Collaborative environments for the learning of design: a model and a case study in Domotics

Crescencio Bravo a,*, Miguel. Á. Redondo a, Manuel Ortega a, M. Felisa Verdejo b

a Dpto. de Informática, ESI, Universidad de Castilla-La Mancha, Paseo de la Universidad, 4, 13071 Ciudad Real, Spain
b Dpto. de Lenguajes y Sistemas Informáticos, ETSII, UNED, Ciudad Universitaria, s/n, 28040 Madrid, Spain

Received 1 February 2004; accepted 12 July 2004

Abstract

Design plays a central role in a range of subjects at different educational levels. Students have to acquire the knowledge necessary for the execution of tasks that enable them to construct an artefact or model that can be tested by simulation and that satisfies some requirements and verifies some constraints. They achieve this by means of a design process. In some design domains there is a lack of teaching tools from a learner-centred perspective. Moreover, when these domains are complex, the design problems that the students have to solve during their learning process require the design activity to be carried out in group. In response to this situation, we have developed a design model and a collaborative learning method. Using this conceptual framework, we have built a collaborative environment for the learning of domotical design by means of complex problem solving, with an emphasis on synchronous collaboration for work distribution, discussion, design in shared surfaces and simulation. This environment has already been evaluated and used in real teaching experiences.

© 2004 Elsevier Ltd. All rights reserved.

This work has been supported in part by the Ministerio de Educación y Cultura (Spain) in the TIC2002-01387 project and by the Comunidad Autónoma de Castilla-La Mancha (Spain) in the PBI-02-026 project.

* Corresponding author. Tel.: +34 926 295300x3717; fax: +34 926 295354.

E-mail addresses: crescencio.bravo@uclm.es (C. Bravo), miguel.redondo@uclm.es (M.A. Redondo), manuel.ortega@uclm.es (M. Ortega), felisa@lsi.uned.es (M.F. Verdejo).

Domotics: It is the technical discipline that studies Housing Automation. It is also called Intelligent Building Design.
1. Introduction

Design is a very general term that refers to many kinds of tasks. At an abstract level, Klein, Sayama, Faratin, and Ban-Yam (2002) consider that a design can be represented as a set of issues (or parameters) each with a unique value. In this study, we understand design as a creative task consisting of building a model or artefact. To carry out this task, a design computer environment provides the computational workspace in which the designers carry out their work.

When the design activities are complex, it is difficult for an individual user to be familiar with all the significant information in a domain and to solve a problem. These situations of complex design are characterized by a Symmetry of Ignorance (Rittel, 1984), and the necessary knowledge to solve a design problem is distributed among all the designers. Therefore, collaboration is required to approach complex design tasks. According to Maamar and Shen (2002), a collaborative design environment enables team members to share information and to coordinate their activities within the context of a design project.

We are particularly interested in the teaching of design, which is not an easy task. The aim is for students to acquire some knowledge, which enables them to execute the tasks necessary for the construction of an artefact or model satisfying some requirements and verifying some constraints. The use of Simulation can improve the cognitive processes since it can be applied to the models to check their operation and the satisfaction of the requirements and restrictions, thus stimulating learning, reinforcing knowledge and promoting discovery. According to Van Joolingen (2000), the main goal of a discovery learning activity is to obtain and construct knowledge about a domain by performing experiments and inferring rules and properties of the domain from the results of those experiments.

In line with the above arguments, we focus on describing how a design task can lead to the learning of a domain. To do so, we follow the conceptual framework that provides the CSCL (Computer-Supported Collaborative Learning) paradigm (Koschmann, 1996). The CSCL is based on the model of instruction of Collaborative Learning and on socially oriented learning theories. Collaborative Learning allows the students to approach more complex problems, and to express designs, critics and arguments to partners, which promotes a reflection type that leads to learning. In collaborative environments of design learning, the design process is seen as a task in which the individuals share a common objective (to design) in which they need to work in group (to collaborate). It is the process the students follow that leads them to learn.

Our general objective is to model a design process consisting of the construction of an artefact and to develop a method or methodology of collaborative learning of design. With that purpose in mind, we follow two methods compatible with the CSCL: Problem-Based Learning (PBL) and Learning By Design (LBD). PBL is a case-centred and learner-directed method of instruction (Koschmann, Kelson, Feltovich, & Barrows, 1996) that is used in collaborative environments. In its ideal implementation, a small group of students, together with a coach or tutor, learn by a process of working through a collection of teaching cases or real life problems. The LBD
(Kolodner, Crismond, Gray, Holbrook, & Puntambekar, 1998) is an approach for the learning of Sciences in which the students learn as a result of developing collaborative activities of design and reflecting on their experiences. The concepts are assimilated by experimenting with their operation and are applied to the resolution of real life problems. The students learn to make decisions and acquire collaboration skills as a consequence of being involved in activities of this type.

Using this conceptual framework, we have developed DomoSim-TPC, a CSCL environment for the learning of domotical design. Domotics is an interesting design domain that studies the integral automation of housing and buildings. This discipline is taught as a subject in Secondary Education (Technical Training) and in Technical University Colleges of Industrial Engineering in Spain. Domotics can be defined as the set of elements that, when installed, interconnected and automatically controlled at home, release the user from the routine of intervening in everyday actions and, at the same time, provide optimised control of comfort, energy consumption, security and communications. Nowadays, there is a growing need for tools to be used in practical classes in Secondary Education to complement the theoretical contents of this subject.

Many environments have been developed for design learning, e.g., C-CHENE (Baker & Lund, 1996) for the construction of energy chains, COLER (Constantino-González & Suthers, 2001) for database modelling and EPSILON (Soller & Lesgold, 2000) for object-oriented design. In general, these systems do not structure the design task in a suitable way, they do not take advantage of the simulation possibilities to test the models that are built nor do they offer an appropriate awareness support. These restrictions can limit the students’ learning. The main contribution of our approach is a PBL method in which a collaborative simulation task is used to test the built solutions. The design and simulation tasks incorporates interaction structuring techniques and a wide support of awareness, communication and coordination.

To develop the aforementioned objective, the rest of the paper is structured as follows: the design and learning methods are described in Section 2. In Section 3 we describe the environment we have developed as a case study using these models. The development and evaluation process of this environment are shown in Section 4. Finally, some conclusions are drawn and future lines of research are outlined.

2. A method of design learning based on structuring

In this modelling task we assume the design characteristics from the LBD perspective (Edu-Tech, 2004): the design is realistic, interactive, iterative, reflective and collaborative; and design uses artefacts, needs representations with suitable tools, is different from inquiry, and gives feedback to the designers.

2.1. The method of design

To characterize the design method, it is necessary to define a domain model and a process model. We represent a domain by means of an object model. We call a domain object with a specific behaviour an operator. The operators have properties called parameters and they can be organized in categories. The domain itself is represented by means of an object with a series of general parameters. The operators are related amongst themselves following a metaphor of interaction
among objects. There is a set of relationship types among the operators. The general parameters can determine if some relationships are possible or not. The artefact to be built collaboratively by the designers is an instance of this object model. When designers work with computer environments, they usually use graphic interfaces in which the objects are represented by means of icons and the relationships are denoted by means of lines. Some of these design environments, such as SCOPE (Miao, Pfister, Wessner, & Haake, 1999), SEPIA (Streitz et al., 1992) and AEN (Johnson & Moore, 1995), are based on hypermedia structures instead of object models that have a bigger expressive capability; others, such as DDA (Kolodner & Nagel, 1999), only manage texts (forms, documents and messages), which makes them unsuitable for interactive design processes.

To model the design process, we identify a set of phases or stages that give expression to the mentioned characteristics of design. We identify a first phase of planning and a second phase of design. The planning allows an abstract plan of design agreed on by all the group members to be built. This will be the starting point to build the definitive artefact. In this phase, the aim is to encourage reflection and discussion in the construction of a scheme of the artefact to design so that the requirements and design restrictions are satisfied. Tools based on asynchronous communication to specify, discuss and organize the general positions of the design can be used. The advantages of planning as a previous phase to design and simulation have been pointed out by Bravo, Ortega, and Verdejo (2000) in previous research carried out by our group. The Collaborative Planning of Design according to this model is described in depth by Redondo (2002).

In this study we focus on the second phase. This is more interactive, artefacts and visual representations are managed, and a more realistic type of design in which simulation plays an important role for the iterative improvement of the designed artefacts is reflected. Since it is not such a reflective task, we propose synchronous collaboration as a better way of carrying it out. This alternative also facilitates the appearance of spontaneous ideas and promotes personal experience (Newman, Johnson, Webb, & Cochrane, 1997). Due to its complexity, this phase is structured in four tasks: (i) the design itself consisting of the collaborative construction of the artefact or model; (ii) the work distribution and the coordination of the designers; (iii) the decision-making relative to the values of the general parameters; (iv) the simulation of the designed model.

This method causes difficulties to be implemented by means of a computational environment. Tools are required that allow the representation of the artefacts to be manipulated, the navigation of the group through the tasks in a dynamic way, and the carrying out of the collaborative work in an efficient way that limits the conflicts of intentions.

2.2. The method of design learning

In the learning of a task requiring design by means of a PBL method, the artefact or model that is built is the solution to a problem previously outlined. Moreover, we incorporate a simulation task in PBL in order to test the designed models. The learning situation can be outlined in the following way:

- There is a collection of design problems.
- The students are organized in groups of 2–5 people.
- The teacher selects problems from the collection and proposes them to the groups of students.
The students solve the problems in groups, collaborating both synchronously and asynchronously. Individual work is also allowed.

The students solve the problems with the help of specialized tools:
1. They organize and distribute their work, drawing up a resolution strategy that divides the problem into sub-problems.
2. They design a model or artefact:
   • They set general conditions (for example, to give value to the general variables of the model).
   • They identify subsystems to solve the sub-problems.
   • They define the subsystems inserting operators on the model, linking them and giving value to their properties.
3. They simulate the model in order to validate and refine the design.
4. They communicate, exchanging information in relation to the domain, coordinating their actions and making decisions by coming to agreements.

This way of learning by design fits in with the previous design model. To organize and structure this process in a collaborative learning environment, we follow the characteristic support methods that are distinguished in CSCL (Hron, Hesse, Reinhard, & Picard, 1997; Mancini, Hall, Hall, & Stewart, 1998): global methods that structure collaboration at a general level and guided methods that structure dialogue and learners’ actions. In our proposal, the global structuring of the former methods is achieved with workspaces and subsystems including the available tools. The latter methods refer to collaborative interaction structuring. We propose a semi-structured model for the synchronous collaborative resolution of design problems that is based on Scripting to materialize collaboration protocols (Wessner, Hans-Rüdiger, & Miao, 1999) that structure the process, on Language/Action Perspective (Winograd, 1988), based on speech acts, to express and categorize actions by means of a repertory of dialogue primitives, and on Flexible Structuring (Lund, Baker, & Baron, 1996) to build effective tools of communication and coordination. A system that follows this structuring approach is SCOPE, which combines facilities for structuring tasks, activities and processes with complete tools for collaborative work on design surfaces. This is a very flexible system because it allows the work structures to be defined, but it uses design models based on hypermedia structures that limit its field of application and does not structure the communication.

In Fig. 1, the general architecture for our design learning method is shown. There are two kinds of participants with a different role in the use of the system: teacher and student. Each one has different tools for the execution of the tasks. These tools are organized in four shared workspaces.

The teacher configures the learning experiences, organizes the participants and defines activities in the Activity Management workspace. In order to define a generic design problem it is necessary for the teacher to specify the following information: (i) an identification; (ii) a formulation; (iii) a scenario or environment – expressed as a background or plan – on which to make the design and its features; (iv) a set of constraints that express design restrictions; (v) a set of requirements or design necessities; (vi) a behaviour model of the scenario or environment; (vii) some cases and hypotheses to be used in the simulations; (viii) the complexity level. An activity is an abstraction that represents the proposal of a specific problem for its resolution to a group of students with a help level that can be high, medium or low. This level offers scaffolding (Rosson & Carroll, 1996),
determining the quantity of help the environment offers. The students solve problems starting from a low and moving up to a high complexity, and the higher the complexity is, the less help is offered. Thus, the students reinforce the intrinsic structure of the problem resolution process. In this way, knowledge is gained by means of a leverage process of re-elaboration and integration. Apart from the activities, the teacher must define the work sessions. We understand a session as the period of time in which the group can carry out design activities in the shared information space.

Teachers and students have tools to organize themselves, to communicate with each other and to schedule their work before and after the sessions: electronic mail, chat and agenda of sessions. These tools are available in the Communication and Coordination workspace.

The students, in groups, solve problems in the Design and Simulation workspace during the sessions configured by the teacher. We describe the general collaboration protocol we propose for problem solving by means of a state diagram (Fig. 1, Design and Simulation workspace) in which each node represents a task and the set of tools that can be used to carry it out. We can call each node a work subspace. The protocol consists of four subspaces: Design, Work Distribution, Parameterisation and Simulation. The subspace which is first accessed is Design, from which the students move to other subspaces in order to build the solution to the problem in a refinement
process, which shows the iterative character of design. This collaboration protocol has been contrasted according to the design experience of domain experts and teachers, and supports our design learning method and the four tasks of design identified.

Collaborative interactions for problem solving include two interdependent cognitive tasks: solving the problem (design) and collaborating. The problem solving is carried out by means of the domain tasks in the Design and Simulation workspace. The interaction style of direct manipulation will be used for carrying out these tasks, such as building the model, interacting with simulation, defining variables, etc. However a complete collaborative support has to be available in these tasks. This support is composed of the following mechanisms:

- Communication support: collaboration requires communication to exchange information, to coordinate actions and to reach agreements. This support consists of functions allowing the users’ written communication during the design tasks. It reflects the Flexible Structuring approach and will be implemented by means of a Guided Chat.
- Coordination support: the communication allows the users to be coordinated in a general way by using the language. However the Work Distribution and Parameterisation tasks require a structured coordination consisting of materializing proposition and agreement processes by means of actions, according to the Language/Action Perspective. Moreover, we identify the necessity of a generic decision-making tool that allows users to resolve, democratically, the conflicts among them when carrying out an action.
- Awareness support: we have given great importance to awareness, which can be understood as the perception and knowledge of the interaction that other people carry out in a shared space (Gutwin & Greenberg, 1997).

All this collaboration support is ready to be used during the design tasks. On the contrary, C-CHENE differentiates between construction (design) and communication, and the users have to indicate explicitly if they want to build or to communicate, with it not being possible to do both things simultaneously.

In the Monitoring and Analysis workspace the teacher can analyse the process that has been followed and the solution designed using semiautomatic tools; for this the students' interactions and the solution built are registered in the Design and Simulation workspace. The analysis method is presented in depth in Bravo (2002). The teacher, besides being the organizer of learning experiences and the students' work evaluator, can be a partner to them during their tasks, a tutor and a facilitator, so that he/she can create specific learning situations, promote discussions, etc.

3. A collaborative environment for the learning of domotical design

In this section, we describe how we have applied the presented method of design learning to approach the teaching and learning of Domotics. We have developed DomoSim-TPC (Bravo, 2002; Bravo, Redondo, Ortega, & Verdejo, 2002; Redondo, Bravo, Ortega, & Verdejo, 2002) as a CSCL system for the learning of domotical design that incorporates tools for organization, authoring, model building and simulation, communication and coordination, and analysis.
The Domotics domain is clearly a design domain. The design to be carried out consists of the construction of a domotical installation in a house or building. We have modelled this domain by means of: (i) a set of types of operators (receivers, activators and systems) organized in management areas: comfort (luminosity and thermal comfort), energy control and security control (accidents and intrusion); (ii) a set of types of relationships among them (control relationship, electrical connection...); (iii) a list of parameters which describe the scenario and the environmental characteristics; (iv) the definition of a discrete event simulation model for the designed installation. Therefore, DomoSim-TPC is domain-oriented: it incorporates explicit domain knowledge.

In DomoSim-TPC, the problems are created by teachers and experts in the domain, assuring the maximum similarity between the problem representation and the reality. They use authoring tools. The elements that define a generic design problem are instantiated in specific components of the Domotics domain: the scenario consists of the characteristics of a house described by a plan formed by rooms; the requirements and constraints are expressed by means of conditions on the parameters or with regard to the inclusion of certain operators in the solution; and the behaviour model is characterized by means of a list of assignments of value to the parameters.

Next, we describe DomoSim-TPC focusing on the Design and Simulation workspace. A description of the rest of the workspaces can be read in Bravo (2002). To illustrate this description, we will show an example of a solution of a problem that consists of automating the thermal comfort and of installing some electrical appliances in two rooms (kitchen and lounge) by two students who collaborated in a remote way (cbravo and mredondo). The specific requirements are to reach a temperature between 18 and 22 °C in the house and to install the domotical system of Power-Line Carrier. There are constraints in the maximum number of radiators (three) and of air conditioners (two) to be used, and a maximum limitation in the electric power (3000 W).

3.1. Design

In this subspace the students design the model that constitutes the solution to the problem. To carry out this task, the subspace includes four tools according to the developed design learning model: a domain-specific design tool, a graphic annotation tool, an awareness tool and a discussion tool. The first one is a specialized domotical design tool. The others are general domain-independent tools.

The domain design tool is a direct manipulation tool based on the object-action model that follows the collaborative electronic whiteboard metaphor. The actions the students carry out on the design surface are edition actions (insertion, deletion and movement of domotical operators and of links), links between operators and their parameterisation. Typically, the students build domotical subsystems inserting the operators of the management area involved, linking the control system operator with the receivers and activators, assigning values to the properties of these operators and linking them to other elements (buses, plugs...). A less adequate and interactive alternative is that followed by the C-CHENE system, which uses option menus for the construction (design) task.

With the graphic annotation tool the students can draw different figures (circles, lines, arrows, crossings...) and insert text in the design surface. These annotations can be generalized for any
domain and have commonly accepted semantics: circles, lines and rectangles allow the marking of areas; arrows are used to point to objects or areas; and crosses allow the crossing out of objects. This helps students to express ideas in relation to the model.

The awareness tool keeps a set of tele-pointers, an interactions list and a panel with the session members containing the users’ pictures, their names and their state. The states we have identified for each user are: editing, parameterising, selecting, linking, simulating, designing, drawing and communicating. The name and state are shown in the same colour, which is unique for each user and corresponds to that of the tele-pointer. In the central part of Fig. 2 we can see the tele-pointer of the student collaborating with the student to whom the interface shown corresponds. Systems such as SCOPE, SEPIA and DOLPHIN (Mark, Haake, & Streitz, 1995) use similar tele-pointers, but they do not represent the user’s state. All the interactions are distributed immediately to the group members to be shown in the shared space. The system informs of these interactions in the interactions list and may beep optionally to capture the user’s attention. All this allows users to know, for example, what the other students are doing, where they are, what they are likely to do next (for example, the selection of an object suggests that it is going to be deleted, parameterised or moved next), etc. The Session Panel and the interactions list are available in the four work subspaces.

The discussion tool materializes the communication and coordination support. It consists of a Guided Chat and a Decision-Making tool which are described in the support tools section. These elements are also available in all the subspaces.

---

Fig. 2. Design subspace.
In Fig. 2, the subspace user interface with the tools that it integrates is shown. On the left side of the window the domotical operator toolbars can be seen, and on the right is the drawing toolbar. The Session Panel, the Guided Chat, the interactions list and the error message area are available in this subspace. The work surface contains a house plan on which a set of operators has been inserted. The data model that represents the solution that is being built is persistent. It is stored and is available for both the teacher to consult as well as to be worked on in subsequent sessions.

The model that the students have built in relation to the example given (Fig. 2) contains, in the lounge (left room), two radiators, one air conditioner, one temperature sensor and the regulator system, as well as some appliances (computer, television and stereo). In the kitchen (right room), the students have inserted another thermal comfort subsystem and the appliances that the problem demands (microwave, oven, cooker, refrigerator and washing machine). The operators are connected to the plugs to obtain electricity and to facilitate the regulation control, according to the domotical system of Power-Line Carrier. With a different colour link to the electrical connection, the receivers and activators are linked to the systems that regulate them. The students have defined parameters of each operator such as the energy consumption, the calories of the radiators (100 cal), the frigories of the air conditioners (100 fr), etc. This is a possible solution, but there are several valid ones. While building a solution, the students reflect on the satisfaction of the requirements and the verification of the constraints.

In the Design and Parameterisation tasks a model that complies to the problem constraints must be built. The system performs controls on the interactions, informing with help messages when the constraints are not verified. In this case, the students can reflect on the reasons that have caused the error, thus drawing their own conclusions and working to solve it. The quantity of help automatically offered depends on the help level that has been predefined for the activity, according to the scaffolding level specified in the problem.

3.2. Work distribution

Previously to the design stage, the students must follow a strategy of division of the problem into sub-problems, each one entailing the design of a subsystem. According to this, the functionality of this subspace is to give support to a distribution of the actions that the group members will carry out in the design, in such a way that the work is coordinated. The detailed protocol of Fig. 3 shows the two stages of the distribution: the choice of the distribution criterion and, according to this criterion, the assignment of responsibilities to the participants. The criterion can be changed at any time.

![Choice of Distribution Criterion](image1)
![Assignment of Actions](image2)

Fig. 3. Protocol for work distribution.
In the case of DomoSim-TPC the distribution criteria are:

(a) By plan areas: for example, rooms in a house.
(b) By domotical management areas: working with one or several areas is allowed.
(c) By action types: editing, linking and parameterisation of operators.
(d) No criterion: there is no task assignment, so free work is allowed; the participants must coordinate without support from the system.

These criteria can be easily applied to other domains. They allow collaboration to be structured according to (i) roles (each student plays a role that consists of carrying out certain actions; for example one student is the main designer—he/she takes charge of inserting operators— and another is the linker—he/she takes charge of connecting elements—), (ii) specialists (each student is assigned those actions or management areas in which he/she is specialist; so that the specialist’s advanced knowledge contributes to the group knowledge and learning according to the Symmetry of Ignorance) or (iii) responsibilities (each student takes responsibility for a functional unit: management area or room).

The assignments which the group has agreed upon relate each student to one or more actions. This distribution strategy makes each student assume his/her responsibility in the part of the design problem assigned to him/her, having achieved a social organization of the work (in criteria a, b and c). It is not possible to leave actions without assignment and one action type will be carried out by only one student. A student will also be unable to carry out actions assigned to another student or access another work area. Thus, with the appropriate assignments, the distribution...
would allow only one student (or teacher) to carry out all the design actions, becoming the tutor of the rest of the students in the group. The COLER system allows a work distribution similar to this last one to be defined, so that only the student who has the pencil can update the shared workspace. This is defined by means of two operations: ask/take pencil and leave pencil.

In Fig. 4 the window corresponding to this subspace is shown, which includes a work distribution tool, an awareness tool (Session Panel and interactions list) and a discussion tool (Guided Chat and Decision-Making tool). The work distribution tool is the coordination support necessary to organize and distribute the users’ work. This tool is domain-independent, except for in the case of the domotical management area criterion. On the left side (1, 2) are the user’s proposals. The partners’ proposals and the buttons to express agreement or disagreement can be seen in the central part (3, 4). The right part (5, 6) contains the decisions already defined, which are the criterion and the assignments that will be used in the design.

The criterion as well as the assignments are chosen by means of a process of proposals (Fig. 5) according to the Language/Action Perspective. A student makes an initial proposition and the rest of the group members must respond to it with an agreement (OK) or disagreement (Not OK). When all the participants agree, the proposal is carried out.

In Fig. 4, the students have carried out the work distribution they consider appropriate for the problem outlined (see Fig. 2). A student has proposed a distribution by management areas (1) and his/her partner has agreed (3), being assigned this criterion (5). Next, they have proposed that each of them should take charge of an area (2); this can be seen in the proposals and agreements list (4). The final list of assignments is shown in the corresponding part (6): cbraavo takes charge of the thermal regulation and mredondo takes charge of the energy control. Later on, they should integrate their work and evaluate the combined model checking to see if the parts are compatible. After simulating, and if necessary redesigning the model, the students can define a new work distribution.

3.3. Parameterisation

This subspace allows the group to give a value to the general variables of the problem that form part of the solution. This is carried out by means of a process of proposals according to different decision models. In Fig. 6, the window corresponding to this subspace is shown. It integrates a parameterisation tool, which contains the parameters to be defined, an awareness tool and a discussion tool. The parameter list is on the left and their values (initial and final) on the right. The initial value is the one that the variables have when the subspace is accessed, and the final value is the new value assigned by the students. The proposals of the group members appear in the column corresponding to their picture and in the row corresponding to the parameter (2).
In the column corresponding to the user there is a button to make proposals (1). A parameter can be numerical, logical, textual or a value from a list. The students will only be able to modify those parameters that have been marked as modifiable in the definition of the problem.

There are two possible dialogue models: Based on Proposals and Democratic. In the first one (see Fig. 6), each student should show his/her agreement or disagreement with the proposals of the others, and the final decision is the value that has reached consensus, that is to say, the value which all the students have agreed to. As in the Work Distribution, this process is represented by the graph in Fig. 5. In the second model, the students propose different values, and the final decision is the value given by majority (arithmetic mean in the case of the numeric values and mode in the rest). In this case the interface does not include the Ok and Not Ok buttons.

In the example problem the students have defined the power of the load lines and have assigned the plugs to these lines. The student cbravo has proposed a value of 1500 W for the load lines 1, 2 and 3 (1), and the student mredondo has agreed. As a result, this value is assigned as the final value. The student mredondo has proposed assigning two plugs to line 1 and two to line 2; to express his agreement, cbravo presses the Ok buttons corresponding to these proposals (2). With this they aim to give the necessary electrical supply to the operators but without exceeding the total electric power of the problem.

3.4. Simulation

This subspace allows the domotical solutions that the students have built to be tested by means of a simulation tool. In order to do this, it is necessary to define and select a case set (instantiated
variables) allowing users to study how the model behaves and to state hypotheses about the variable values. For this a two stage protocol is outlined (Fig. 7). In the first stage, the task of proposing and selecting simulation cases and managing simulation hypotheses is carried out (Bravo et al., 2002). Once a case has been selected, the second stage is carried out and simulation starts.

The subspace corresponding to the simulation stage integrates three tools: the simulation tool, which is based on direct manipulation on an electronic whiteboard as is the design tool, the awareness tool and the discussion tool. The window corresponding to the subspace (Fig. 8) contains the different elements of the simulation tool: the whiteboard with the designed model (1), buttons to carry out simulation actions (2), a panel to propose the finishing of the simulation (3), a panel of operator properties (4) and simulation information (temperature, illumination, consumptions and clock) (5). It also contains the awareness and discussion elements: the Session Panel, the interactions list, the Guided Chat and the Decision-Making tool.

The simulation actions, based on the object-action model, are used to modify the automatic behaviour of the designed solution. The main objects are operators and links, and the actions are to switch on/off, to open/close, to break a link, to simulate human presence in a room…
The solution is visually represented in a realistic and interactive way. The changes in this behaviour, automatic or caused, are reflected in real time on the whiteboards of all the participants in a graphic way and in the interactions lists in a textual way. In so doing, the experimentation space is available to all the participants in the same view, something essential in the opinion of Van Joolingen (2000). Therefore, in this Simulation stage, as in the Design stage, the actions to be developed by the users are not structured and free interaction is allowed.

The teachers can pause the simulation (with the Pause button), for example, to propose a question, cause a reflection, etc., and to continue with the simulation afterwards. This function, together with the possibility of the teacher taking part in the simulation with any action, offers interesting ways of mediation in the students’ learning.

Fig. 8 shows a simulation session of the designed model in the example problem (see Fig. 2). In the environment panel the students can see that the rooms are not reaching the desired temperature (18–22 °C) at any time during the day (in the sample figure, 17.89 in the lounge and 17.12 in the kitchen) and they should reflect on the best way to correct this malfunction of the model: either by increasing the number of radiators or by modifying their heating power. Also, the students have activated all the appliances to bring about maximum consumption and to check if the electric power that has been contracted has been exceeded. It can be seen that line 3 has a consumption of 1100 W, therefore not exceeding the maximum of 3000. The activated radiators (lounge) show a different icon from the deactivated radiators (kitchen).

In this process the students will check if the model behaviour is what was expected, that is to say, they will check to see if the problem requirements are reached: they will look at the value that the variables have (the general and operator-specific ones), detect breakdowns or accidents, and interact on the model to provoke reactions. If the requirements are not reached the solution is not correct, and the students must go on collaborating to modify the model in a process of successive refinements (design-simulation) until the requirements are reached, as LBD proposes.

A collaborative design system allowing simulation is WebNet (Stahl, 2000), but the collaboration is limited to discussion about the models, and the design and simulation tasks are individual.

3.5. Support tools for dialogue and coordination

In a similar way to C-CHENE, we decided to incorporate a chat linked to the design and simulation tools (see Figs. 2, 4, 6 and 8). This chat is defined as Guided because it offers a pre-established set of goal-oriented communicative acts with a semantic meaning that, in some cases, have to be completed with text by the user. In this way, a structure for the conversation is provided as an understandable form of expression. For example, in the simulation, while the students observe and discover by themselves, they can share their ideas, arguments and conclusions. The messages, selected by teachers, students and domain experts, are classified according to their type and place in the dialogue (Table 1). The tool activates and deactivates the available buttons for communication according to the conversation state in order to focus the dialogue. The C-CHENE chat interface does not allow this possibility.

In this Guided Chat there is a button that allows users to invoke a generic Decision-Making tool, also known as Voting tool (Fig. 9). This tool allows the proposal of a question to the group to reach a decision about some topic. There are three kinds of questions: (1) those with an affirmative or negative answer; (2) those with a real value as an answer; (3) those with an op-
Table 1
Message types of the Guided Chat

<table>
<thead>
<tr>
<th>Message</th>
<th>Type of message</th>
<th>Place in the dialogue</th>
</tr>
</thead>
<tbody>
<tr>
<td>I think that...</td>
<td>Assertion</td>
<td>Initial</td>
</tr>
<tr>
<td>I think so</td>
<td>Assertion</td>
<td>Reactive</td>
</tr>
<tr>
<td>I don’t think so</td>
<td>Assertion</td>
<td>Reactive</td>
</tr>
<tr>
<td>Why...?</td>
<td>Question</td>
<td>Initial</td>
</tr>
<tr>
<td>I don’t know</td>
<td>Answer</td>
<td>Reactive</td>
</tr>
<tr>
<td>Because...</td>
<td>Answer</td>
<td>Reactive</td>
</tr>
<tr>
<td>We’re doing well</td>
<td>Assertion</td>
<td>Initial</td>
</tr>
<tr>
<td>I see a mistake...</td>
<td>Assertion</td>
<td>Initial</td>
</tr>
<tr>
<td>I’ve finished</td>
<td>Assertion</td>
<td>Initial</td>
</tr>
</tbody>
</table>

![Decision-Making tool](image)

(1) Definition
(2) Decision
(3) Results

Fig. 9. Decision-Making tool.

This process involves selecting among a set of possible answers. This completes a process of Definition, Answer and Results. The definition is made by the student who proposes the question (the voting initiator), and the answer is carried out by all the group members. They all receive the results at the end of this process. There is a limited time to answer, and when this time is over, the non-activated vote is considered an abstention. With this tool, democracy is incorporated in the collaborative session.

This tool is internally used by the system to provide the group with synchronized navigation through the work subspaces. When a student proposes the navigation, a yes/no question takes place, and when all the users agree the group moves to the new subspace.
4. Implementation and evaluation of DomoSim-TPC

The system, entirely implemented in Java, is based on the client/server model in TCP/IP networks for its operation on the Internet/intranet, allowing distance work. The data access is achieved by means of JDBC and the communication architecture is based on a centralized approach. The synchronization techniques operate with sockets because of their efficiency. Through these sockets compressed and optimised data packets are transmitted in order to reach an optimal performance. Servers with different operating systems (Microsoft Windows and Linux) and diverse DBMS (Microsoft Access, Oracle and MySQL) have been used.

The development of the environment is based on iterative design, prototyping and participative design. Formative (Tessmer, 1993) and heuristic (Baker, Greenberg, & Gutwin, 2000) evaluation has been applied during its development, and real experiments with students have been carried out in educational centres. A total number of 69 activities have been carried out (Table 2), in which 50 students, 2 teachers, 2 experts in Domotics and one usability expert have taken part.

The EXP experiment allowed us to verify the Domotics model and to evaluate the environment at a general level, so that different limitations were detected and solved and user requirements were incorporated in the prototypes obtained in the development of the system. The CHI and UI experiments, carried out with students who had studied the Human–Computer Interaction and User Interfaces subjects, aimed to evaluate the environment as a collaborative system, emphasising its usability.

The FP experiment consisted of using the environment in real Domotics teaching situations. The aim was, on the one hand, to benefit the educational community and, on the other hand, to confirm the improvement that takes place in Domotics learning and in the students’ problem solving skills.

Two teachers and 14 students from two different teaching centres took part in this experiment. They approached the solution of seven problems (p1–p7) of growing complexity in a total of 47 activities. These students were organized in pairs. In 74.5% of the problem solving activities a design that was simulated once or several times was carried out, and in 10.6% of the cases a model that was not simulated was designed. This experiment has been analysed in depth by Bravo (2002), studying aspects such as the size, time and quality of the solution, the favourite way of communication and the students’ learning. For this purpose a quantitative and qualitative analysis tool incorporated in DomoSim-TPC has been used. Next, some results are described.

The time used to solve a problem depends on its size. The average solution time by room and management area of the problems p1–p4 by the groups g1–g6 has been calculated (Fig. 10). In the

<table>
<thead>
<tr>
<th>Experiences</th>
<th>Activities</th>
<th>Users</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Students</td>
<td>Teachers</td>
</tr>
<tr>
<td>EXP</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>CHI</td>
<td>13</td>
<td>26</td>
</tr>
<tr>
<td>FP</td>
<td>47</td>
<td>14</td>
</tr>
<tr>
<td>UI</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Totals</td>
<td>69</td>
<td>50</td>
</tr>
</tbody>
</table>

Table 2
Experiments with DomoSim-TPC
first problem the students used up a lot of time (47.9 min) because this was the first problem that the students had faced after the training process of DomoSim TPC and they found it difficult to work quickly. In the problem p2 they took even more time (60.1 min), for the same reason and because the administration area is different to that of the first problem. In the problems p3 and p4 the average time by room and area begins to decrease (25.9 and 13.3 min, respectively): the students have got used to the system and they start to apply what they have practiced and learned in the previous problems, for this reason they work faster.

When the students continue to advance in the problem solving they make fewer errors, decreasing the relationship between deleted objects and inserted objects. In Fig. 11 the percentage of deleted objects with respect to inserted objects for the problems p1–p5, which have been solved by groups g1–g6, is shown. In groups g1 and g4 there is a general tendency to decrease in the deletions. Groups g3 and g6 keep their deletion percentage below 20%, which means that these groups make few mistakes and they do not need to delete many objects. Group g2 has a tendency to decrease the number of deletions from problem p2 onwards. Only group g5 has gone on increasing the number of deletions made, which surprisingly decreases considerably in the last problem. Studying this case in more detail, we have realized that this group hardly worked out problem p4, with as many deletions as insertions (100%) being made.

If all the simulation actions experimented automatically by the model (turn an operator on/off, malfunction of an element, etc.) with respect to the interactions carried out manually by the students are considered, a ratio of 8/1 is obtained. This indicates a high degree of students’ participation introducing changes in the model to explore its behaviour.

With respect to communication, when observing how the students behave during the simulations, we can see that they talk and discuss. The conversations arise in response to the events
and results of the simulation and they refer to aspects of it. In this way, the simulation is good as a
stimulus and resource to provoke conversations and discussions by means of which the learners,
in group, build on their own knowledge.

The problems p6 and p7 have been solved by pairs as well as by individual users. This allows
their solutions to be compared. Based on the evaluation of these solutions by the teachers, we
have realized that the pairs carry out models of better quality, although they need more time
due to the necessary coordination. The individual users take less time, but due to the complexity
of these problems they find it more difficult to obtain a solution that solves the problem
satisfactorily.

When concluding these activities some post-tests have been carried out. On a scale from 5 (yes,
very much) to 1 (no, nothing), the students indicated that they had learned something of Domotics
with these experiments (4.1 points), that they considered group work more positive in contrast
with the individual work during the design and simulation (4.7 points), that the simulation is a
good mechanism to refine the design in successive stages, allowing them to discover aspects that
lead them to learn (4.1 points), that the need for face-to-face meetings for the carrying out of prac-
tical exercises decreases with this kind of support (3.7 points), that the division in the Distribu-
tion–Design–Parameterisation–Simulation stages for design is a good choice (4.4 points), and
that the approach of synchronous design we have modelled is realistic and effective (4.5 points).

To summarize and gathering all the obtained data, we can conclude that collaboration assists
learners positively in the resolution of these kinds of problems. With DomoSim-TPC the students
learn Domotics, apply the knowledge they have acquired, generate new knowledge and solve the
problems better and faster every time.

Nevertheless, in order for work with the system to be productive certain circumstances are re-
quired. The learning experiences should be planned and organized carefully. This requires some
effort from the teachers, since they have to manage a lot of information. Before approaching
the problem solving, the students should receive complete training in the use of the system.
The efficiency of the synchronization in the navigation through workspaces or of the decision-
making processes depends on the degree of experience of the students in the use of the system.
In order that the Guided Chat is effective the students should understand the different communi-
cation possibilities. There are processes in which the presence of conflicts of intentions reduces the
efficiency or blocks the system use. The lack of participation (students that do not answer or
work) or the lack of agreement (contradictory intentions) can hinder the development of the work
and of the students’ learning.

Certainly, technology is fundamental to the domotical design and its learning. The execution of
the simulation model and the complexity of the calculations to check if the requirements and con-
straints of the problem are verified make computer support necessary. Technology also makes
working at a distance possible and offers a computational environment in which to manipulate
shared artefacts and to synchronize the users’ work.

5. Conclusions and future work

In this research we have approached the collaborative learning of design. We have modelled a
design process and a teaching situation that has allowed us to propose a collaborative learning
method in complex design domains based on the PBL and LBD methods. This learning method is learner-centred: the learner is active and solves problems similar to those in real life; collaboration among students can increase motivation and the teacher is a tutor and a facilitator of the learning. Starting out from our model and method, we have approached the Domotics domain and a CSCL environment has been built. With this environment the students manipulate artefacts using interactive visual representations, experiment with these solutions and reflect on improving their solutions in an iterative design process. The proposed methodology and the collaborative strategies go beyond this system and form a framework for the creation of learning environments, which use real time collaborative design and simulation, adapted to other domains or domain-independent.

The collaborative work is organized according to the structuring principle. The global process, the execution of certain tasks and communication have been structured by means of Collaboration Protocols, the use of Language as Action and Flexible Structuring, respectively. In relation to structuring, we do not believe that greater freedom contributes positively to the resolution of problems in learning processes, and consequently with our system we have aimed to add the advantages of structuring to the ones of free and interactive work on shared work surfaces, thus offering an environment that combines phases of both kinds.

Although asynchronous CSCL tools have several advantages, one of them being the more reflexive condition of the contributions, we favour synchronous collaboration, now easier to achieve thanks to the technological advances in hardware and software. As in real life, we believe that some domains or tasks are more appropriate for asynchronous collaboration (reflection) whilst others are more appropriate for synchronous collaboration (immediacy and spontaneity). Clearly, a symbiosis of both may be the best solution depending on the kind of learning tasks. This is the approach we have followed with the phases of planning (asynchronous) and design (synchronous).

After the experiments with the environment, we have seen that in addition to learning how to design, which is the main objective, the student should learn how to collaborate, becoming responsible for their tasks and coordinating with others. We have realized that when the students distribute their tasks, design, simulate and refine the solution according to the design restrictions, their knowledge of the domain and their abilities are consolidated. In addition, we can see that the time spent on problem solving decreases when it is carried out in a group: several people obtain a better solution in complex projects than one individual.

Currently, we are doing a study to identify the influence of the process carried out by the group on the solution (design) created using Artificial Intelligence techniques. In the near future we are going to look at aspects of reusability and generalization. We will use specification languages for the generic definition of case studies, describing object models and behaviour, which represent new domains for the application of our approach. Some of these domains, where design and simulation can be used, could be the design of information systems, of networks and of circuits (logical, electric or electronic circuits).

References


