World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:4, No:3, 2010

Software Model for a Computer Based Training for an HVDC Control Desk Simulator

José R. G. Braga, Joice B. Mendes, Guilherme H. Caponetto, Alexandre C. B. Ramos

***Abstract*—**With major technological advances and to reduce the cost of training apprentices for real-time critical systems, it was necessary the development of Intelligent Tutoring Systems for training apprentices in these systems. These systems, in general, have interactive features so that the learning is actually more efficient, making the learner more familiar with the mechanism in question. In the home stage of learning, tests are performed to obtain the student's income, a measure on their use. The aim of this paper is to present a framework to model an Intelligent Tutoring Systems using the UML language. The various steps of the analysis are considered the diagrams required to build a general model, whose purpose is to present the different perspectives of its development.

***Keywords*—**Computer based training, Hypermedia, Software modeling.

I. Introduction

THE intelligent tutoring systems are software that support the learning, covering program with some intelligence and can be used for learning. Its instructional systems are based on computer models of instructional content and present new ways for education [1].

The classical architecture of intelligent tutoring systems is composed of four elements that are related: the Domain Model, Student Model, Tutor Model and Interface Model. The Domain Model handles all content to be taught by the system, providing mechanisms for example generation. Mechanisms of Artificial Intelligence are used, so that knowledge must be consistent with the reasoning of the student and modeled according to a taxonomy.

The Student Model presents aspects of behavior and knowledge of the student and should be able to detect errors made by the user and verify changes in their profile. There are basically three models of relationships between models and the field of student: modeling by overlap, differential modeling and disturbance modeling.

The overlap modeling is a very simple technique, where the student is aware of a subset of the domain. The differential modeling is an extension of the overlapping, which divides knowledge into two: that the student should know and not to be expected that he knows. In this approach the student's model is very restricted.

The disturbance modeling introduces an improvement on the other. Considers the student's knowledge goes beyond the boundary of the domain model, including errors and false conceptions of the student.

The Tutor Model is pedagogic knowledge of the system, consisting of a system of rules, which selects the content to be displayed and monitors the use of the student, and provides assistance when necessary. Tutor exist in the model of education and training strategies, Socratic, mentoring, cooperative others. In addition there are strategies for dialogues which serve to help to motivate the student.

Finally, the Interface Model is the model that interacts directly with the student, according to the model field of student and tutor proposed and plays a crucial role in the system. In this model, several computer items to be considered, such as response time, the look to be attractive, clear presentation and easy to use.

All these models are defined according the need of the company or entity to use the software, and vary depending on the subject being taught. These computer-assisted systems are being increasingly employed in education in various branches of activities, both commercial and teaching, as significantly reduce the cost, in addition to increase income of the students, because coupling audiovisual resources, which make the teaching more attractive, interactive and interesting.

Thus, detecting the need for tools for intelligent tutoring systems, special tools have been developed as a research and development agreement, by the Group of Applied Computer Vision - VisCAp, of Institute of Exact Sciences, from Federal University of Itajubá.

II. Objective

This article aims to do the analysis and modeling software for Intelligent Tutoring Systems – ITS, using the UML language. This analysis should take into consideration, and in different stages, the diagrams required to construct a model of the system, which are intended to present its different perspectives, starting with the use case diagrams, passing through the activities diagrams, state and sequence, and finally the class diagram.

The ITS framework developed to an Helicopter Pilot Training System to the AS350-X helicopter from HELIBRAS, was used for develop the Operator Training System at Furnas – Centrais Elétricas S.A. with a few modifications.

J. R. G. Braga is with Universidade Federal de Itajubá, Itajubá MG 37500-000 Brazil (corresponding author to provide phone: +55 35 36291428; fax: +55 35 36291140; e-mail: jgarciabraga@yahoo.com.br).

A. C. B. Ramos is with Universidade Federal de Itajubá, Itajubá MG 37500-000 Brazil (corresponding author to provide phone: +55 35 36291428; fax: +55 35 36291140; e-mail: ramos@unifei.edu.br).

III. Methodology

According to recent research literature and, despite being a universal concept, the adoption of computer-based systems for training is still often inadequate and inappropriate. However, there are so many works in this field [2], directed to the training of the various sciences of knowledge. One of the most recent is the work of some students from the University of Fortaleza, Brazil, in conjunction with Imperial College London in England, who developed and implemented a system for interactive simulation of orthodontic treatment based on computer [3].

The Intelligent Tutoring Systems - ITS have been designed under the basic guidelines to provide students with equipment for training based on computers, advanced, allowing the practice of the learning activities effectively, and enabling the students to carry out their activities in real time, this through familiarity with the various components of the problem, considering normal conditions and emergencies. An ITS serves as a tool for support and guidance to students in the process of training and is related to the occurrence of events that include drawings, graphics and animation when necessary. The subjects dealt with the tutorials are in accordance with the manuals provided by the company, and address various issues. Figure 1 presents the use cases, used to explain the interactions of the student and instructor with the training, including evaluations and simulations.

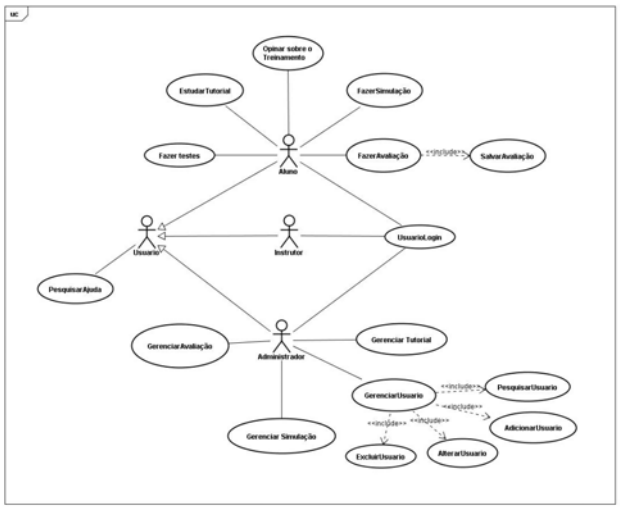

Fig. 1 Use case Diagram of the ITS Modeling

Unified Modeling Language - UML, was used to model the system because it is a language completely extensible, adaptable and easy to understand, which uses a graphical notation with a widespread pattern and with different levels of abstraction [4]. The tool used to make the modeling is JUDE (Java and UML Developers Environment) an easy to use and intuitive tool that can be used to made complex models [5].

The process for developing the ITS is complex and is formed by a series of activities for various elements [6], the basic concepts is:

- The planning and specification of requirements, which were detected in cases of use, enabling the achievement of the requirements of the system and creation of scenarios and
- Construction, which includes two phases: the analysis of the problem and design. In the analysis, identify concepts, associations and attributes, and with them, the diagrams are constructed of activities, sequence and state of the system. They define the dynamic behavior of the system. Already in the design, it produces the diagram of classes, which defines the static structure of the system.

IV. Results

A. Nouns involved in modeling

Been the principal actor "User", who can be:

• System Administrator: As the person responsible for managing the operations relating to: registration of users, integration of tutorials, simulations and assessments in the system. Take care of the correct functioning of the system and its connection to the database.

• Student: As a direct user of the system of tutorials. The student can do their training with the help of tutorials, but also do the simulations and evaluations to assess their learning.

• Instructor: Also as a user of the system, but its most important function is to monitor the performance of students in the use of tutorials.

TABLE I
Objects and Attributes Table

Objects	Objects
User	Student
Code of user	Code of Student
Type of user: student, instructor or administrator	ID
ID	Password
Password	Notes
Instructor	Administrator
Code of Instructor	Code of Administrator
ID	ID
Password	Pasword
Tutorial	Simulação
Code of Tutorial	Code of Simulation
Name of Tutorial	Name of Simulation
Content	Content
Avaliação	
Code of Evaluation	
Name of Evaluation	
Content	

All objects and attributes of the system can be extracted by the systems' description, Table 1 shows some objects and attributes extracted using this method.

B. Use cases

The use cases diagram specifies the behavior if the software tutorial from a description of a set of sequences of actions [7]. Fig. 2 illustrates the general diagram of the cases of use. There

is the interaction of each actor with the processes involved in software.

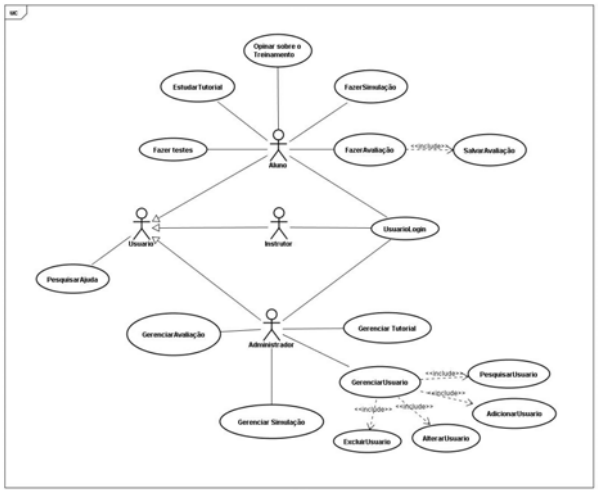

Fig. 2 General diagram of use cases

Each use case can be expanded in a diagram, for example a diagram that includes variations performed by the software tutorial to produce a result value of the observable actor "user." The result is applied to capture the desired behavior of the software without the need to specify how this behavior is implemented. Fig. 3 shows this behavior.

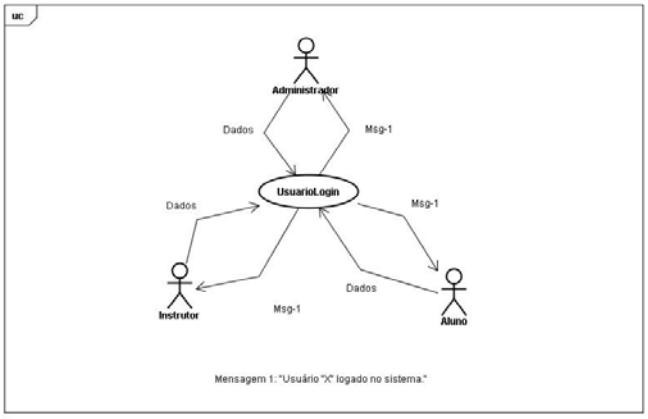

Fig. 3 Use Case "UserLogin"

C. Activity diagram

This diagram was made to the modeling of dynamic aspects of software tutorial, which involves the modeling of sequential steps and / or competitors in each of the subsystems. In the activity diagram shown in Figure 4 is made to model the flow of the system, as it passes from one state to another at different points of the flow of control.

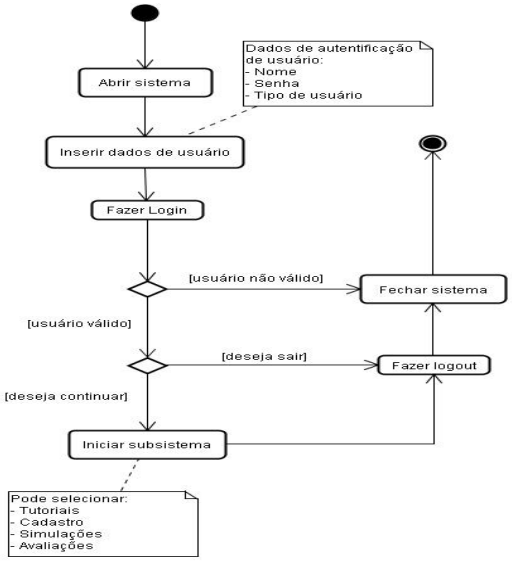

Fig. 4 Activity diagram

D. Sequence diagram

In sequence diagrams of models for the software tutorial, is for the interactions that emphasize the structural organization of the various objects involved in it. Figure 5 is shows the sequence diagram for the registration and entry of a user in the system.

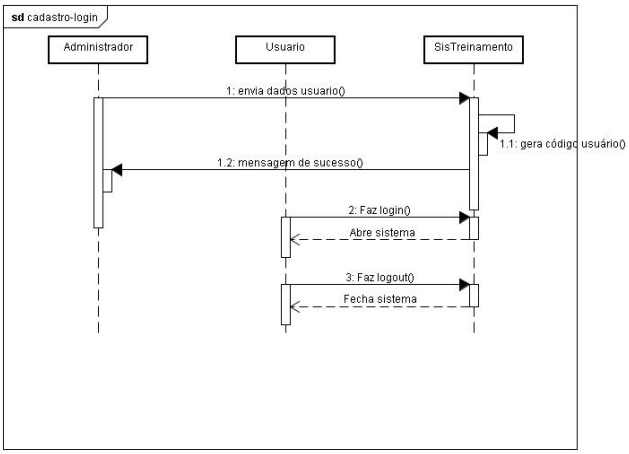

Fig. 5 Sequence diagram "Register and Login"

Fig. 6 represents the interaction of a user "Student" with the tutorials.

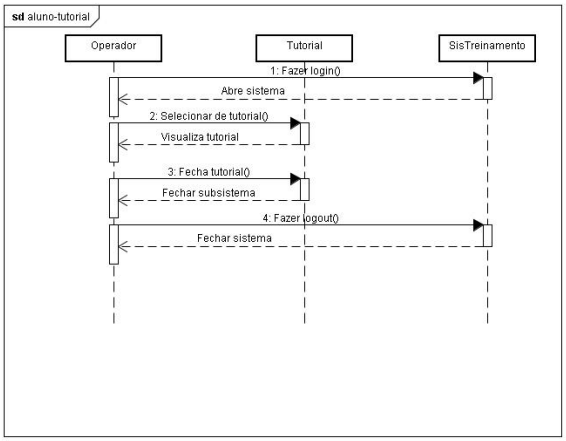

Fig. 6 Sequence diagram "Student Interaction"

A state diagram involves the specification of lifetime of class instances of a use case or an entire system. In the case of software tutorial, this work presents the diagram of state for the case of use "UserLogin" (Fig. 7).

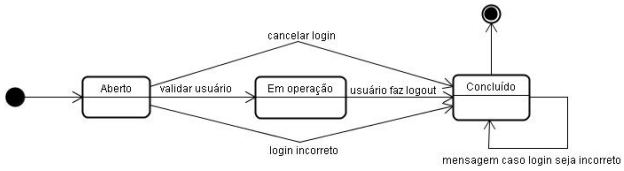

Fig. 7 State diagram "User-login"

Another important state diagram is the one that represents the actions: add / modify / search / delete an object of the system (Fig. 8).

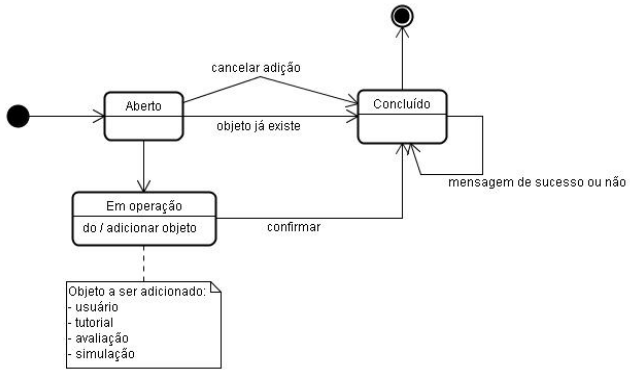

Fig. 8 State diagram "Add object"

E. Class diagram

The class diagram modeled in this paper gives a static view of the software tutorial. The training system, the main class has four classes depending on it (for which you create a relationship of dependency): "Tutorial", "Register", "assessment" and "Simulation". For Class "Tutorial", have the relationship of generalization with the types of tutorials, which means that each subclass is a kind of tutorial class, which is shown in Fig. 9.

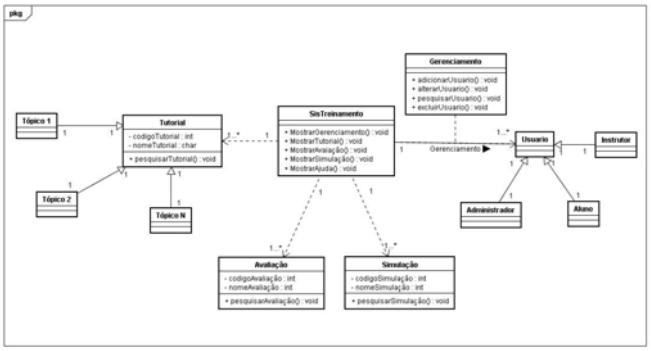

Fig. 9 Class diagram

F. Graphical interface

As already mentioned, the intelligent tutor system is composed of the Model Interface, which defines the structure of interaction with the user. The defined interface is characteristic of each system in particular. Below is the interface of the intelligent tutor which was developed at the University for Furnas. The system was built using the Java language and its interface.

Fig. 10 Main screen to enter the system

Fig. 10 represents the login screen of the system developed for intelligent tutor Furnas.

Fig. 11 represents the screen of selection of the subsystem, where the user can choose "Tutorial", "Register", "simulations" or "Reviews", depending on their function in the software. If you select "Tutorials", you can access any of the tutorials to be added to the system.

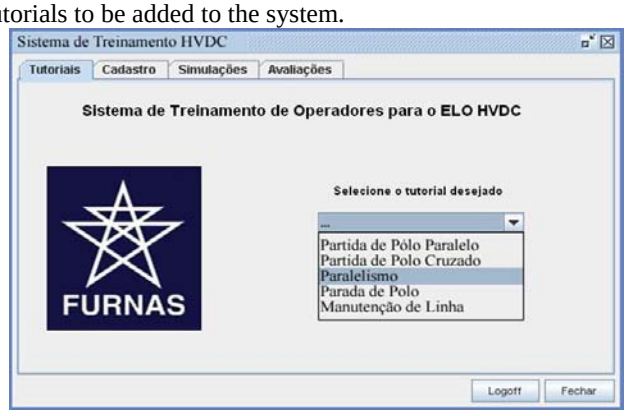

Fig. 11 Selecting tutorial

Fig. 12 shows the table of control built on the premises of VisCap in UNIFEI, to simulate the control of the operator of Furnas, providing high level training to operators of Sub-Station of San Roque and the Station of Foz do Iguaçu, this training which currently is done directly on the table or form of control theory.

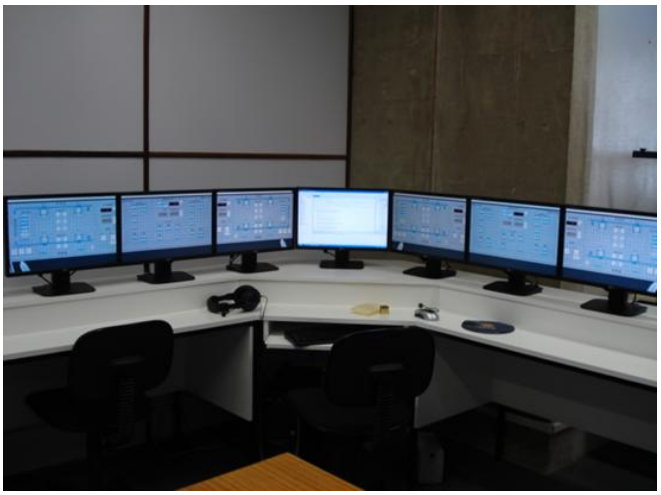

Fig. 12 Virtual control table from HVDC- Furnas

V. Conclusions

This article presented the analysis and modeling of Intelligent Tutoring Systems using UML programming language, using the tool JUDE (Java and UML Developer Environment). The methodology adopted in developing the software tutorial was based on the modeling of the system of training, and was presented as an example the case of software developed for the company FURNAS, taking into account the planning and specification of requirements and analysis and design of the system. We considered the diagrams to build the model of the system, both with the interaction of the actors found in the analysis, as to their responsibilities and duties.

References

[1] Gamboa, Hugo and Fred, Ana, 2001. *Designing* Intelligent Tutoring System: A Bayesian Approach. *Proceeding of International Conference on Enterprise Information System*, Setubal, Portugal.

[2] Kassem, Saleh. Documenting electronic commerce systems and software using the unified modeling language. *Proceeding of Information and Software Technology*. United Arab Emirates.

[3] Rodrigues, Maria *et a, 2007. An interactive simulation system for training and treatment planning in orthodontics. Proceeding of Computers & Graphics, Elsevier.* Fortaleza, Brasil.

[4] Arlow, Jim and Neustadt, Ila, 2005. *UML 2 and the unified process: practical object-oriented analysis and design.* Addison Wesley.

[5] *JUDE – Java and UML Developers' Environment*. Disponível em: <http://br-linux.org/linux/node/3335/>. Access: september, 8 2007.

[6] Rumbaugh, J; Blaha and M; Premerlani, W, 1994. *Modelagem e projetos baseados em objetos*. Campus, Rio de Janeiro, Brasil.

José R. G. Braga was born in Pouso Alegre, Brazil, on Febuary 3, 1983. He received the B.S. degree in Computer Science from Universidade Federal de Itajubá, Itajubá, Brasil in 2008 and the M.S. degree Computer Systems and Techonology from Universidade Federal de Itajubá, Itajubá, in 2009.

Since 2007, Mr. Braga has been working for Furnas in the HVDC-SIM project. His current interests are in intelligent user interface design for eletrical and aeronautical systems.

Joice B. Mendes was born in Lorena, Brazil, on April 21, 1986. She received the B.S. degree in Computer Science from Universidade Federal de Itajubá, Itajubá, Brasil in 2009.

Since 2007, Ms. Mendes has been working for Furnas in the HVDC-SIM project. His current interests are in intelligent user interface design for eletrical and aeronautical systems.

Guilherme H. Caponetto was born in Amparo, Brazil, on April 17, 1988. He will receive the B.S. degree in Computer Science from Universidade Federal de Itajubá, Itajubá, Brasil in 2010.

Since 2007, Mr. Caponetto has been working for Furnas in the HVDC-SIM project. His current interests are in intelligent user interface design for eletrical and aeronautical systems.

Alexandre C. B. Ramos (S'92) was born in Niterói, Brazil, on April 18, 1959. He received the B.S. degree in electril engineering from Faculdade de Engenharia de São José dos Campos, são José dos Campos, Brasil in 1985 and the M.S. and PhD. degrees in Electrical Engineering and Computer Science from Instituto Tecnológico de Aeronáutica, são José dos Campos, in 1992 and 1996, respectively.

Since 2003, Mr. Ramos has been working at Universidade Federal de Itajuba and for Helibras and Furnas in the HVDC-SIM projects. His current interests are in intelligent user interface design for eletrical and aeronautical systems.