An environment for constructing and exploring visual models of logic propositions by young students

Christos Fidas[1], Panagiotis Politis[2], Vassilis Komis[1], Nikolaos Avouris[1]

[1] University of Patras, 26500 Rio Patras, Greece
[2] University of Thessaly, Volos, Greece,
fidas@ee.upatras.gr, ppol@uth.gr, {komis, avouris}@upatras.gr

3

Abstract. This paper presents the main characteristics of Logic Models Creator (LMC). LMC is a new educational environment for young students to build and explore logic models in graphical form. LMC allows visual representation of logic models using IF/THEN/ELSE constructs. In this paper we provide an overview of LMC architecture and discuss briefly an example of use of LMC. As discussed, LMC users in the reported case study managed to achieve effective communication and task evaluation during exploration of problems involving decision making.

1 Introduction

Logic Models Creator (LMC) is a new learning environment which supports logic modeling activities for students of 11 to 16 years old. LMC is a derivative of the decision support component of an earlier modeling environment, ModelsCreator version 2.0 (MCv2), originally built as a tool to be used for qualitative and semi-quantitative reasoning with real world concepts [1]. The original Decision Support component of ModelsCreator included a validation and model diagnosis module described in [2]. The limitations of that module have been tackled in LMC, as discussed more extensively in [3]. The logic propositions that can be built and explored with LMC meet the requirements of many curriculum subject matters, like mathematics, science etc., permitting interdisciplinary use of the logic modeling activity. LMC puts great emphasis on visualization of the modeling entities, their properties and their relations. In fig. 1 an example of a model built using LMC is shown. On the left the hypothesis is visualized and on the right hand side the conclusion of the logical proposition. In this example the conditions are tested for helping somebody that has received an electric shock.

An important aspect of LMC, as with the original MCv2, is its open character regarding the ability provided to the teachers to create new logical domains (i.e. new subject matters) as well as new primitive entities which are needed for the creation of the logic models.

In the rest of this paper we present first the architecture and basic functionality of the LMC environment. We describe then an example of use of LMC by groups of young students and discuss the implications.

A version of this paper is published in Proceedings SETN'06, Heraklion, Crete, May 2006, (ed. G. Antoniou et al.), Lecture Notes in AI, Springer.

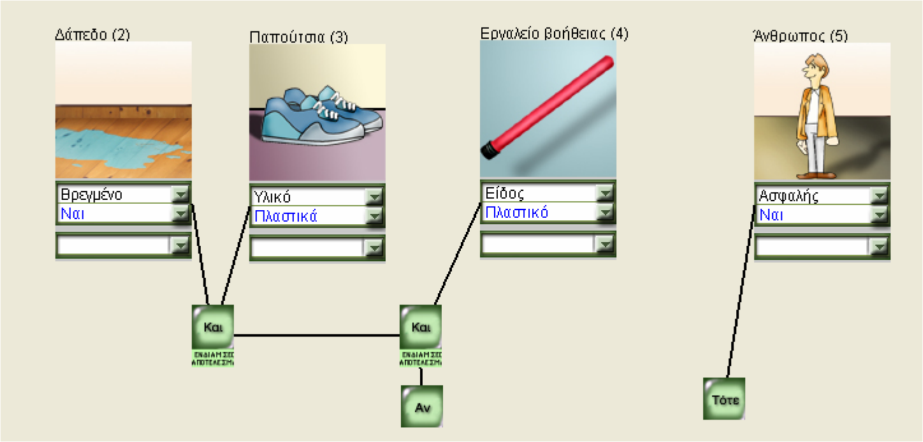

Figure 1:A typical logic model : A solution to the “the electric shock” problem: IF FLOOR=WET AND SHOES=RUBBER AND STICK=PLASTIC, THEN MAN=SAFE

2 The Logic Models Creator (LMC) environment

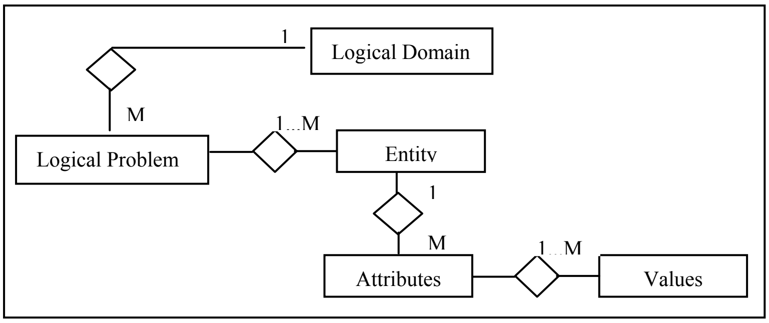

Figure 2.: Representation Scheme of Logical Models

A typical LMC model to solve a given problem may be express according to the following syntax:

Proposition = *IF* Construct *THEN* Construct
| *IF* Construct *THEN* Construct *ELSE* Construct
Construct = (Construct *AND* Construct)
| (Construct *OR* Construct)
| *NOT*(Construct)
| Attribute=Value

The system considers two basic user categories: (a) the students and (b) the teachers, who interact with the visual environment in order to accomplish specific tasks. While the main task of the students is to build and check the correctness of their logical models the main task of the teachers is to create new logical domains and define the reference models, i.e. valid expressions for a given problem.

An important aspect of LMC is its open character regarding the ability provided to the teachers of creating new logical domains (i.e. new subject matters) as well as new primitive entities which are needed for the creation of the logical models. Each logical domain represents one or more logical problems which describes decision making concepts that the students must explore. For each logical problem the teacher can create entities which describe textually and visually the concepts included in the problem domain (e.g. the stick, the floor, etc.). Each entity may include a set of a attributes which describe specific characteristics of the entity. Furthermore to each attribute of an entity can be assigned one more possible values. These values belong to a set which is defined in the creation phase of an entity. Figure 2 depicts the main domain entities that define a logical problem.

In the frame of a specific domain a teacher can define more than one Reference Models against which the students models will be subsequently tested. We consider these as models which describe alternative correct solutions to a given logical problem. The Reference Model should not violate rules related with the syntax of logical propositions of LMC. The knowledge representation used for expressing the Reference Model has been a matter of discussion during development of LMC and the previous environment MC. As discussed in [3], a first attempt was to express the Reference Model through logical propositions, however this approach produced rigid logical models. An alternative proposed here is to use Truth Tables for representation of the Reference Model.

So for each Reference Model in LMC a teacher must complete a Truth Table which contains all the combinations of different events that exists in the Reference Model. If the hypothesis graph connects a set of $\{G_1,\ldots,G_N\}$ events of different attributes and each attribute can take values from a set $\{G_{11},...,G_{1M})$ than the whole set of different states in which the hypothesis graph can be found is the Cartesian product $G_{if}=\{G_{1M}\, x\, G_{2M}\, x\, \ldots G_{NM}\}$

In a similar way we can define the decision part of the statement. If it connects a set of $\{1,...,M\}$ events of different attributes then we add to the truth table M columns which will be associated with values by the teacher. These values of the attributes in the decision graph depend on the values of the attributes of the hypothesis graph.

We consider that a *logical problem* consists of a set of correct logical models L $L=\{l1,...,lN\}$. The purpose of the system is to support the student with appropriate feedback in order to build a model that is equal to a correct logical model in the knowledge base of the active logical domain. To attain this aim, the system creates and displays messages using a relevance factor.

In the case that the student model is equal to a model in set L a message is produced in order to inform the student about the correctness of his model. In any other case the system has diagnosed that the student model has no equal model in set L, it sttempts to find a model in set L which is similar to the students model. With the aim of achieving this goal, the validation module scores each model in set L in relation to the student model

The logical model with the highest score is defined as the closest logical model to the developed student model. The validation module of LMC checks the level of similarity at (a) the Entities level (b) the Attributes level (c) the Attributes values level and (d) the Relations level and provides the student with appropriate feedback messages in order to support and scaffold the modelling process.

3 LMC in use: some first findings and conclusions

In a case study that had as the main objective to validate the use of this environment by typical users, a pair of two 11-year old students were asked to explore a logical model under the supervision of their teacher. The model is based on a scenario of a dog that is in conduct with a live wire and received an electric shock. The children were asked to investigate the conditions under which they could safely rescue the dog. The mode includes attributes like Material of the *stick* to touch the dog, Material of the *shoes* of the child, Material and condition of the *floor* (see figure 1). The teacher asked the students to investigate various alternatives and to check the validity of the model. The session that lasted one hour was recorded and subsequently analyzed using a dialogue annotation scheme. An extract of the dialogue follows:

Instructor: Well, let's try the iron stick.
Student A: OK, I select iron.
Student B: The iron is the worst choice!
Student A: Hmm, you are right; the iron is the worst...

An interesting finding of the study was that the two children were engaged in dialogue with the LMC environment and discussed their own experiences related to the subject domain. They investigated for instance the conducting capability of materials like plastic and rubber in relation to the shoes and inferred that plastic is insulating material, as in cables of household electric appliances. One of the children recalled that her grandmother received a strong shock when she touched a bare live cable. The messages received by LMC were considered relevant and supported the specific task.

The children seemed to trust the software environment when they engaged in dialogue with it and expressed their wish to further interact with models of this nature in other subject domains. Despite the fact that the children of this age group were lacking strong conceptual models of the domain, they managed to reason about it with the support of LMC.

References

1. Fidas, C., Komis, V., Avouris, N., Dimitracopoulou, A.: Collaborative Problem Solving using an Open Modelling Environment. In G. Stahl (ed.), Proc. CSCL 2002, Boulder, Colorado, USA, Lawrence Erlbaum Associates, Inc., (2002) 654-655
2. Partsakoulakis I., Vouros G.. Helping Young Students Reach Valid Decisions Through Model Checking. Proc. 3td ETPE Conf., pp. 669-678, Rhodes, Greece, 2002.
3. Fidas C., Avouris N., Komis V., Politis P., On supporting young students in visual logic modeling, Proc. 3rd IFIP Conference on Artificial Intelligence Applications & Innovations (AIAI) 2006, Athens, June 2006.