A distributed architecture of intelligent tutoring systems in distance education context

Arturo Hernández-Domínguez
ahd@vcnet.com.br or arturo@dsc.ufpb.br
Universidade Federal de Alagoas - UFAL, Brasil
Departamento de Tecnologia da Informação - TCI

Aleksandra do Socorro da Silva
alekas@brhs.com.br
Universidade da Amazônia – UNAMA, Brasil

Abstract: This paper describes a virtual class, which is based on the principles of ITS (Intelligent Tutoring Systems). The virtual class allows to a group of learners to participate in training sessions of an adapted teleteaching system (adapted virtual class), this system takes into account progression rhythm different inside a community of remote learners. The architecture of virtual class is distributed by several training sites. The virtual class allows adapting the teaching system in a flexible, individual, and collective way. This system allows also the implementation of a pedagogical method: cooperative learning. Object-oriented and agent-oriented modeling are being used to specify and implement an architecture of the adaptable virtual class. The implementation of virtual class is being developed in distance education context using the Internet and Java.

Keywords: Intelligent tutoring system, distance education, agents, and object-oriented development

I Introduction

The principle of the architecture proposed for an adapted training service is to allow the adaptation of knowledge transmission from a teaching function managing a virtual group of learners by a communication system (figure 1). This architecture takes into account the principles of ITS (Intelligent Tutoring Systems) [Wenger 87], [Canut 90], [Haton 92]. Teaching functions are distributed taking into account the participation of teachers, system, resources and learners in some cases (when a learner has the knowledge or experience necessary for playing the teacher's role).

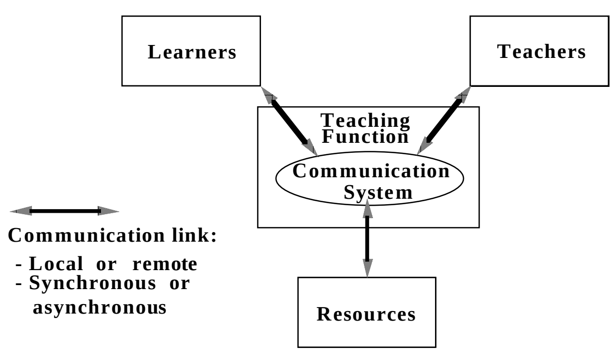


Figure 1: General architecture for each adapted training service.

In general the teachers control the teaching module, in the same way as traditional classrooms. In this paper, we propose for each adapted training service to distribute the teaching functionality with the

participation of teachers, learners and resources in a virtual class system. We take into account two kind of sites in a multi-site environment for distance education (figure 2).
a) The central site, which provides a set of reusable SITB (domain and didactic building blocks), uses a server.
b) The decentralized sites, which allow to learners to participate in an adapted training class.

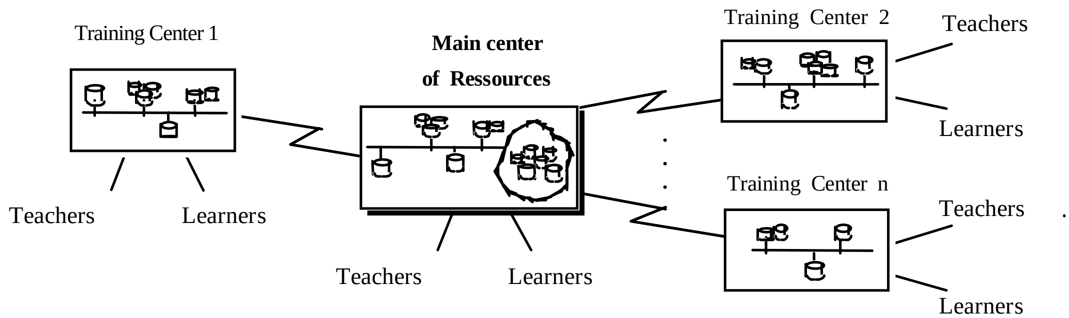
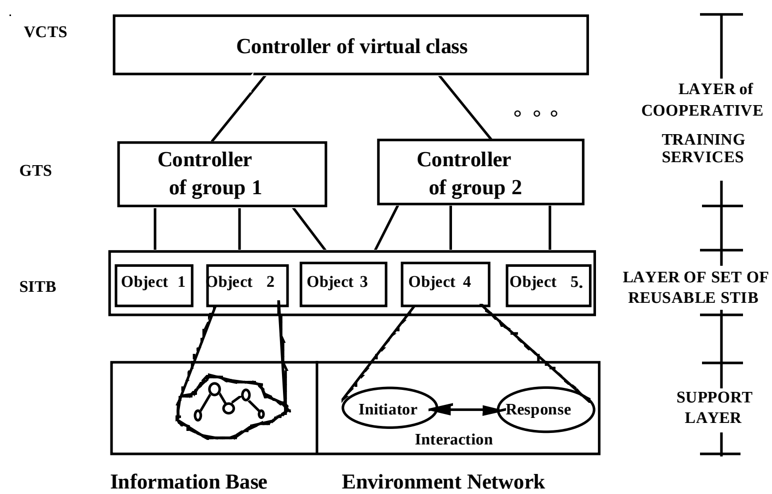


Figure 2: A multi-site environment for distance education.

II General architecture

In the multi-site environment of teleteaching, we propose that any training site must be represented according to the ARESFED (In French: "Architecture pour la Reutilisation et l'Exploitation des Services de Formation dans le contexte de l'Education à Distance") [Hernández 95]. In this way the ARESFED is used as an internal standard in the multi-site environment of teleteaching. This architecture is composed of three layers: support layer, layer of a set of reusable SITB (Service Independent Training Building Block) and a cooperative training services layer [Hernández 95] (figure 3).



SITB: Service Independent Training Building Block
GTS: Group Training Service
VCTS: Virtual Class Training Service

Figure 3: Training architecture of a virtual class.

Thus, the representation of each training site is simplified and standardized, only one architecture we must specify: the ARESFED (in Portuguese it is called ACVA: "**A**rquitetura de uma **C**lasse ou Aula **V**irtual **A**daptativa").

II.1 Support Layer

It represents a set of frameworks from which the training architecture is built:
Information Space provides information management services relevant to the courseware developed (for example: a database management system stocks all the reusable objects),
Communication Space supplies services supporting a network and communication service.

II.2 Layer of a set of Reusable SITB

This layer is represented by a set of functional, independent, and reusable components. These components will be shared and reused for creating a new service. We have identified with a participation of educationalists two kinds of resources: didactic (teaching activities) and information resources ("bricks of domain")[Canut 94], [Vincent 94]. From reusing of these types of resources a training service can be specified as a dynamic training service. The criteria of reusing of didactic and information resources are determined by learner's educational needs during a training session. The dynamic training service must diagnose an adequate resource that responds to learner's educational needs. In this way a dynamic training service is considered as an adaptable service.

II.3 Cooperative Service Layer

This abstraction level represents the two kind of cooperative training services provided by this architecture [Hernández 95]:

a) Global level represents a virtual class composed of a set of remote learner groups,
b) Local level represents a particular learner group; a knowledge level classifies each remote learner group.

These two classes of cooperative services must be adaptable to learner's behavior in a training session.
The composition criterion of virtual class takes into account a pedagogical context; this pedagogical context is based on the cooperative learning [Doyon 91]. A virtual class will be composed of a set of heterogeneous groups, each group allows to a set of remote learners to participate to a training session. In a traditional class the homogeneity of knowledge level is not always assured. Some cases, this homogeneity of knowledge level becomes very superficial. For solving this problem, we consider a first level of adaptation to learner's educational needs; this level corresponds to take into account a virtual class as a composition of heterogeneous groups. These groups have a variable composition. Each heterogeneous group corresponds to a different level of knowledge allowing characterizing a group of learners with the same level of knowledge about a particular domain. However, inside a group of learners is possible to find different behaviors of learners according to progression rhythm of each learner. The second level of adaptation is taken into account into a group of learners. A particular

knowledge level associated to a group of learners may also become superficial and not adapted. For preventing this problem, we propose to consider the behavior zones (figure 4) into a group of learners [Hernández 95]. In this case, it is necessary to allow and manage the change of behavior zone of a learner inside a group (*intra-group change*) for assuring an adapted knowledge level of learners inside a particular group (local view). Thus, for assuring an adapted virtual class (global view) is also necessary to allow and manage the change of group (*inter-group change*).

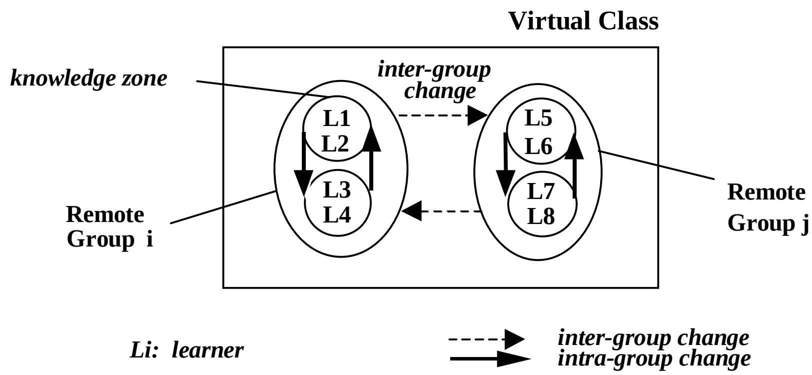


Figure 4: The distribution of learners in the virtual class.

II.3.1 The Adapted Training Service of a Virtual Class Architecture

The adapted training service of a virtual class has as principal functions:

a) The managing of progression of a set of remote learners,
b) The controller of the VCTS (Virtual Class Training Service) takes into account a pedagogical, structural and adaptable control of a virtual class.

An initial composition allows creating a set of heterogeneous groups; this initial composition is based on a learner evaluation. However, for preventing that the initial distribution becomes static and not adapted during a training session, we must consider a *logical mobility of learners (inter or intra group change)*. This logical mobility must respond to learner's educational needs according to behaviors detected of a group of learners during a training session.

II.3.2 Group Training Service (GTS)

A controller of a Group Training Service (GTS) must adapt a training session to learner's educational needs.
The actors considered in a virtual class are remote learners, teacher and shared didactic resources (didactic and domain entities).
A local didactic strategy is represented by a set of didactic interactions. The controller of a GTS manages these didactic interactions.

The controller of a GTS must establish a dialogue with the controller of the virtual class. This dialogue allows indicating either a learner may change of group or a learner has a problem or a learner send a request (when he wants to express a problem or point of view).
We established with educationalist team three behavior zones inside a group, these behavior zones represent three types of possible progressions: normal, intermediate and critical zones [Hernández 95]. The control of these behavior zones represents a *fine adaptation to knowledge level* and to learner's educational needs.
The architecture for each controller of a group training service (GTS) was specified in [Hernández 95].

III Specification of the Adaptable Virtual Class

The architecture ACVA is distributed, that is, the modules of intelligent tutoring system are distributed in the teleteaching environment. Domain and didactic objects are stored in the central site. Each GTS is located in a decentralized site; each GTS must manage didactic strategy, learner and group profile and behavior zones. The VCTS may be located in a decentralized or central site, it is more interesting to locate the VCTS in a central site, in this way communication with other site is simplified because each GTS already communicates with central site for reusing domain and didactic objects from server. The VCTS must manage pedagogical strategy (e.g. pedagogical objectives), course organization by a set of sessions, group and class profiles, and behavior zones of virtual groups for controlling inter-group changes. Each GTS can be analyzed as a micro-tutor associated to a level knowledge in a virtual class. Several GTS or micro-tutor can be executed in the same time (e.g. a virtual class studying object-oriented paradigm with three knowledge level: novice, intermediate, and advanced) using synchronous communication in a session (e.g. each week). Synchronous and asynchronous interaction are used before next session for doing homework and complementary works. Learners must work with activities and learners associated to real level identified during training session.
We are going to show the architecture and specification of each layer.

III.1 ACVA architecture.

We represent each layer of ACVA architecture as a subsystem. Two layers represent layer of cooperative training services: Virtual class and GTS layers. The notation used is the notation UML (Unified Modeling Language) [Sinan 98]. UML is the standard notation for object-oriented modeling. The UML provides the package mechanism for the purpose of illustrating groups of elements or subsystem [Larman, 98]. A package is a set of model elements of any kind, such as classes, use cases, collaboration diagrams or other packages (nested package) [Larman, 98]. Architecture package diagrams of ACVA are shown in figure 5.

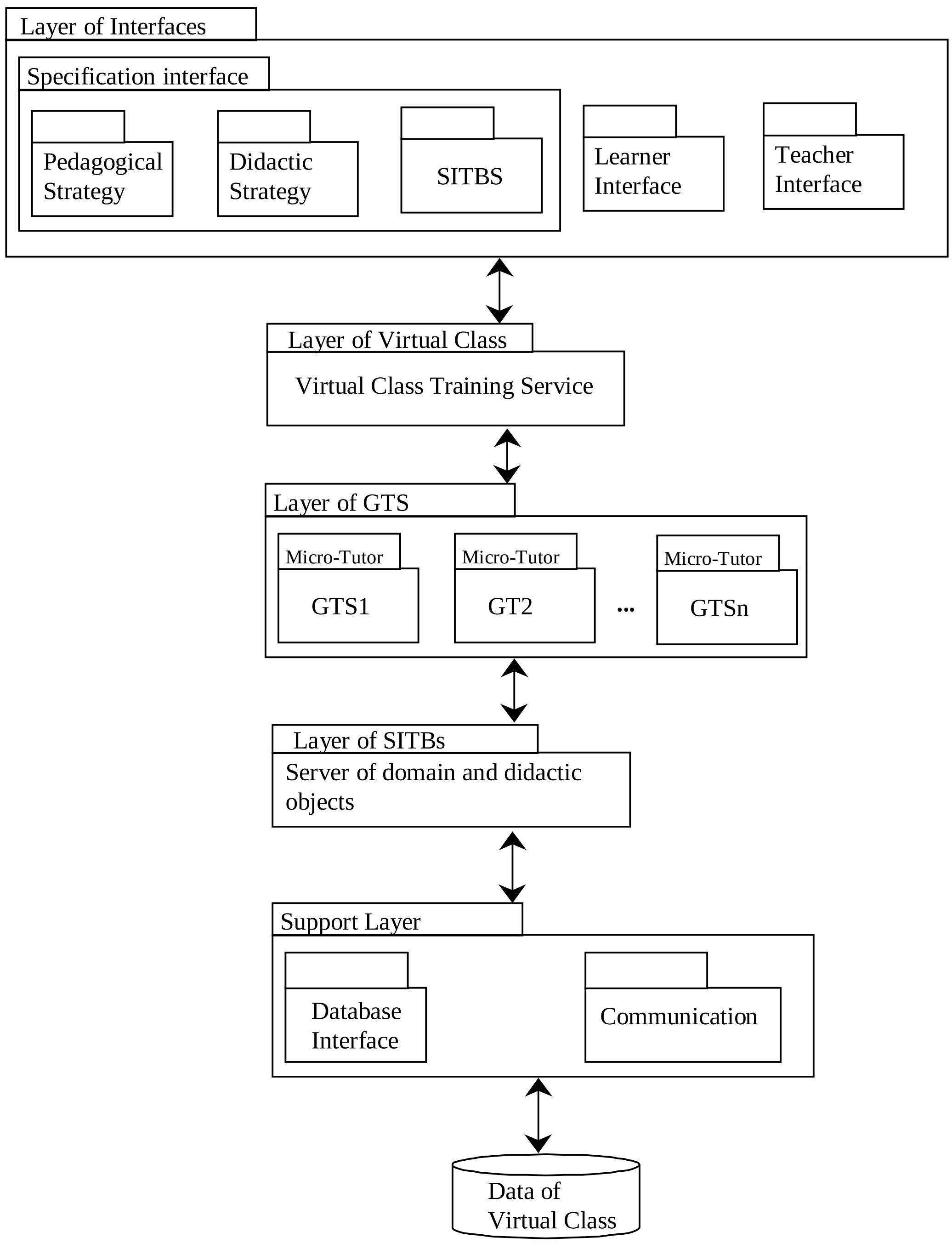


Figure 5: Package diagrams of ACVA architecture.

III.2 Server of reusable SITB (Service Independent Training Building Block)

This server represents the layer of reusable SITB. The coordinator of SITB must control the interactions with GTS (Group Training Service) and VCTS (Virtual Class Training Service) coordinators. A class diagram using UML notation (figure 6) represents the SITBs. The architecture used for implementing this server is shown in figure 7. It is based on Java.

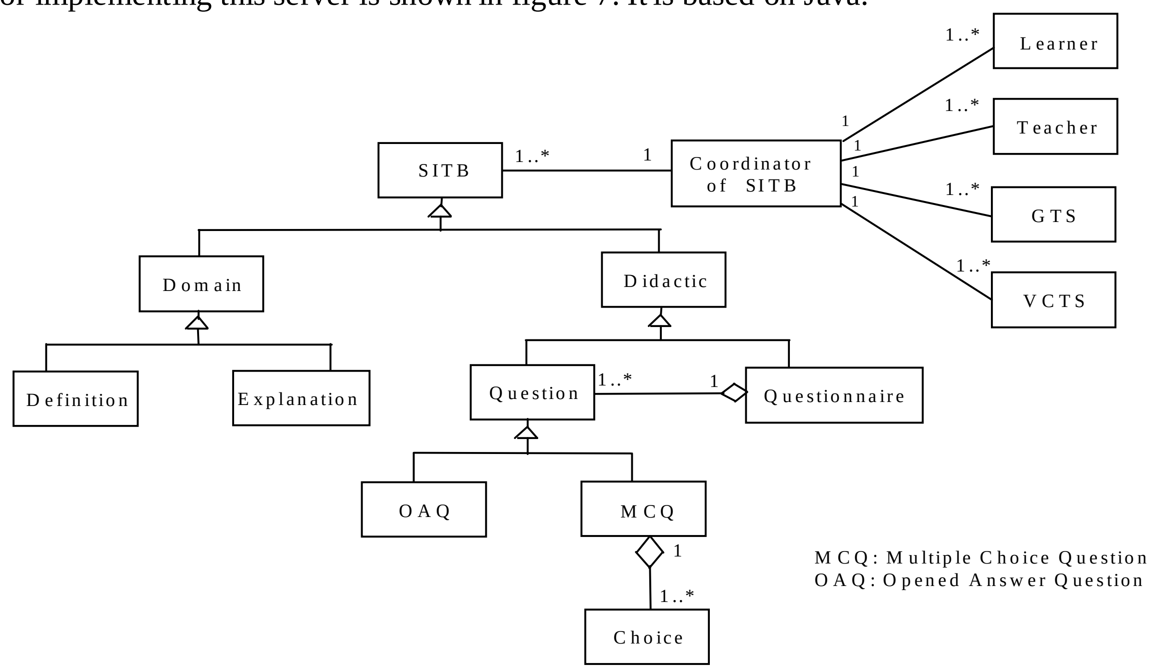


Figure 6: Class diagram representing domain and didactic objects in SITB server.

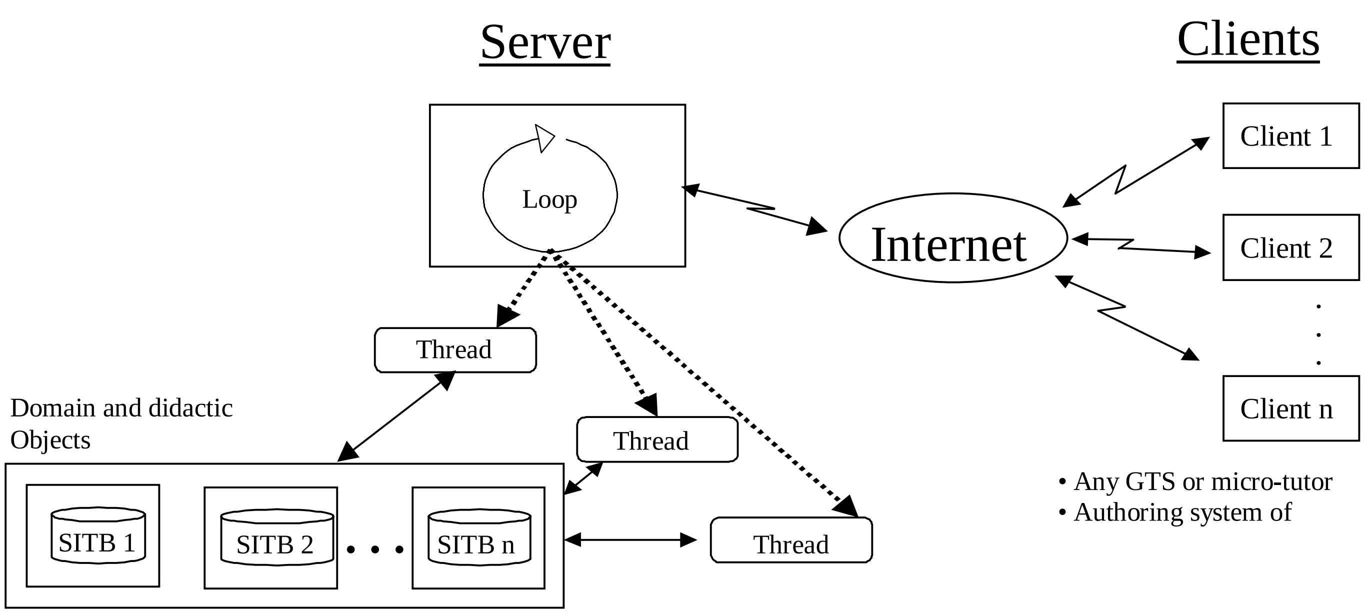


Figure 7: Server implementation of reusable SITBs.

III.3 Group Training Service (GTS)

A Group Training Service belongs to cooperative service layer (figure 8). The coordinator of Group Training Service has to manage: - the communication with the VCTS (Virtual Class Training Service), - the reusing interaction with the server of SITB and - the didactic control taking into account a didactic strategy. One coordinator of GTS allows to consider the behavior of group by behavior zones (which were specified in [Hernandez, 95] and two kind of profiles: learner's profile and profile of group [Hernandez, 95]). In figure 9 a class diagram of a Group Training Service is shown.

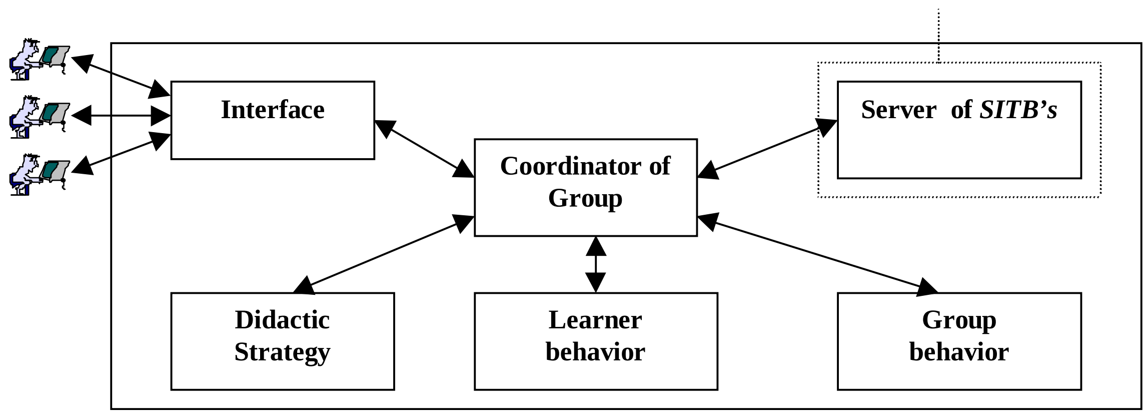


Figure 8: Elements of a GTS or micro-tutor.

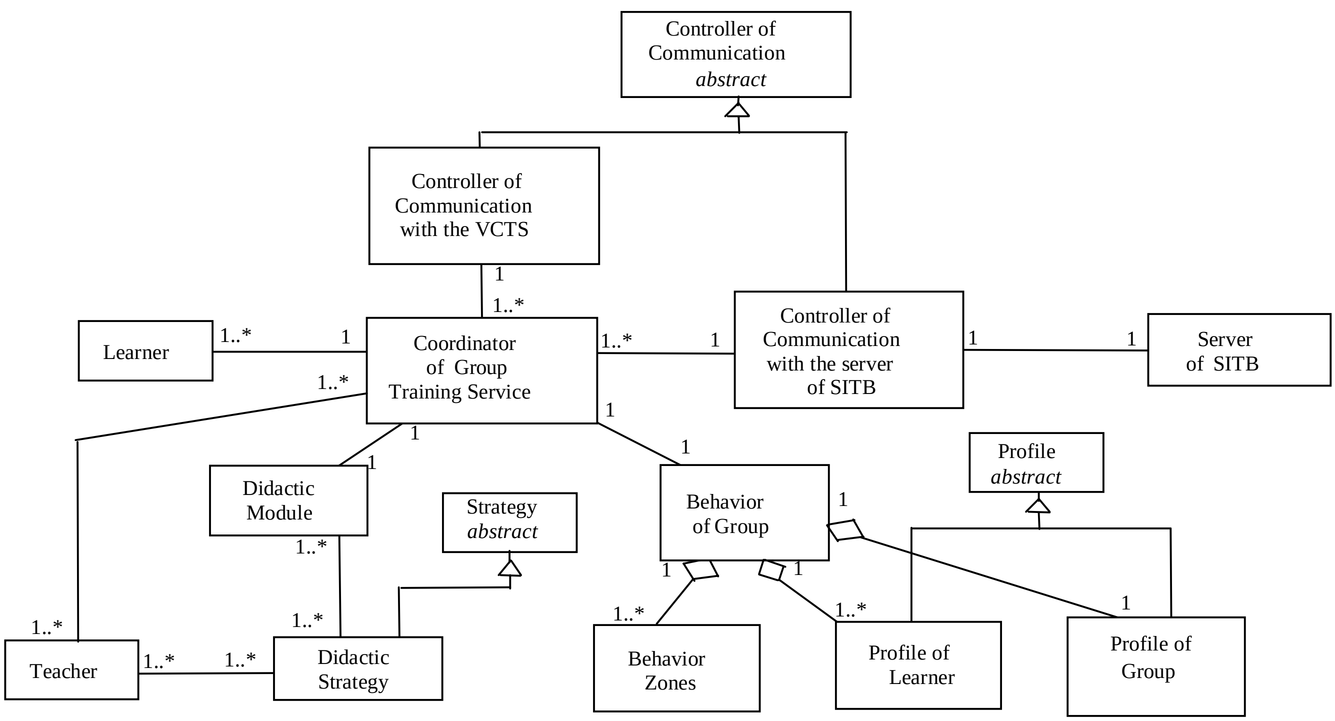


Figure 9: Class diagram of Group Training Service.

From entities represented in the class diagram of Group Training Service a set of agents was identified. Thus a GTS or micro-tutor was specified (based on agent-oriented paradigm) [Socorro, 99] and implemented. The implementation uses reactive agents programmed in Java [Socorro, 2000]. The agents and their interactions in micro-tutor are represented in figure 10.

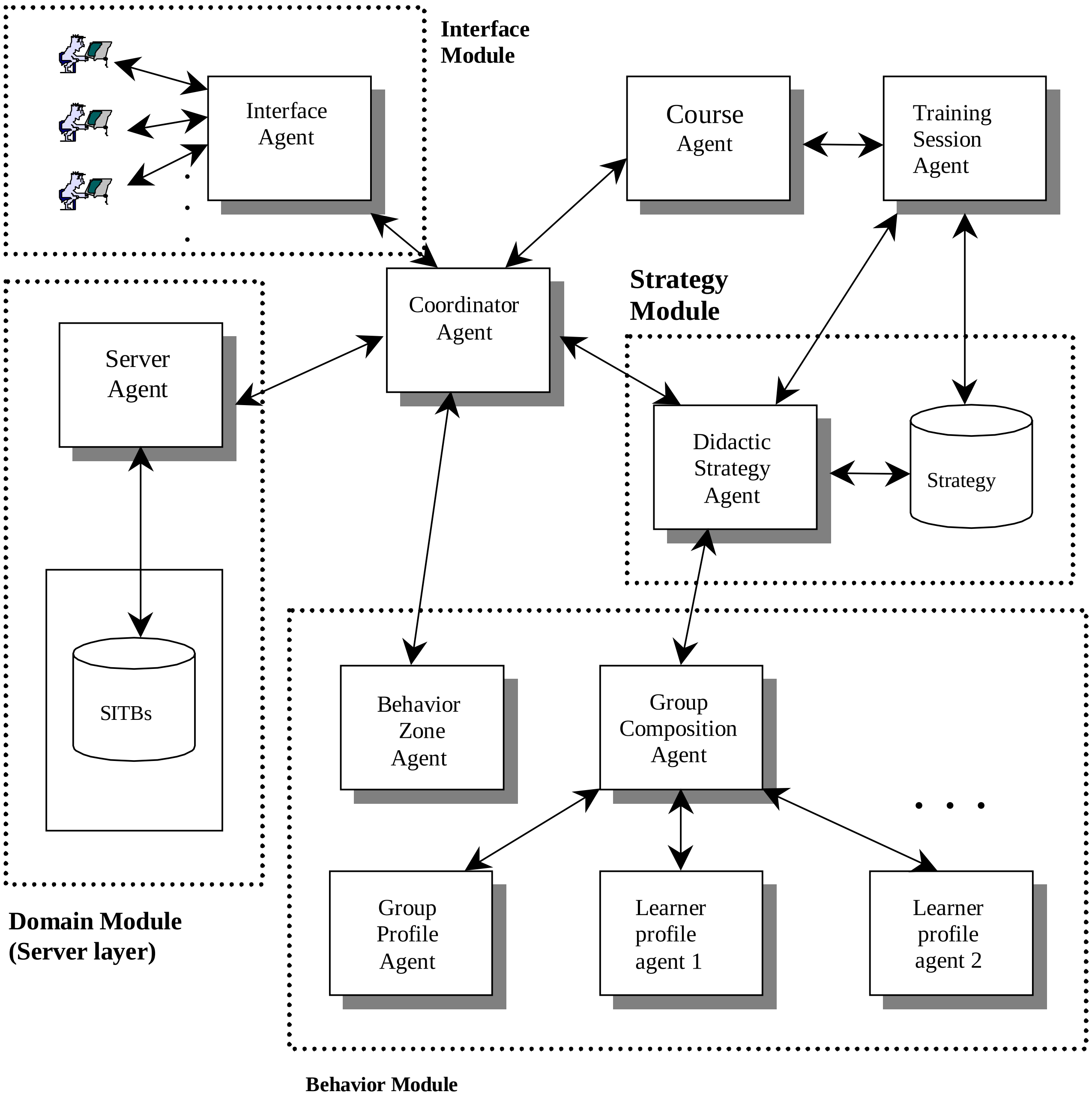


Figure 10: The agents and interactions in GTS or tutor.

Each agent must interact with others agents by sending and receiving of messages. Agent behavior is determined by receiving of events and treatment correspondent. Each agent in GTS has three events (start, message receiving, and finish) based on [Azevedo, 99]. For each event is necessary to specify activities.

Example of agent representation:

<u>Training Session Agent:</u>

if start event:

Connect to communication agent

Send presentation message

if message:

session-start:

Send message to communication agent: Interfaces initialization

Execute current didactic strategy

if finish event:

Disconnect of communication agent

A didactic strategy is specified taking into account several kind of tactic:

<u>Reusing tactic:</u> Recovery of a domain or didactic object from SITB server, the GTS must show interface correspondent to learners.

Example:

Reusing concept ("abstraction", group)

Reusing example ("abstraction", group)

<u>Time tactic</u>: It is possible to establish some time associated to execution of a tactic.

Example:

Reusing exercise (questionnaire, inheritance, 10 minutes, group)

<u>Chat tactic:</u> Represents a conversation synchronous between learners and teacher.

Example:

Chat (group of learners, teacher, 15 minutes)

<u>Rule decision tactic:</u> If condition is true an action associated will be executed, in a condition the group behavior can be taken into account by scores associated to profile of group.

Example:

Reusing concept ("abstraction", group)

Reusing example ("abstraction", group)

Reusing exercise ("questionnaire", "abstraction", 10 minutes, group)

if average of group < 5 then

Reusing concept ("object", group)

Reusing example ("object ", group)

Reusing concept ("class", group)

Reusing example ("class ", group)

Reusing exercise ("questionnaire", "class", 10 minutes, group)

<u>Strategy change tactic</u>: It is similar to a procedure calling, in this case the control flow continues in other strategy.

Example:

Strategy 1

Reusing Exercise (questionnaire, inheritance, 10 minutes, group)

if average of group < 4 then

Change_strategy (strategy 2)

III.4 Virtual Class Training Service (VCTS)

The VCTS also belongs to cooperative service layer (figure 11). The coordinator of VCTS has to manage: - the communication with each group, - the reusing interaction with the server of SITB and - the pedagogical control taking into account a pedagogical strategy. The coordinator of VCTS takes into account the behavior of each group and virtual class by two kind of profiles (profile of group and profile of virtual class).

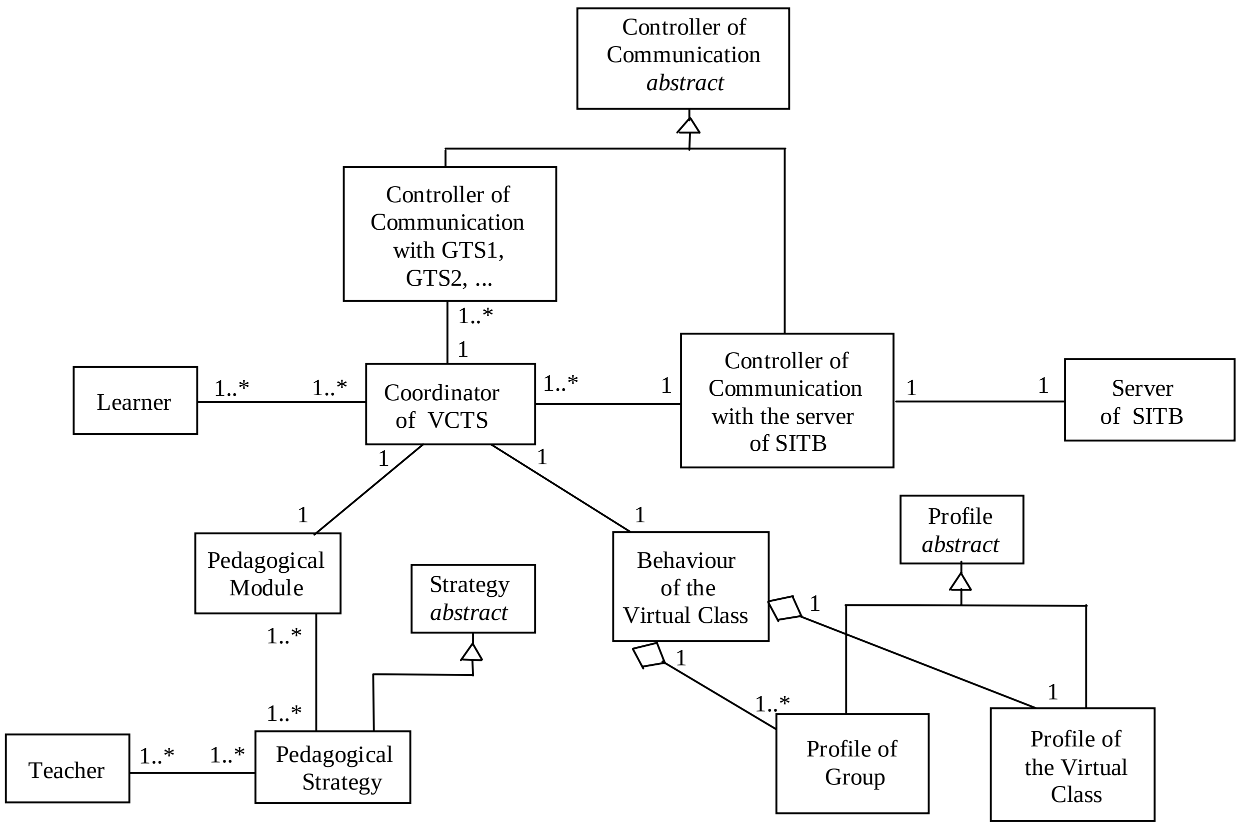


Figure 11: Class diagram of Virtual Class Training Service.

IV Conclusion

The architecture of the virtual class proposed is a response to traditional virtual classes, which are not adapted during training session.

A server of reusable SITB was developed in the object-oriented programming domain. This server was implemented using of Internet and Java. The same server will be used for each coordinator of Group Training Service (GTS). A GTS was developed, the agent-oriented paradigm (in particular reactive agent) was used for modeling and implementing it. Works are in progress for integrating the GTS developed reusing a set of reusable SITBs from SITB server. A tutoring system was developed (using Java), the domain of tutor is the object-oriented approach, and it will be integrate as an element of SITB server. Future works are necessary to integrate the VCTS layer using agent-oriented paradigm.

References

[Azevedo 99] Azevedo (1999). MutAntIS: Uma arquitetura multi-agente para a autoria de tutores inteligentes. 1999. Dissertação de mestrado em informática (master). CT/UFES. Brasil.

[Canut 90] Canut M.F. (1990). Spécification formelle de systèmes d'EIAO pour l'Atelier de Génie Didacticiel Intégré. Thèse de doctorat de l'Université Paul Sabatier, Toulouse III, France, API Laboratory. July 1990.

[Canut 94] Canut M.F. and Eloi M. (1994). Vers la généricité: spécification d'un Shell pour un environnement d'apprentissage basé sur l'entraînement. 9ème congrès Reconnaissance des Formes et Intelligence Artificielle, RFIA 94, 11-14 January, Paris, France.

[Doyon 91] M. Doyon (1991). L'apprentissage coopératif en classe: un mode d'apprentissage. Science et Comportement, 21,2, 1991, France.

[Haton 92] M. C. Haton (1992). L'ordinateur pédagogue. Revue: La recherche, numéro 246, volume 23, Septembre 1992, France.

[Hernández 95]Hernández-Domínguez A. (1995). Systèmes 'Tuteur Intelligent' et 'Collecticiel': L'ARESFED , Architecture pour la Réutilisation et l'Exploitation des Services de Formation dans le contexte de l'Éducation à Distance. Thèse de doctorat de l'Université Paul Sabatier, Toulouse III, France, API Laboratory. July 1995.

[Hernández 96]Hernández-Domínguez A. (1996). An adaptable teaching controller in a virtual class, VII Simpósio Brasileiro de Informática na Educação, Belo Horizonte, Brasil, Novembro 96

[Larman 98] Larman C. Applying UML and Patterns Prentice-Hall, USA 1998.

[Sinan 98] S.S Alhir (1998). UML in a Nutshell. O'Reilly, USA.

[Socorro 99] Socorro da Silva A and Hernández-Domínguez A. (1999)Uma modelagem baseada em agentes de um Tutor no contexto de uma Classe Virtual Adaptativa, Workshop: Ambientes de Aprendizagem Baseados em Agentes , X Simpósio Brasileiro de Informática na Educação (SBIE'99), Curitiba, Brasil, Novembro 99.

[Socorro 2000]Socorro da Silva A (2000). TUTA - Um tutor baseado em Agentes. Dissertação de mestrado em informática (master). Campina Grande DSC/UFPb, Brasil. Fevereiro 2000.

[Vincent 94] B. Vincent B. (1994). Contribution à la représentation des connaissances en comptabilité. Doctorat de l'Université de Toulouse I, laboratoire A.P.I., Nov 1994, France.

[Wenger 87] E. Wenger (1987). Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge. Morgan Kaufmann Publishers, Inc. California, USA, January 1987.