



FACULDADE DE INFORMÁTICA
PUCRS – Brazil
<http://www.inf.pucrs.br>

**Modelling and implementing educational environments
using an interdisciplinary approach**

Lucia Maria Martins Giraffa

TECHNICAL REPORT SERIES

Number 019
October, 2001

Contact:

giraffa@inf.pucrs.br

<http://www.inf.pucrs.br/~giraffa>

Lucia Maria Martins Giraffa works at PUCRS/Brazil since 1986. She is titular professor and leader of the GIE Group. She develops research in Multi-agent systems, Artificial Intelligence in Education, and design of educational software. She got her Ph.D. in 1999 at UFRGS (Brazil).

Copyright © Faculdade de Informática – PUCRS
Published by – PPGCC- FACIN – PUCRS
Av. Ipiranga, 6681, predio 16
90619-900 Porto Alegre – RS – Brazil

Modelling and implementing educational environments using an interdisciplinary approach²

Lucia M. M. Giraffa, Ph.D.

Faculty of Informatics
Computer Science Post- Graduate Program – PUCRS
Artificial Intelligence in Education Research Group
Av. Ipiranga 6681 – prédio 16 – Porto Alegre – RS – Brazil 90610-900
Phone/Fax: + 55 51 320-3611
giraffa@inf.pucrs.br
<http://www.inf.pucrs.br/~giraffa>

Abstract

The aim of our research is developing educational software to assist teaching-learning process from elementary schools to college. We argue that all educational environments have to concern about pedagogical aspects in order to guarantee better quality for students and teachers activities during the educational process support by educational software. To achieve this we must to observe some guidelines before the software modelling process itself (Computer Science viewpoint)

This paper presents some guidelines to design and implement computerised educational environments that have been used in our group to guide our projects. We discuss some important considerations intrinsically in educational environment design, and presents the results achieved with this approach trough four tested environments developed under our supervision. The guidelines and results involve (Intelligent Tutoring System) architecture, multi-agent system architecture (MAS) with game like fashion interfaces, and Collaborative Learning.

Keywords – Educational software, learning environments, design of educational environments.

I. Introduction

The use of computerised environments for teaching-learning activities transforms and enriches the way we represent, store, and communicate information in schools. Students and teachers need to develop and acquire new cognitive, affective and social skills. The new medias integrated by computer offer us an opportunity to create new methodologies in order to improve the learning process, and overcome limitations of traditional methodologies in schools based on classical resources (blackboard, charts, graphics, and so on). We are not argue that traditional methodologies are not efficient because they are “classical”. We are saying that some subjects are resource bounded, e.g., they are very difficult to explore under the teaching-learning viewpoint. As an example we can present Chemical and Physical phenomena, Biology subjects like evolution, food chain, and so on. Nowadays teachers have to explore the new technologies like computers and its networks to support the learning process, and enlarge the possibilities to explore the set of subjects in different and unusual ways.

² Research supported by CNPq-Brazil

Computerised educational environments that support flexible methodologies are quite hard to find. Currently, we can find many tools and environments that allow cooperative work like WebCT³, GroupKit⁴, Microsoft Chat⁵, Microsoft Netmeeting⁶ e Cu-See-Me⁷. Many of them are already used in Brazilian schools. However, there are still many difficulties for the teaching staff and student of elementary schools to use them. This is due to the set of computational prerequisites like abilities to handle with different resources at the same time, different interfaces for the same purpose, how to digitalise images with suitable pattern, and so on.

If we concentrate our analysis on educational applications (stand alone or for groups) developed for specifically purpose (Chemical, History, Geography, Ecology, Math, and so on) we will find another aspects to be taken in account. It is very important to remind that educational process is closing connected with country culture. Educational environments that work very well in a specific context will not necessarily perform in the same way in a different place. It can be viewed as a restriction in many ways: as applicability for other educational reality, pedagogical paradigm, and so on. However, we argue that educational process has a “kernel” of necessities and aspects to be observe when you intend to work considering learner as central person in the educational process, under the *‘learn-to-learn’* paradigm. So, it causes a immediately reflection in the way we design and model an educational software.

We believe that teachers need to create and apply new methodologies in order to promote learning in a more efficient way. So, an educational environment must provide tools for the teachers to explore different tasks using methodologies that allow students to explore the domain, and understand the way they acquire knowledge about something. To achieve this we need to offer open environments that allow teachers and students to test hypotheses, and create alternatives to better understand the different possibilities that arise from a specific domain exploration.

This paper is organised as follows: section 2 presents some issues to be considered to develop educational software and our approach to guide the educational software project. Section 3 presents four examples of educational environments tested with children and teacher from elementary schools, and discuss the way we projected them, and made the evaluation process under pedagogical viewpoint. Section 4 brings up some matters to be discussed about design and evaluate educational software. Section 5 presents the references.

II. Designing educational software

We have developed a research with teachers in elementary schools in order to identify their necessities and to elicit them concerning to:

- software selection: The way they selected software to use with their students;
- methodologies adopted when using educational environments as a

³ <http://www.webct.com/webct> or <http://www.homebrew.cs.ubc.ca/webct/>

⁴ <http://www.cpsc.ucagary.ca/grouplab/groupkit>

⁵ http://www.microsoft.com/windows/ie_intl/br/chat/

⁶ http://www.microsoft.com/windows/ie_intl/br/netmeeting/

⁷ <http://www.cu-seeme.net/>

complementary resource for their classes;

- Resources that could be included in an environment that intends to support students in stand alone or cooperative work.

We began our work in 1989, and since then the author has been the group leader and co-ordinator of many projects. The aim of our first research was to identify school reality concerning about the use of computer as a tool, to support teaching-learning activities [3]. We developed a research in our city and surroundings, considering private and public schools. We found a picture not so enthusiastic: inadequate conditions (hardware and software) to develop educational activities in schools, teachers without necessary skills about computers, and specially computers apply to education.

There was (and still there is) a big gap between theory and practice. We made two more researches with the same rigorous control (methodological aspects) in the same target public [2; 8; 9; 10; 15] and found almost the same situation. Ten years ago we had rudimentary interfaces (character mode) they were a problem for students and teachers interact with the systems. They need much knowledge of operated system to interact with the educational software. Nowadays, we have interfaces with high level of intuition and the problem still persist. If the interfaces (system interaction) improve, the necessary set of prerequisites increases as well. In different levels the problem of adequacy of teachers background still remains as a big challenger to be overcome. Those days we have to consider not just the *one-to-one* interaction. Now, we have the Internet and the cooperative environment that integrates tools to facility the communication among students and teachers. The set of user skills to use educational environment enlarges.

In order to try to solve this gap between theory and practice (researches and “real life in schools”) we created a *latu-sensu* graduate course named IESC (Informatics apply to Education Specialisation Course). The aim of that course was to spray out in our community our beliefs and research results about Computer Science apply to Education. Our research group has a large experience with teacher’s formation because the IESC was offered from 1992 until 1998, for many different groups of teachers from elementary and high school. After 1998, we concentrated our efforts on educational software design.

Since the beginning of our research we observed the gap between educational software design and other applications like banking, commercial, industry, etc. Almost all the systems designed to banks, industry and so on have a formal modelling and they use Software Engineering resources. However, with educational software projects are the opposite. It very rare (and recently) to find an educational environment with a carefully project that observe methodological issues, even under Computer Science (CC(or Pedagogical viewpoint. Due this gap between educational community and the others society communities, the author decide to create some guidelines to model and design educational software.

These guidelines were refined along the years, during many projects under our supervision (Abracadabra [14], Tunel do tempo [7], Gutemberg [4], Wholler [13], HEI [1], Eco-logico [15], MCOE[8;9;10], PINGUIM [11], IMAGUS [16] and Tirinha [2]). They became part of a methodology to develop educational software that considers two parts in the educational software project:

1. the definition about the environment concerning to pedagogical viewpoint, connected to Computer Sciences (CS) aspects (guidelines). The set of steps that will be followed by the project team members. This first part it is very important to do because the main decisions about the system will be doing. The elicitation process will give elements for the second part. We argue the importance to clarify those elements before working with a more formal process;

2. the formal system modelling related with CS (methodologies). After deciding the main guidelines and specifications concerning about software pedagogical aspects, we need to specify the sequence of activities to be followed during the developing process. It is necessary to identify the life cycle for the application that must be always compared with the real evolution of the system [Koz98]. The developing methodology (life cycle) is different from the technique or method selected to specify the system itself. The methodology must have a certain degree of independence comparing to the set of chosen techniques.

In this paper we are focussed on the seven steps to be followed in order to attend the first part of our framework:

1. Select the subject (domain). We believe that is impossible to create good educational software to general purpose or a large set of contents. We need to select a specific subject to be modelled in a deep way. Especially when we try to create environments based on Intelligent Tutoring Systems, where we need to model students and tutor interventions related with students mistakes. It is very hard to create a good choreography (the set of activities to be taken by an artificial tutor connecting with the mistake made by student when s/he is solving a problem);

2. Select the system users. It is important define for what range of age the system will be concerned. We believe that it is impossible to model a general environment that is suitable from “six to sixty”. Each group has special characteristic that must to be taken in account when we define interfaces, user interaction, the way to present some tips, some rules, etc.

3. Select the modality of educational software. It means decide what kind of software will be adopted: tutorial, drill and practice, simulation, educational games, and so on. For us the traditional taxonomy to classify educational software is not suitable with the current reality. Those taxonomies were related with a classical paradigm that classify software according their resources and interfaces characteristics. Our group adopted a constructivist paradigm, and for us the educational software can be classified by the type of knowledge (skills) that it helps to acquire (student’s viewpoint) [10]. We divided the skills to be acquire in two groups: *specific skills* (basic levels of Bloom cognitive domain taxonomy – memorisation, identification, recognition, understanding, application in specific situations) and *wide cognitive skills* (synthesise, generalisation, and problem solving).

Currently it is very hard to classify an educational software using traditional taxonomies, because the interfaces integrates many resources that may confuse the user that intends to do it. For us it is not the time to classify educational software according observing characteristic. It is time to analyse the environments observing their potential

to achieve educational purposes. Nowadays, they are more than just a program, they become environments with many resources and options to be explored, even by teachers (using or creating new methodologies) or by students (testing hypothesis, discovering facts, and so on).

4. Select team members. After selecting a subject, we need a specialist for domain modelling. S/he is responsible to select exercises, strategies and tactics related to student's mistakes. Educational games modelling need also a specialist to help us with interface (a specialist in Computer Graphics and simulation), and last, but not the least, a specialist in Informatics apply to Education. When an ITS (Intelligent Tutoring System) architecture is used as a base to design the system, we need some help from AI (Artificial Intelligence) people;

5. Design the interface. The interface is the communication window between the user and the system. It must be carefully designed. We depart from the interface elements to guide all system design. Our experience showed us that are very important test interface with user and teachers to better guarantee that it will work according our expectations. The interface design and implementation must be performed using an incrementally approach. We use Nielsen guideline [14] to guarantee that the final interface version will be accepted by users.

6. Select the environment to implement the system (tool or set of tools). The choice of specific tools or environment to implement the system is related with environment characteristics. The computer science support it is very important after deciding the steps above. Currently, there are many possibilities to implement an environment. Consult specialists in order to decide what are the best is a safe way to avoid disappointments. The computer science specialists have knowledge to analyse the role project and indicate what is better, taking in account the hardware where the system will be performed. It is very common to find updated computer in universities, but not in schools. Sometimes we have to drop the use of the ultimate technology because it is not available in schools. We must remember that our client is there.

7. System evaluation. Nowadays we need to perform some kind of evaluation process in order to guarantee the quality of system under educational aspects. We can design and implement a great system under Computer Science viewpoint. However, it might be not so good under educational viewpoint. We must consider at the same time both aspects. In our experience, we faced with some situation where the solution was not so creative or good enough under system performance (computer science aspect). However, it was necessary in order to guarantee system quality under the pedagogical viewpoint. In an interdisciplinary team, sometimes it is very difficult to handle this. The team members seem not to agree with the final decision that must be taken.

Evaluating a system under pedagogical viewpoint is something quite hard to do. If you decide to measure students learning process it can be very difficult to perform in a short period of time. The student's learning evaluation is a process that always remains after concluding a prototype connected with a ungraduate final work, a Master Dissertation or Ph.D. thesis. To measure how much a specific system helped the students to acquire knowledge and skills about something we need a rigorous scientific methodology with statistics analysis, groups control, and time, much time. So, what we can do in order to perform some kind of system evaluation under educational

viewpoint? We use a “*evaluation of pedagogical aspects*” developed with teachers that will use our system in real school environment (classroom). We created an instrument (questionnaire), under Psychologist and Education supervision, to identify what system aspects are adequate or suitable to be explored by a special methodology during learning activities. They have a set of questions that help us to identify the teacher’s opinion. It can be adapted for specific situations by cutting some questions or rewriting systems names.

III. Some Environments that exemplify our methodology result

We focus our effort to build educational systems to work as solving problem environments. We choose to model educational software inspired on ITS based on multi-agent system architecture (MAS) and game like fashion interfaces. From ITS architecture we explore the idea of student model to personalise the interaction between systems and user. From games and entertainment applications we explore the interfaces. From MAS approach we explore the domain modelling to create better conditions to be explored by student

This section presents some tested software developed in our group to support learning activities for children in elementary schools. They were chosen because we apply on their design and implementation our methodology as a whole. They allow us to evaluate our methodology and better understand the complexity intrinsically in this activity (developing educational software respecting technical and pedagogical aspects).

- **Eco-Lógico** [<http://www.inf.pucrs.br/~giraffa/eco-logico/>]

The Eco-Lógico is an educational game developing using Toolbook Multimedia⁸. The domain application is Ecology (water pollution). In this ecological environment the student chooses a character to play using six options: Mother Nature, Mayor, Citizen, Fish, Ecologist, and Tourist. After that, student defines the game configuration (foreign elements that will cause pollution) by a selecting process from a graphical menu. The student’s first see a lake in equilibrium (represented through a background bitmap) where there are different types of fish swimming. After 30 seconds the chosen polluter’s appears which soon begins to show their action (take out energy from scenery elements).

There is a visual gauge (the Ecometer) that helps the user to observe the conditions of the lake. A well-balanced situation is shown when the gauge is full and green. As the equilibrium is lost, the colour changes gradually to yellow, finally changing to red when a dangerous situation is reached. The number of fish swimming in the lake indicates the situation. The student’s challenge is to maintain the equilibrium and fight against the action of these foreign elements using a set composed by four tools. Each character has its own tools with different power to fight against the pollution (put energy on scenery elements). The period of time is five minutes.

The game was modelling to be played by one student. There are sounds effects during the animations, and some information about the scenery elements can be obtained pressing mouse right button. At the end student receives a message explain if s/he achieve the goal or not. Figure 1 presents system interface:

The system was tested in real situation, i.e., and students from an elementary school worked with environment as a complementary activity during their classes. The

⁸ Toolbook Multimedia is a trade mark from Assimatrix Inc.

results allow us to measure interface limitations and possibilities. It has been used for several schools in our city, and it is sprayed out in Brazil as well. All information and software can be download at URL above.

- **MCOE** [<http://www.inf.pucrs.br/~giraffa/mcoe/mcoe.html>]

The conception of the ‘MCOE’ was based on an ITS architecture proposed by Giraffa, Viccari, Self, 1998) and ended by Giraffa [GIR99] This architecture is composed of a hybrid society of agents that work to achieve a common goal: to fight against the pollution resulting from foreign elements (pollutants) and to maintain the system equilibrium. We have *reactive agents* (bottom of the lake, micro-organisms - plankton, water, plants and three types of fish) and *cognitive agents* (the Tutoring agent, and students represented by characters).

The *cognitive agents* are modelled using a *mentalist approach*, where the term agent means a computer system that can be viewed as consisting of mental states such as beliefs, intentions, motives, expectations, obligations and so on. In this case the question of which is an agent is replaced by the question of which entities can be viewed as possessing mental states. The model of each student and the tutor contains a set of basic beliefs, desires and expectations⁹. From this set emerges the dynamic selection done by Tutor to select a personal teaching strategy.

We designed the system to be played by two students. The first student chooses a character to play using four options: Mother Nature, Mayor, Citizen, and Tourist. After that, the second student can choose one of the three remaining characters. The system defines the game configuration (foreign elements that will cause pollution) by a sorting process using a random function. Their challenge is to maintain the equilibrium and fight the action of these foreign elements for a period of time previous defined by the teacher or the students.

The three types of fish, micro-organisms, and plants compose the reactive agents. So, when the game begins the student see a lake with the select scenery elements (sorting by the system trough a random function) which soon begins to show the action of the polluters. The number of plants, micro-organisms and fish swimming in the lake indicate the situation. There is a visual gauge (the Ecometer) that helps the user to observe the conditions of the lake. A well-balanced situation is shown when the gauge is full and green. As the equilibrium is lost, the colour changes gradually to yellow, finally changing to red when a dangerous situation is reached.

The system equilibrium is based on the energy level; the occasional predatory actions that can happen in the environment and the number of elements that remain in the system after the pollutant impact. The fish, plankton, and plants also reproduce using a function based on the energy level. However, when they reproduce they naturally loose energy, because they give energy to their descendants. If an agent dies (zero energy), it is removed from the system.

There is a help system, which allows the student, solve some doubts about the game rules, the domain, and a small dictionary (because we use some technical words to explain the polluters).

The environment was tested and compared with Eco-Lógico and the results show us that using agents improve users interaction and enlarge system possibilities under pedagogical aspects.

- **Tirinha** [<http://www.inf.pucrs.br/~giraffa/tirinha/>]

⁹ The expectation is considered for us as a future belief.

The Tirinha environment has a set of tool that makes it possible to create comic strips in a cooperative way. This set consists of a chat tool, a whiteboard to visualise the chosen images (pictures or drawings made by the student) and the current version of each picture that compose the comic strip. The images are stored in a shared database and are accessed through an editing window.

The environment architecture integrates a set of tools that makes it possible to augment the teaching-learning activities used in the conventional classroom. We have been using “conventional classroom” as synonymous of presential classes with the traditional approach. The “virtual classrooms” are meetings supported by chat tools, and the cyberspace available on the Web. We have been using the virtual classroom as an extension or as a substitute of the conventional classroom.

The central element is an interface that integrates resources, where the students co-operatively construct comic strips. The teacher can define or not the composition of the groups. The subject of the stories is related with the lessons being given. Each group member interacts with the others through a chat tool. Each student has his/her own database of pictures to be shared with other group members. They can choose the scenario and characters of story. In the scheme represented in Figure 1, DB represents the database, while S represents the students. The environment supports local or distributed databases.

Tirinha has several activities that can be accomplished individually (activities like creating student’s database of pictures) and others that can be performed co-operatively (like making decisions about the comic strip’s pictures, texts, and so on). It is being developed in Java and the interface has the following resources:

- Chat Tool;
- Access to shared and distributed databases;
- In the database management each student has his/her own database that can be shared among the workgroup and with images ready to use.
- Window for comic strips’ picture edition;
- Group members’ visualisation ;
- Visualisation of the available comics in the shared databases;
- Whiteboard for cooperative pictures edition.

This tool has been developed under the context of the Global Campus research group – PUCRS. These group co-ordinates the Virtual University activities in our campus [2]. The work resulting from student’s activities is a set of HTML pages. Students can also create presentations, for example, in the Microsoft Power Point tool pattern.

The Tirinha environment integrates, in the same interface, the whiteboard and chat tool to permit the cooperative comic’s construction. The Figure 2 shows the Tirinha interface with four main regions:

- (1) comic construction area;
- (2) area that shows the students interaction through the chat tool (example in Portuguese);
- (3) area when the students type the interaction to send to other students and
- (4) open picture button to select and add images in the whiteboard screen.

This environment is supported by TCP/IP protocol because is used in local area networks and Internet.

Besides the computational environment support, Tirinha has a methodological support to help the teaching activity. A homepage was built to assist teachers in their classes. In this homepage is available some suggested methodologies to use the Tirinha computational tool. Also, you can find system documentation, correlated works, and prototype on the URL above.

The Internet and associated technologies brought something really good for educational environments: it enabled us to do much more than just transport traditional methodologies to virtual classes.

The need to create new environments and methodologies that allow teachers to explore different possibilities to improve the educational process is something very exciting, and the big challenge for educational researches.

We have tested the different tools that were developed (chat, whiteboard, and database). These tests' results show us the potential of the environment. To better achieve a good design for it, a specialist in comic's strip is supervising our prototypes, and helping us to test the available possibilities with the children. The design, the implicit methodology, and all the aspects that are found in our environment were possible because we work in an interdisciplinary team.

- *TCHê- a trip around Rio Grande do Sul (RS)*
<http://www.inf.pucrs.br/~giraffa/tche/>

This educational software (game interface) has the purpose to support students from elementary schools (first grade) to understand how to solve basic Mathematics problems.

The game was designed to present a set of activities that can be performed during a configurable time. The time adjust is an important approach adopted since MCOE system [10] that guarantees flexible tuning according different students profiles. The set of activities is divided in four scenarios. The first screen introduces the map of the state of RS with the indication of the areas that we can find the game activities. The chosen areas were Country side (the student must identify cattle and sheep herd, count them and put into a fence in), Mountains (the students have to produce a certain quantity of wine), seashore (select fish and count them identifying types and weight), and the capital and surroundings (connected washes with their function related to the building where it on)). We used agent techniques to model and implement the scenario elements (reactive agents). It allows the student to explore the possibilities of analyse the result of his/her choices.

All these systems were design using an interdisciplinary team composed by: a set of specialist on the domain area (teachers from elementary and high schools), interfaces designers that help us to solve many problems connected to interface overhead information, computer graphics programmers, networks specialist, psychologists, and educational software designer (senior and beginners).

The approach suggested on section two was followed in a rigorous way. We had to develop specific instruments to elicit the information we need (domain model) and teaching strategies related to student's mistakes. The instruments were different questionnaires structured according our specific goals in each project phase. Each software delay 18 months in average to be modelled, implemented (prototype), and testing. All are available to be downloaded for Portuguese community . We create

homepages to describe all the systems and offers the opportunity for Education research community to check our results and collaborate with us through their critical analysis. All the system was sprayed out in different schools (in different Brazilian states).

V. final considerations

Developing educational environments is a complex task. We must guarantee software quality from both aspects: Computer Science and Educational view point. It is a task not easy to do. Especially because we need an interdisciplinary team. Work with people from different areas, with different backgrounds is quite hard. They have natural incompatibilities related with their original areas to overcome this restrictions we need to integrate all activities in such a way that guarantees team interaction.

If all team members have their responsibilities clarified and remarked good results can be achieved. . Our experiences showed us that make educational software can be done if the specialist (domain area) participate in every steps of software design and evaluation. When we start to model a system we create some expectatives in our specialist that can be or not fulfilled by the final version (generally, the prototype). It is important to perform some kind of evaluation during the role process with s/he in order to have a continuo feedback.

We have been spending more than ten years working with educational software design and their implementation. We reached interesting results, and the more significance were researched with Eco-Logico[15], MCOE[8;9;10], Tirinha[2], and TCHE [14]. They were modelling using our set of beliefs about the role of educational environments in learning process. We do believe that a good educational software must allow teachers and students to explore hypotheses about domain, skills, and personal capabilities.

We also found some common interesting aspects concerning teacher's behaviour. They, in general, do not have interest to full fill instruments. When you talk with them they show enthusiasm in participate. However, when they discover that they need to work more than usual, they give up. This work is connected to maintain their ordinary schedule plus allow time for project activities. This small number of participants needs further reflection from Education/Psychologist specialists. The reasons of teacher's behaviour need much more reflection, because it seems to us a typical profile of teachers from elementary school (not exclusive). We developed in our group another researches with the same users, and we found the same profile. Of course we can not generalise this for all teachers. We can find many teachers involved with good educational projects in different school levels. However, we notice that their profile is different from the ordinary teachers. They showed some special characteristics:

- They did not mind to work on new tasks. Even if it means some degree of "overworking";
- They area always trying to find different options to improve their teaching performance;
- They are not afraid to use technological innovations. Again, even if it means to learn/to aquire new skills or abilities.

These teachers have also a special profile concerning to educational software

selection:

- they like to use educational software as a complementary activity in their classrooms;
- they prefer software written in their native language (in our case, Portuguese);
- they like to select games instead of other educational software modalities;
- they prefer software with sounds;
- they prefer software with animation;
- they prefer software that allow interaction among children;
- they prefer software that is possible to save the children production (work);
- teachers have shown interest in using educational environment in their classes mixed with traditional resources or methods.

We have observed that private schools normally use educational software in classrooms and public schools still have some problems to adopt computer as a tool in their classes. We identify reasons linked with teacher's profile and physical restrictions. Other important aspect is that educational environment can be enriched if students are not space limited and students can use Internet to extend their interaction with teacher and colleagues.

Acknowledges

I say thanks to my students and colleagues from Faculty of Informatics of PUCRS that collaborate with us in different ways during the last ten years. Special thanks to Andre Raabe, one of the authors of Eco-Logico for his ideas, perseverance, creativity, and good implementation during Eco-Logico project, and MCOE evaluation; Leandro Bernsmüller for the implementation of MCOE system, and Elisa Boff, author of Tirinha, for her responsibility and high quality of work. Angela Cristina Mazzorani, Luciana de Araujo Spagnoli, and Sabrina dos Santos Marczak, team members of TCHE project. Good partners made a good team.

References

- [1] Alves,I.;Kolberg,L.;Giraffa,L.M.M.;Polonia, E. HEI:assistente inteligente para o suporte ao ensino de Lingua Inglesa.VIII SBIE:Simposio Brasileiro de Informatica na Educacao,8,1997.Anais...Sao Carlos:ITA,1997
- [2] Boff, E.; Giraffa, L. M. M. A cooperative environment on the Web to support teaching-learning activities. In: ICECE 2000 (International Conference on Engineering and Computer Education), 2000, São Paulo, SP. **Proceedings...** São Paulo, 2000.. (CD-ROM)
- [3] Giraffa,L.M.M. Reflexões sobre o computador na escola. Porto Alegre: FAGED/PUCRS, 1991 (Mater Dissertation)

- [4] Giraffa,L.M.M.; Souza,L;;Garcia,L. F. GUTEMBERG- ferramenta de hipermedia para fins educacionais. VI SBIE - Simpósio Brasileiro de informática na Educação, 1995.**Anais ...** Florianópolis: UFSC, 1995
- [5] Giraffa, Lucia M. M.; Oliveira, Marco Andrei. LEONARDO - máquina de auxílio à resolução de problemas. VI SBIE - Simpósio Brasileiro de informática na Educação, 1995.**Anais ...** Florianópolis: UFSC, 1995
- [6] Giraffa, L.M.M; Viccari, R.M.; Self, J. Multi-Agent based pedagogical games. In ITS'98- Fourth International Conference on Intelligent Tutoring Systems, 4., 1998. **Proceedings...** New York: Springer Verlag, 1998.
- [7] Giraffa,L.M.M;Oliveira, J.; Borges,L. Túnel do Tempo - Ferramenta de auxílio ao ensino de História. III Congresso Ibero-Americano de Informática na Educação, 3., 1996. **Anais...**Barranquilla: RIBIE,1996.
- [8] Giraffa, L.M.M; Viccari, R.M.; Self, J. Multi-Agent based pedagogical games. ITS'98- Fourth International Conference on Intelligent Tutoring Systems. **Proceedings...** San Antonio, Texas,1998.
- [9] Giraffa, L.M.M; Mora, M.; Viccari, R.M. Modelling an interactive ITS using a MAS approach: from design to pedagogical evaluation. 3rd International Conference on Computational Intelligence and Multimedia Applications - ICCIMA'99. **Proceedings..** New Delhi, India, September 1999.
- [10] Giraffa, L.M.M. **Uma arquitetura de tutor utilizando estados mentais.** Porto Alegre: CPGCC/UFRGS, 1999. (PhD thesis)
- [11] Kohlmann, C; Blotta, G; Silva, M.C. **PINGUIM**: processo inteligente guiado à multiagentes. FACIN: PUCRS, 1998.
- [12]Luzzi,F;Ferreira,R.;Senna,R.;Giraffa,L.M.M.;Bastos,R.M. **WOHLER**:assitente inteligente para suporte ao ensino de Química.VIII SBIE:Simposio Brasileiro de Informatica na Educacao,8,1997.**Anais...**Sao Carlos:ITA,1997.
- [13] Mazzorani, A; Spagnoli, L.R; Marczak, S; Giraffa, L.M.M. Desenvolvimento de jogos educacionais com abordagem interdisciplinar. WIE 2000 - Workshop de Informática na Escola - Congresso da SBC 2001. **Anais ...**Fortaleza: UFCE, 2001. (CD-ROM)
- [14] NIELSEN, J. **Heuristic Evaluation.** Disponível em <http://www.useit.com/papers/heuristic>
- [15] Ortolan, A; Barrinuevo, L; Giraffa, L.M.M. ABRACADABRA Abracadabra-Ambiente de ensino-aprendizagem computadorizado.IV SBIE - Simpósio Brasileiro de Informática na Educação, 4, 1993. **Anais ...**Recife; UFPE,1993.
- [16] Raabe, A.L.A; Javimczik, A .M.; Giraffa,L.M.M. Eco-Lógico : Ambiente interativo para suporte ao ensino de educação ambiental .VII Simpósio

Brasileiro de Informática na Educação, 7., 1996. **Anais** Belo Horizonte: DCC/UFMG, 1996.

- [17] RAABE, André L. A.; GIRAFFA, Lúcia M. M.; Ambiente para Construção de Materiais Instrucionais Apoiados por Vídeo. Revista Brasileira de Informática na Educação. Florianópolis, SBC, (9), Abr.2001.