Journal of Research in Science Teaching*, 24, 403-415.

[8] Elliot Soloway (1986), “Learning to Program = Learning to Construct Mechanisms and Explanations”. *Communications of the ACM*.