

VIRTUAL COMMUNITY IN THE CLASSROOM: AN INNOVATING TOOL FOR ELEARNING

F. Bruschi¹, P. Faverio², R. Hodges², L. Mari², D. Restelli² and D. Sciuto¹

¹*Politecnico di Milano, Dipartimento di Elettronica e Informazione, P.zza L. Da Vinci 32, 20133 Milano, ITALY;* ²*LIUC, CETIC, C.so Matteotti 15, 21053 Castellanza, Varese, ITALY*

Abstract: This research in progress introduces a learning architecture where ICT tools complement the traditional models of formative process management through a moderated virtual community in the classroom. In particular, the research presents the main features of the conceptual framework leading to the implementation of such architecture in a web-based system. Among the several experiences of its usage, we report here the example of the application of this framework in a class of Information Systems Planning, to teach how to model the requirements of a system and it is being customized to teach modeling of hardware systems.

Key words: virtual community; e-learning framework; web-based system; community software architecture.

1. INTRODUCTION

The most typical use of the new information and communication technologies (ICT) in the learning processes is *substitutive* when the learner can't physically be where the lessons take place (Piccoli, 2001; Smith et al., 2001). The lack of the community dimension, which is so important in learning processes (Kaplan, 2002; Mitchell and Hope, 2000), justifies the usual denomination of "distance learning" for these scenarios. On the contrary, the learning architecture presented here is founded on the strategic choice of adopting ICT to *complement* the traditional management of learning processes. It is a technology-based approach, whose aim is to create new formative situations in the classroom (Girod and Cavanaugh, 2001; Marold et al, 2000).

This architecture differs from traditional supporting tools for another fundamental reason. Traditional learning processes are usually based on two very different formative configurations: classroom sessions and individual study. It is noticeable how these configurations diverge with respect to the media employed (see Table 1):

Table 1. The two possible configurations of traditional learning processes

Classroom sessions	Individual study
interaction person → person	interaction person → didactic material
greatly multimedial and broad band	unimedial and narrowband
not repeatable without a teacher	constantly repeated without a teacher

The “intermediate” formative situations (e.g. practice in laboratories) witness the importance of innovative training solutions as a middle course between these extremes of a potential *formative continuum* (Guttormsen Schär and Krueger, 2000; Levine, 2002; Saunders and Werner, 2000). This research in progress presents the fundamentals of the *moderated Virtual Community (VC) in the classroom*, a software-based environment that represents an attempt to experiment the educational potentialities of these “intermediate” formative situations.

2. THE MODEL OF THE MODERATED VIRTUAL COMMUNITY IN THE CLASSROOM

The idea of using ICT tools as a support for discussion among groups of students for teaching purposes is clearly not new: typical examples can be found in the so-called forum and chat systems, supporting the management of asynchronous and synchronous communication, respectively. In a communicational perspective, these systems are extremely poor: they do not carry any contents in themselves and they are weakly structured and structuring.

On the contrary, our model of the VC is based on the structuring of information according to the teaching purposes: that is why the VC is driven by a learning path organized through subsequent phases. Within each phase a discussion between the participants takes place, being defined both the subject and the planned outputs of the discussion (Simpson, 2002; Smith et al., 2001). Unlike what happens in forums and chats, in the VC the role of the moderator is neither accidental, nor banally censorious.

The choice of experimenting with this model of VC in the classroom, instead of among students interacting at a distance, originates from the interest in empirically testing one of the hypotheses which have inspired the

eLearning project (<http://elearning.liuc.it>) where the VC took place: unlike what is often maintained, eLearning *is not synonymous with* distance learning; more specifically, distance learning supported by ICT tools represents just an example of eLearning. In particular, the usage of the VC has led us to verify the possibility of managing everybody-with-everybody discussions in the classroom, a communication setting that would be unfeasible with traditional modalities (Mitchell and Bacic, 2000).

The model of VC we have defined interprets a discussion as a game structured in a temporally linear sequence of *game phases*, each phase being characterized by the different involvement of three actors (see Table 2): the configurer (the *coach*), the student (the *player*) and the teacher/tutor (the *referee*).

Before the actual start of the game, in the *set* state the *coach* configures the VC system, by specifying the number of phases and for each phase:

- *its ordinal position within the play sequence*;
- *its information/communication structure*, defined according to the Composite View design pattern: the front end of the VC system with which the players interact during the game is a table of one or more elements, each one of them being chosen among a set of pre-defined high-level configurable components, e.g., a read-only text box (used in particular by the *coach* himself to specify some descriptive characteristics of the phase, such as its title and aims), a read/write area, a chat subsystem, a forum subsystem, ...; as a further option, the *coach* can specify that the default contents of an element will be generated runtime, as a copy of the data introduced in a previous phase by the *players*;
- *the players' identity, role and permissions*: players can be impersonated by either single students, pre-defined groups, or groups established by the *referee*, according to the results of the previous phases); moreover, the *coach* sets the exclusive or concurrent access to each element of the front-end structure and, in the latter case, the corresponding read/write rights.

The second actor, the *player*, interacts with the VC previously configured (the *set*) by the *coach*, and the *referee* behaves as a player interacting with the participants. However, it is only up to the *referee* to establish the timing for the activation and the conclusion of the phases and to evaluate the contents provided by the players during each phase. This last operation is necessary in order to configure the next *runtime* phase, called "*hidden game*".

Table 2. The range of actions of each actor in the VC

State	Actors		
	coach	referee	player
Set	X		
Game		X	X
Hidden game		X	

A formal model for the Virtual Community can be represented as stack automaton, constituted by four states $Q = \{q_0, q_1, q_2, q_3\}$, where:

- q_0 = Set (initial state);
- q_1 = Hidden Game;
- q_2 = Game;
- q_3 = End of the Game (final state $F = \{q_3\}$).

The edges notation chosen is the following: $\alpha, \beta/\beta\gamma$, where the information carried out is the following:

- α : indicates the action performed on the Virtual Community;
- β : indicates the reading action of an information on the Virtual Community, stored in a previous action;
- γ : corresponds to the storing action of an information concerning the state of the Virtual Community.

The notation $\alpha, \beta/\beta\gamma$ describes the situation in which the character β is reported as read and the character γ is added to the stack. Furthermore the notation $\alpha, \beta/\beta$ shows that β is read and reported on the stack with or without an alteration done by an actor.

The set of characters allowed on the stack ($\beta \in \gamma$) are:

- Z_0 : default character for the initialization of the stack;
- A: character that identifies a *game* phase. The total number of characters in the stack identifies the number of phases for the given Virtual Community;
- B: character that defines the end of the state *set* and the possibility of beginning the Virtual Community;
- C: character that identifies the end of a *game* phase and the beginning of a *hidden game* phase.

The alphabet of characters on the input data stream is the following:

- a: corresponds to the insertion of an information during a phase;
- b: corresponds to the insertion of an information that ends a configuration to go from a *set* phase into a *hidden game* phase;
- c: corresponds to the insertion of an information that defines the transition from a *hidden game* phase to a *game* phase;
- d: corresponds to the insertion of an information during the *game* phase;
- f: corresponds to the insertion of an information that defines the transition from a *game* phase to a *hidden game* phase;

- g: corresponds to the insertion of an information during the *hidden game* phase.

The three classes of participants to the Virtual Community are prefixes of the alphabet in the input data stream, and are indicated by:

- c: coach;
- r: referee;
- p: player.

Given the defined grammar, the finite state model representing the Virtual Community is shown in Figure 1.

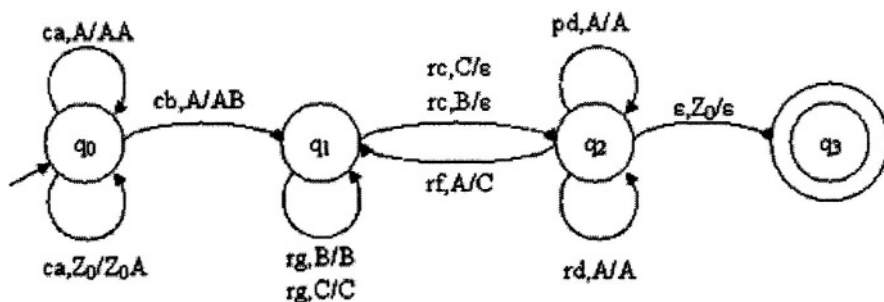


Figure 1. The finite state automaton representing the Virtual Community

3. THE DEVELOPMENT TOOLS

From a technological perspective, the VC is based on three-tier architecture that has been implemented by an Intranet web-based system.

The system development phase has been focused on the three logical parts typical of three-tier architectures: the user interface (clients), the business logic (web server) and the data manager (DBMS).

The critical factors that have driven the technological choices for the implementation were the possibility of a fast deployment to experiment, an inexpensive implementation and the use of user friendly interfaces.

Each of these modules is characterized by some critical factors which have driven the resulting technology choices. As a consequence, HTML has been chosen to build the presentation layer, which is, in turn, dynamically generated by a server side script. Consequently, actors can simply use a browser to operate the virtual community without the need of setting up any other software.

The business logic module and the data manager modules have been implemented using Microsoft Windows 2000 server and Microsoft SQL Server and no other solution has been explored.

The combined use of the Microsoft web server family (IIS, Internet Information Server) and of ActiveX components has been useful to build a simple architecture which allows management of files, scripts and the access to a database. In particular, IIS supports the ASP (Active Server Pages) environment which interprets and executes the server side scripts needed to implement the business logic described in the theoretical model, to generate the specific user interfaces for each game phase and, finally, to interface the application with the DBMS.

4. AN EXPERIENCE OF VIRTUAL COMMUNITY

After some years of experimentation of the VC in short courses, the validity of the teaching assumptions of the model has been recently more significantly tested during a seminar “Cases of Business Information Systems”, held at University Cattaneo – LIUC and attended by 50 students. During the 15 hours of the seminar the students were introduced to the problem of business information flow analysis through two case studies, presented in electronic format (an e-book) and structured accordingly, so that each couple of students, thanks to the support of a VC, could experiment with a problem-solving approach. The constant interactivity which derived from this, together with the quality of the proposed cases with respect to both realism and complexity, made it possible to build a situation where the students progressively played the part intended for them by the VC, simulating the role of analysts/business consultants. Though with the limits of an estimation which is only quantitative, the number of proposals submitted by the different groups in each phase seems to confirm that the VC can be an effective tool to develop a modality of interaction among workgroups which is based on the maximum cooperation and productivity, more than what happens when a face to face interaction is requested.

This successful experience can be replicated on different subjects. In particular, we decided to verify its applicability in the modeling phase of digital systems, i.e. the most complex competence to grasp for undergraduate students in engineering courses.

In this example of Virtual Community, different players have to face the problem of partitioning a RSA public-private key generation algorithm into sub-functionalities, suitable for a subsequent hardware implementation. The players, that can be either individual students or small group, can access the specification of the system to be specified, written in natural language. As

starting point, the algorithm for the pair generation is analyzed by the players. As an additional input, a library of general purpose functional components is given.

In the first phase, players propose a partitioning of the algorithm into interacting sub-functionalities, by compiling the row heading of a table, whose columns represent the following information:

- *provided interface*: set of services that the sub-functionality must offer, described with the corresponding input data types;
- *required interface*: set of services that the sub-functionality requires to be offered by other sub-functionalities, in terms of data types employed;
- *type*: classification of the functionality among the following classes: data intensive, control intensive, storage;
- *library IP*: setting the value of this field (that can be either *yes* or *no*), the student gives his opinion on the possibility to implement the functionality with a particular component of a library. The considered library is given as an input of the game;
- *target component*: if the value of the *library IP* field is *yes*, in this field the library component on which the functionality can be mapped must be pointed out; otherwise, another resource (processor, custom logic, etc.) must be indicated.
- *decomposable*: the value of this field records the opinion of the player on the opportunity of further decomposing the functionality into simpler tasks.

The first two columns contain information that in most design languages such as VHDL can be directly exploited to characterize modules boundaries. The other columns contain information that summarize some significant design choices.

Table 3 shows the table with the identification of the sub-functionalities that must be ratified during the hidden game by the referee. The column headings are preset by the coach during the system configuration, while the row headings result from the ratification and aggregation of the proposals of the two players. After the ratification, the players will have the same row headings, i.e. the same functionalities decomposition, and will be required to complete the columns. All tables are then analyzed by the referee who will ratify each single cell, identifying the correct answers and providing to both players a single view, which may be a mix of both tables or may be a modified version by the referee.

Table 3. Output for the ratification of sub functionalities phase

functionality	provided interface	required interface	type	library IP	target component	decomposable
random number generation						
primality testing						
prime numbers caching						
prime numbers multiplication						
exponentiation						
small random numbers generation						

Table 4 shows an output example for the “hidden game” phase of ratification of the functionalities proposed in phase 1.

Table 4. Output for the phase of ratification of the proposals made by the two players (decomp.: decomposable)

functionality	provided interface	required interface	type	library IP	target component	decomp.	player
random number generation	seed: bit[512]	rnd: bit[512]	Data intensive	no	processor	yes	A
primality testing	number: bit[512]	isprime: bit	Data intensive	no	custom logic	no	B
prime numbers caching	store: bit	p: bit[512]	Storage	yes	bank register	no	A
	prime: bit[512]	q: bit[512]					A
prime numbers multiplication	p: bit[512]	n: bit[1024]	Data intensive	yes	multiplier	no	A
	q: bit[512]						A
exponentiation	p: bit[512]	d: bit[512]	Data intensive	no	custom logic	no	B
	q: bit[512]						B
	e: bit[512]						B
small random numbers generation	seed: bit[512]	e: bit[512]	Data intensive	no	processor	no	A

Next phase of the game foresees a synchronous discussion between players working on the same specification. This is implemented in the Virtual Community through a chat, to keep track of the discussion among the “competing” players about the table ratified by the referee in the previous phase. As an output a finale table, agreed upon by the group is obtained as output. In the hidden game phase, the referee reviews the output of the discussion and approves (or modifies) the table.

Next step of the game may be to restart the process with those modules that could be further decomposed, always considering two competing players.

Furthermore, different uses of the Virtual Community can be considered. For instance a competition among players in the code writing phase (either SystemC or VHDL or Verilog or any other language for which a simulator is available) could be setup. The goal is to teach students to write correct and efficient hardware specifications, by stimulating both competition and cooperation. The same could be said considering also the results of the synthesis, by making them discuss both the constraints and parameters set to the synthesis tool, and the results obtained, starting from a synthesizable version of the code.

5. CONCLUDING REMARKS

Aim of this paper was to present the model of a Virtual Community and its supporting framework. The results are encouraging and have shown that such a virtual community is very flexible and allows the creation of different learning situations that stimulate the students' interactivity. The experiences made so far have shown that, thanks to the confrontation and the direct cooperation required to the students, they become active elements in the class instead of passive listeners.

An interesting feature of the proposed model of Virtual Community is that it allows the possibility to create and alternate both competitive and cooperative situations during the same game. This aspect has been perceived as fundamental by the students that attended the Information Systems seminar. They have valued the experience as extremely positive. In particular, the students have pointed out the psychological effect which, thanks to the minor exposition filtered through computer mediated communication, helps to reduce the communicative and exposure difficulties which are typical of direct interpersonal relationships.

The results of this didactic experience have shown how the synergic employment of different hypermedial tools is useful to improve learning. In the classroom, the chat has enabled a quick and efficient communication, thanks to its synchronicity and focalization on problem solving. Moreover, the input of information in a highly structured context has made it possible to focus on the operative scope, with the benefit of the conciseness of the contents (for both students and teacher). Finally, the autonomous employment of the forum has allowed reaching a higher quality of the answers, significantly higher than the ones that what would be obtained through traditional tools.

REFERENCES

- Girod, M., and Cavanaugh, S., 2001, Technology as an Agent of Change in Teacher Practice, *T.H.E. Online Journal*, <http://www.thejournal.com/magazine/vault/A3429.cfm>
- Guttormsen Schär, and S., Krueger, H., 2000, Using New Learning Technologies with Multimedia, *IEEE Multimedia* 7(3):40-51, <http://computer.org/multimedia/mu2000/u3040abs.htm>
- Kaplan, S., 2002, Building Communities – Strategies for Collaborative Learning, Learning Circuits, August 2002, <http://www.learningcircuits.org/2002/aug2002/kaplan.html>
- Levine, L.E., 2002, Using Technology to Enhance the Classroom Environment, *T.H.E. Online Journal*, <http://www.thejournal.com/magazine/vault/A3819.cfm>
- Marold K.A., Larsen G., and Moreno A., 2000, Web-based Learning: Is It Working? A Comparison of Student Performance and Achievement in Web-based Courses and their In-classroom Counterparts, in: *Challenges of Information Technology Management in the 21st Century*, M. Khosrowpour, ed., Idea Group Publishing, Hershey, USA, pp. 350-353.
- Mitchell G., and Hope B., 2000, Educational Technology and Emergent Forms of Education, *Challenges of Information Technology Management in the 21st Century*, M. Khosrowpour, ed., Idea Group Publishing, Hershey, USA, pp. 833-834.
- Mitchell, S., and Bacic, M., 2000, Using Real-Time Chat to Increase Student Interaction, *Learning Technology*, IEEE Computer Society, http://ltf.ieee.org/learn_tech/issues/july2000/index.html#mitchell
- Piccoli, G., Ahmad, R., and Ives, B., 2001, Web-Based Virtual Learning Environments: A Research Framework and a Preliminary Assessment of Effectiveness in Basic IT Skills Training, *MIS Quarterly* 25(4):401-426.
- Saunders P., and Werner K., 2000, Finding the right blend for effective learning, Center for Teaching and Learning, Western Michigan University, <http://www.wmich.edu/teachlearn/new/blended.htm>
- Simpson, W., 2002, Online learning: a Project to Develop an Innovative Approach to Control, *Learning Technology*, IEEE Computer Society, http://ltf.ieee.org/learn_tech/issues/july2002/index.html#8
- Smith, G.G., Ferguson, D., and Caris, M., 2001, Teaching College Courses Online vs Face-to-Face, *T.H.E. Online Journal*, <http://www.thejournal.com/magazine/vault/A3407.cfm>
- Syed, M. R., 2001, Diminishing the Distance in Distance Education, *IEEE Multimedia* 8(3): 18-20, <http://computer.org/multimedia/mu2001/u3018abs.htm>
- Tsinakos, A., and Margaritis, G., 2001, Employing Case Based Reasoning in Asynchronous Distance Education, *Learning Technology*, IEEE Computer Society, http://ltf.ieee.org/learn_tech/issues/july2001/index.html#3