

# Organizing Problem Solving Activities for Synchronous Collaborative Learning of Design Domains

Crescencio Bravo, Miguel A. Redondo, Manuel Ortega, and José Bravo

Departamento de Informática. Universidad de Castilla - La Mancha  
Paseo de la Universidad, 4. 13071 Ciudad Real (Spain)  
{Crescencio.Bravo,Miguel.Redondo,Manuel.Ortega,  
Jose.Bravo}@uclm.es

**Abstract.** Scientific community is showing a growing interest in Collaborative Systems, although most of developed systems deal with specific applications. This, together with the effort that implies the construction of complete environments, leads us to explore the application of specification techniques to model some entities of CSCL/CSCW systems, such as the approached domain or the problems to solve. In this work this task is carried out starting from DomoSim-TPC, a web-based CSCL environment for distance education of domotical design. First, its positions are generalized, and next, a specification that will allow users to generate software systems that approach other design domains is defined.

## 1 Introduction

Nowadays, the CSCW and CSCL paradigms attract the attention of many researchers. However, most of developed collaborative systems approach specific applications and are not flexible nor adaptable to different situations and purposes, especially in the area of the CSCL and of the systems for real time collaboration. One of the directions for future research in the field of CSCW –and of the CSCL– should be to build generic models for collaborative work, by means of architectures or formal languages, allowing us to describe properties and conditions of the systems [1], and to develop tools that can generate software systems starting from these descriptions.

We are interested particularly in CSCL systems for the learning of design tasks. These systems follow instructional methods as Problem Based Learning (PBL) or Project Based Learning, as these methods reinforce the commitment among the learner and the activity he/she carries out. Particular types of problems are the ones concerned with design. Most of them are complex, requiring the designer to use domain knowledge to build an artefact or to carry out a process under certain conditions. The cognitive strategies of problem solving are domain specific [2], and it is necessary to model the domain in order to model the problem solving.

In this work we deal with the modelling of domains and generic problems of design in the context of a collaborative learning environment. Thus, the DomoSim-TPC environment will be described first, which is taken as a reference to generalize its approach.

## 2 The DomoSim-TPC System

We have developed DomoSim-TPC [3, 4, 5], a collaborative system for the distance learning of domotical design (Home Automation). It includes support for setup, realization, monitoring, analysis, and storage of learning activities. This system has been evaluated following diverse techniques and it is being successfully used in different teaching centers. The DomoSim-TPC system has been developed in Java to allow the execution with a web browser, so that the system can be used at a distance.

We represent the Domotics domain by means of a set of object types and of relationships among them. We call operators to these domain-specific objects. The types of operators we identify are regulation systems, activators, receivers, and other elements (plugs, buses...). The systems are related with the activators and the receivers they regulate. Another type of relationships among operators are, for example, the connections to plugs or data buses. The operators are classified in five management areas: thermal comfort, luminosity comfort, energetic control, accident security, and intrusion security.

The process of problem solving is structured in two stages:

- Planning of the Design Strategy [5]: The learners decide the steps that determine the strategy to follow to build the model they consider satisfy the problem objectives, first individually and then in group.
- Detailed Design and Simulation in group [4]: The learners build a model and simulate it by means of successive refinements, checking if this model solves the problem.

In this work we focus on the second stage, a synchronous task, which facilitates the generation of spontaneous ideas, brings in personal experience, increases motivation, and encourages a content focus.

## 3 Activities Organization and Collaborative Problem Solving

The users of the DomoSim-TPC system are classified in two profiles: teacher and student. The students are arranged in groups coordinated by one or more teachers. The access to the tools is regulated according to the user's profile. The teachers use management tools, for the definition of the activities and participants, and authoring tools, to create the problems.

The students work actively in groups on problems extracted from the real practice (PBL). Each problem describes the requirements of the model that the students have to build. In this situation the student plays the role of a professional designer. This collection of problems is organized in three levels of complexity: low, medium, and high. Thus, the teacher can propose problems with a certain complexity level according to the knowledge level of the group. The *activity* abstraction allows proposing a concrete problem to a concrete group of students. From the organizational point of view, a problem is a learning object that can be reused. The activity help level, together with the complexity level of the problem, allows teachers to use scaffolding techniques [6]. This way, the problem solving can be considered with

different levels of help (reinforcement), depending on the previous knowledge of the participants and in a process of successive integration of knowledge.

The process of detailed solving of problems is framed in the Design and Simulation subsystem. This subsystem is materialized by means of the shared workspace metaphor. There are five workspaces available: Design, Task Distribution, Parameterization, Cases and Hypothesis, and Simulation. The navigation along these workspaces is defined by means of a Collaboration Protocol that synchronizes the joint access of the students belonging to the group. The Design workspace is the first one accessed when the group starts the resolution of a problem. From this, the Work Distribution, Parameterization, and Cases and Hypothesis workspaces can be accessed; from here it is possible to come back to the Design. The Simulation workspace is accessed from the Cases and Hypothesis workspace. All these subspaces integrate different mechanisms of direct manipulation, support for communication and coordination, and awareness techniques [7].

### 4 Generalization to Generic Design Domains

To carry out this generalization we propose the use of XML to build a language that allows specifying the characteristics of the design domain (DDL, Domain Definition Language) as well as of the problems (PDL, Problem Definition Language). Other authors also use specifications in XML to represent interactions in CSCL systems [8], to describe learning activities (users, tasks, tools...) [9], and to implement an Educational Modelling Language, which allows users to define Units of Study [10]. The choice of the XML notation is because of its feature of standard language and its interoperability. We aim that this specification language of domains and problems has certain degree of formalization, that can be processed automatically and be compatible with standards, allowing the reusability of the learning objects (problems, plans...). For example, XML would allow, by means of XSLT transformations, the generation of XHTML pages with a description of the domain or the problem visualizable on a browser.

The domain and problem models are defined by means of a DTD (Document Type Definition) that describes the valid syntax of the XML documents. So, four files make up the specification: DTD files for the domain and the problem, and the domain itself and problem definition by means of XML (see table 1).

**Table 1.** A schematic example of specification in DDL and PDL.

Domain specification	Problem specification
<pre> &lt;domain name="dom1" ...&gt; &lt;graphics&gt;...&lt;/graphics&gt; &lt;operators&gt;...&lt;/operators&gt; &lt;relationships&gt;...&lt;/relationships&gt; &lt;variables&gt;...&lt;/variables&gt; &lt;/domain&gt;                     </pre>	<pre> &lt;problem id="p1" domain="dom1" ...&gt; &lt;description&gt;...&lt;/description&gt; &lt;comments&gt;...&lt;/comments&gt; &lt;behavior&gt;...&lt;/behavior&gt; &lt;variables&gt;...&lt;/variables&gt; &lt;requirements&gt;...&lt;/requirements&gt; &lt;constraints&gt;...&lt;/constraints&gt; &lt;cases&gt;...&lt;/cases&gt; &lt;hypotheses&gt;...&lt;/hypotheses&gt; &lt;/problem&gt;                     </pre>

## 5 Summary and Future Work

We propose the use of XML for the modelling and specification of the domain and of the problems in collaborative design systems. This way, we have built a Domain Definition Language (DDL) and a Problem Definition Language (PDL) on this domain. These languages would also facilitate the automatic generation of problems as well as the elaboration and verification of solutions for them.

The following objective is to build a tool that automatically processes these languages and can generate collaborative systems to approach the learning of design in different domains.

**Acknowledgements.** This work was partially supported by the Junta de Comunidades de Castilla – La Mancha and the Ministerio de Ciencia y Tecnología in the Projects PBI-02-026 and TIC2002-01387.

## References

1. Chang, C.K., Zhang, J. & Jiang, T.M.: *Formalization of Computer Supported Cooperative Work Applications*. Proceedings of the Eighth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'01) (2001)3.
2. Mayer, R.E.: *Thinking, problem solving, cognition* (2<sup>nd</sup> ed.). New York: Freeman (1992)
3. DomoSim-TPC Tutorial: <http://chico.inf-cr.uclm.es/domosim>
4. Bravo, C.: *Un Sistema de Soporte al Aprendizaje Colaborativo del Diseño Domótico Mediante Herramientas de Modelado y Simulación*. Doctoral Thesis. Computer Science Department, University of Castilla - La Mancha (Spain) (2002)
5. Redondo, M.: *Planificación Colaborativa del Diseño en Entornos de Simulación para el Aprendizaje a Distancia*. Doctoral Thesis. Computer Science Department. University of Castilla - La Mancha (Spain) (2002)
6. Rosson, M.B. & Carroll, J.M.: *Scaffolded Examples for Learning Object-Oriented Design*. Communications of ACM, vol. 39, no. 4 (1996)
7. Bravo, C., Redondo, M.A., Ortega, M. & Verdejo, M.F.: *Collaborative Discovery Learning of Model Design*. Cerri, S.A., Gourdères, G. & Paraguaçu, F. (Eds.) Intelligent Tutoring Systems, 6<sup>th</sup> International Conference, pp. 671–680, LNCS, Springer (2002)
8. Martínez, A., Dimitriadis, Y. & de la Fuente, P.: *Aportaciones al análisis de interacciones para la evaluación formativa en CSCL*. Llamas, M., Fernández, M.J., Anido, L.E. (Eds) Proceedings of 4<sup>th</sup> International Symposium of Computers and Education, p. 41 (2002)
9. Verdejo, M.F., Barros, B., Read, T. & Rodríguez-Artacho, M.: *A System for the Specification and Development of an Environment for Distributed CSCL Scenarios*. Cerri, S.A., Gourdères, G. & Paraguaçu, F. (Eds.) Intelligent Tutoring Systems, 6<sup>th</sup> International Conference, pp. 139–148, LNCS, Springer (2002)
10. Koper, R.: *Modeling units of study from a pedagogical perspective. The pedagogical meta-model behind EML*. Educational Technology Expertise Centre. Open University of the Netherlands. Draft. <http://eml.ou.nl/introduction/docs/ped-metamodel.pdf> (2001)