ONTOLOGIES AND PETRI NETS TO ALLOW THE MANAGEMENT OF STUDENT GROUPS

Eliane Pozzebon[1]

Abstract *- Managing learners' groups is a dynamic task, which depends on several asynchronous events (learners' availability, data in the models, current pedagogical situation, declared preferences). Once the group is formed, its activity must be managed, generating an interaction protocol among the members and a group model, with information on their activities. This paper proposes a model based on ontologies and Petri nets to allow the management of student groups in the Intelligent Tutoring System (ITS) context. The model includes a predefined library of the group's activities. To establish a group activity, the teacher chooses a scenario from the library, supplies the parameters and the content of the activity. The information treatment of the student model is automatically included in the Petri Network's transition as preconditions and conditions.*

1. INTRODUCTION

Groups learning techniques offer advantages for learners and teachers, since they contribute to: the development of critical thinking and abilities to solve problems; the increase in motivation levels when the learners are familiar with group work; improving the learners' development, leading to a decrease in anxiety and to improvement on their self-esteem; searching for the learners' different learning styles, such as verbal and visual [8].

Managing learners' groups is a dynamic task, which depends on several asynchronous events (learners' availability, data in the models, current pedagogical situation, declared preferences). Once the group is formed, its activity must be managed, generating an interaction protocol among the members and a group model, with information on their activities.

Using multi-agent systems allows a level of abstraction more adequate to approach complex and distributed problems, such as work environments and group learning. With an agent architecture developed to support a group formation process, it will be possible to establish negotiation among the agents, thus defining meetings or reunions with the group's learners.

The ITS maintains information on the student in a database called student model. A student's models contains information about him/her, such as: preferences, knowledge levels, solved exercises, tasks to be completed, etc. With updated information, it is possible to check which learners have the same knowledge level and group them.

After creating the groups, it is necessary to define the means to integrate the group members, so they can help each other when they have difficulties with an activity. The group support issue in intelligent tutoring systems is quite complex, and deals with the interactions between the students and the ITS, and also presents a knowledge domain according to the pedagogical strategies determined by the teacher.

The purpose of this paper is to contribute to the conception of interactive learning environments, helping developers by defining a model to manage groups, using an authorship tool. This way, teachers who are not familiar with the programming tool can configure the learning format individual or in groups with diversified scenarios.

GROUP LEARNING SUPPORT MODEL IN AN ITS

The model to support group management in Intelligent Tutoring Systems is divided into the specification and execution levels. In the specification level, it adopts ontologies to represent the domain, student and group models, and Petri nets to specify the interaction protocols. In the execution level, it makes group learning operational with multi-agent architecture; this architecture allows collaboration among the students.

The model integrates the Artificial Tutoring Agents Society (ATAS) of the MATHEMA model [6] and the Tutoring Systems' Authorship Tool (TSAT) [7] to create the ITS's.

Specification Level

MATHEMA [6] provides a modeling scheme for the domain, divided into two view, the external view and the internal view. This enables the construction of group interactions that, although they are not dependent of the domain, they can explore the domain structure, going beyond the mere communication support between the group members and the instructor. This is possible because these group interactions can use the same problem solving activities already defined in the ITS context.

In the group interaction management student models are used; one of the advantages is that they can be defined as an extension of the student model in the ITS, in a way the group interaction manager can explore the preferences and previous results achieved by each student in the individual learning context during their interaction with the ITS

1 Eliane Pozzebon, Colegiado Engenharia da Computação, Universidade Federal do Vale do São Francisco, CEP 56304-917 Juazeiro – BA – Brasil
Pós-graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina, CEP 88040-900 Florianópolis – SC – Brasil

The student, group and domain models are represented by ontologies and were inspired by the works of [7, 5, 9, 11].

The **domain model** contains definitions for the internal view of the MATHEMA conceptual model. A course developed by using the proposed model is represented as an instance of the domain model and contains all the information provided by the author. This information is of two kinds: properties and contents. Examples of properties are requirement relationships, the detail or difficulty level, etc. Contents is what is presented to the student, for instance, in HTML pages. The concept of Problem and its sub-concepts is used in the definition of Content Unit in this model.

The **student model** contains definitions for all the concepts needed to characterize a student and his/her history of interaction with the system. Its contents include static information, such as education level, based on a preliminary test, and preferences; and also dynamic information, which consist of student activity descriptions during their interaction sessions with the system.

The **group model** contains definitions of the concepts needed to specify the group activity, which include: Group, Scenario, Activity Unit, Group Activity and Interaction Protocol (figure 1).

The concept of **Group** is a set of students already in the ITS. These students' individual information is stored in the Student Model.

The concept of **Scenario** consists in an operational definition of the group's activity. It includes activity units and interaction protocols. Scenarios are defined by the Developer and stored in a scenario library. They are built using predefined activity units, which can be reused in different scenarios.

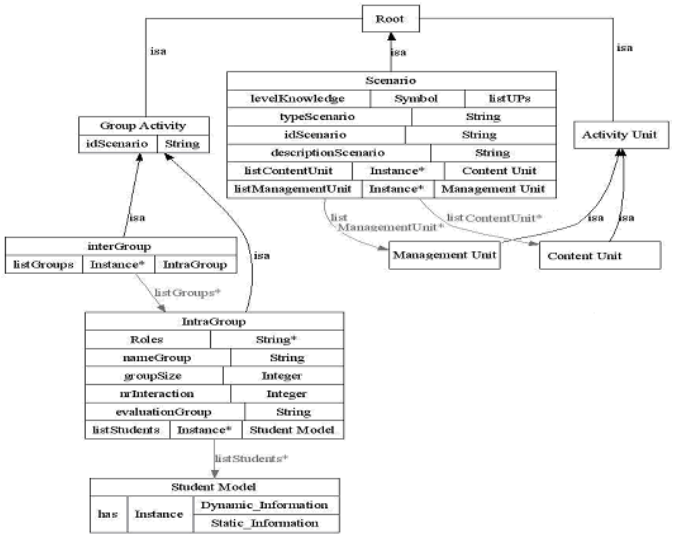

Figure 1 : Group Activity Model

The concept of **Activity Unit** defines the different activities that occur in a certain scenario. There are two kinds of activity units, according to the ontology presented on Figure 2.

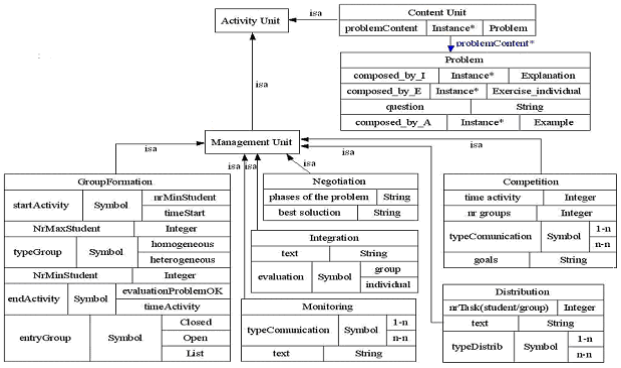

Figure 2 : Activity Unit

Content units: used to define the problem solving tasks associated to a given scenario. These tasks are defined using the Problem specification (which includes Interaction Units) from the ITS domain model. The content definitions are provided by the Author, using an authorship tool interface.

Management units: used to define the typical group interaction activities, such as group formation, problem distribution, negotiation, competition, solution integration, group member instruction, etc.

The concept of **Group Activity** is the control of group activities. Each group uses a scenario. The type of scenario can be called Inter-Group or Intra-Group.

In the Inter-Group activity interactions occur among various student groups, such as a competition, as in the Intra-Group activity the interactions are restricted to a student group.

The Inter-Group class contains the group lists and the Intra-Group class contains the information resulting from the interactions, such as the group participant list, group evaluation and others.

The group members can be assigned different roles. The concept of **Role** structures the group members into classes, according to their participation in a scenario. Some examples of parts would be Team Leader, Participant Member, played by the students, and Instructor. This information is stored in the Student model.

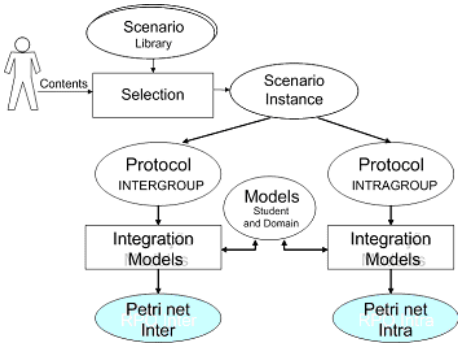

Figure 3 : Specification of an Interaction Protocol

The concept of **Interaction Protocol** contains the operational specification of the group's activity, i.e., the order in which the activity units are carried out in a given scenario. It is represented by a hierarchical, two-level Petri net. Specifying an instance for an interaction protocol is a two-step process (Figure 3).

The first step consists on selecting a scenario from the scenario library and instantiating all of its attributes.

The second step compiles this information to produce the Petri net that defines the interaction protocol. The compilation process automatically integrates the models in the transition conditions of the Petri net. This integration provides the adaptive character of the interaction protocol.

Model Application to Support Group Learning

The scenario chosen for the group activity in an on-site course was defined based on the ontology.

This group activity develops the "divide and conquer" strategy to solve problems. It assumes a problem that can be divided into a certain amount of sub-problems. Each sub-problem can be solved independently and their solutions should be combined to solve the original problem.

The "divide and conquer" technique was chosen because it is a classic model of a knowledge fixation technique, and used as a technique applicable to classrooms. It adjusts to any technical, artistic or scientific content, and can be applied to different education levels.

1.1 Specification Level

According to the specification level in a group activity, defined in Section 2.1, the following concepts are defined:

- **Group:** the activity needs at least one student per sub-problem.

- **Roles:** the activity includes three roles: supervisor, sub-problem solver and solution integrator. The supervising role can be given to the course instructor or to the supervising agent. The solution integrating role should be given to one or more students, who will be responsible for integrating the sub-problem solutions. These students can be chosen dynamically. For example, the first one to complete a sub-problem solution or the one with the best score in an ITS. Finally, the role of sub-problem solver is given to all the students participating in the activity. The group members involved in the activity should have the experience needed to solve the problem in question.

- **Scenario:** it is defined by the management units and content units.

- **Management units:** the following management units are necessary to control the scenario:

  - Group Formation: invites the students, receives acceptances and controls the number of members per group.
  - Initiates the groups' activity: choosing the scenario and creating the supervising agent responsible for the group.
  - Sub-problem distribution.
  - Monitoring the sub-problem solutions.
  - Coordinating the interaction among group members that incorrectly implemented the interface in their solutions.
  - Integration group designation.

- **Content units:** the scenario contents, to be provided by the author through the authorship interface, consist on the following problem descriptions:

  1. A general explanation of the problem and of its sub-problems.
  2. To each sub-problem:
     - a detailed explanation;
     - one or more examples of similar problem solutions; and
     - exercises: one that tests the sub-problem solution and one that verifies the implemented solution.

One should notice these contents are instances of the problem concept (and of the interaction unit) of the domain model ontology, and can also be used in the context of an individual interaction with the ITS.

- **Interaction protocol:** the highest level in the interaction protocol corresponding to this scenario is represented by the Petri net illustrated on Figure 5.

In this context, the tokens contain an instance of the group concept. For readability reasons, we omitted the object dimension in the Petri net, which is the token values, Preconditions, Actions and Emission Rules for the transitions.

The resulting abstract Petri net only maintains the aspects related to the behavioral structure of the protocol. However, from this abstract structure, standard properties of the Petri net can be proved, like the presence of cycles (transition sequences that can be infinitely repeated), deadlocks (a blockage state from which no transition can occur), a state's (initial or final) (in)accessibility , boundness (infinite growth of the number of tokens) or the token loss in a place/state.

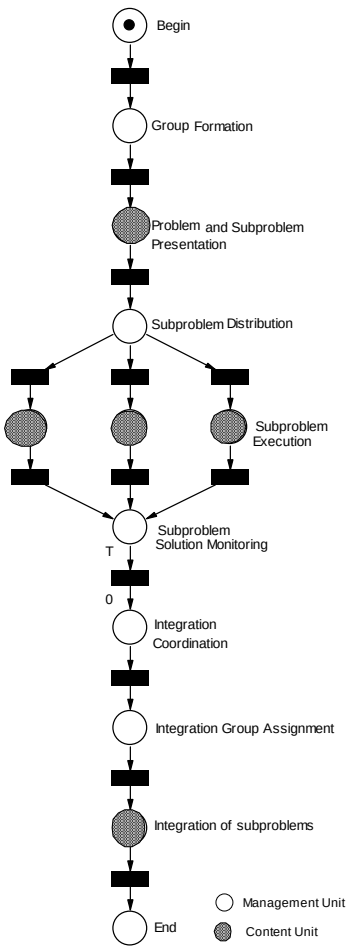

Figura 5: Petri net modelling the interaction protocol of a scenario [14]

1.2 Scenario Instance

To instantiate the group activity for the "divide and conquer" problem solving strategy, a group activity was implemented, based on an existing individual learning ITS for the Information Structure domain, a, undergraduate discipline in the Automation and Control Engineering in the Federal University of Santa Catarina.

The problem to be solved during the group activity is defined as follows:

- Problem description: given a programming language that supports mathematical operations with integer numbers, how can it be used to support operations with other types of numbers (rational, fluctuating point and complex).
- Sub-problems: mathematical operation packages to each of the three new types of numbers, including conversion functions.
- Integration: a function package that integrates all four types of numbers.

The implemented group activity must be developed during an on-site course. The instances of the relevant concepts involved in the definition of the group activity are defined a follows:

Group: the Students that participate in the activity must have completed the necessary pedagogical units (basic programming, abstract data types).

Roles: the supervising role is given to the course teacher. The sub-problem solving role is given to all the students in the class, and the solution integrating role is given to the Students who are members of the first group to successfully solve the given sub-problem.

Management units: the implemented activity uses a synchronous group formation in which an invitation is sent to all the students in the class. The Students must reply identifying their favorite sub-problem. The system automatically controls each group's maximum size. The necessary requirements are also checked for each Student. The highest level Group Formation location in the Petri net (see Figure 5) is expanded to the lowest level Petri net, illustrated on Figure 6.

Problem distribution is generated automatically. The sub-problem solving monitoring is based on exercises included in the content units. The interface problem coordination is done under the supervisor's responsibility, through a chat and mural tool.

Content units: the lowest level Petri net that implements the sub-problem and solution integration units is implemented through the use of the TSAT tool. Its general shape is illustrated in Figure 7, where Exp, Exa and Exe are interaction units that present to the students, respectively, explanations, examples and exercises.

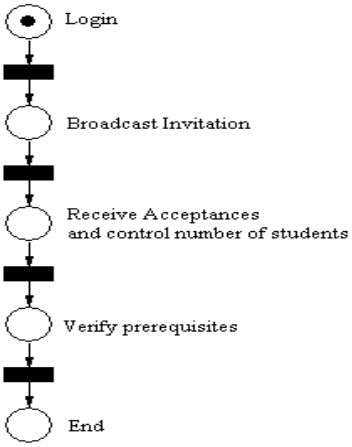

Figure 6 : Petri Net Modelling the Group Formation Protocol

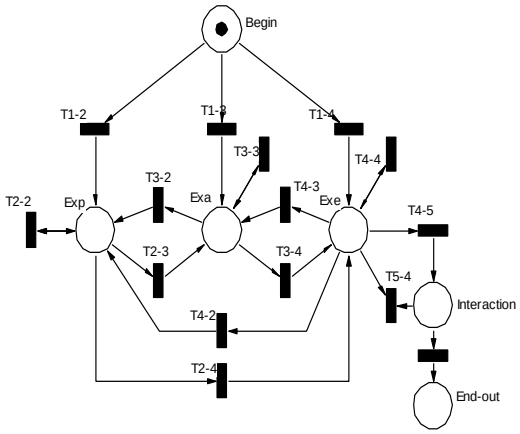

Figure 7 : Petri Net Modelling the Problem Solving Protocol [14]

In literature, we found some works partially similar to the model presented in this paper, such as *Redeem* [1], which adapts the tutor to the students in the group using the student and domain model, the WHITE RABBIT [13] and the Computer Model based on Vygosky's [2] theory, using an agent as the student's representative and mediating agents for group work, EASE [3], which uses ontologies and rules, and COLE [4], which uses the idea of scenario instantiation, but does not store information in a scenario library so it can be reused by the teachers.

Conclusions

In this paper a new model was proposed to manage groups in Intelligent Tutoring Systems. This information is compiled into a Petri net that monitors the group activity.

The proposed model explores the domain model and the student model information, not to mention the group model, created in a way the scenario is divided into units. Each scenario unit can be reused by the Authors to build new scenarios, with new pedagogical strategies.

As future work, implementing different scenarios and testing with student groups. Enabling, from the scenario library, customized and automatic content selection to generate new resources and adaptation, according to the students and teachers' preferences.

REFERENCES

[1] Ainsworth. S.E. Using a single authoring environment across the lifespan of learning in Educational Technology and Society. vol 3, pages 22-31, 2007.

[2] A. F. de Andrade, P. A. Jaques, J. L. Jung, R.H. Bordini, R.M. Vicari: A Computacional Model of distance Learning Based on Vygotskys Socio-cultural Approach, *X International Conference on Artificial Intelligence on Education*, San Antonio-Texas 19-23, pg 33-40, Maio de 2001.

[3] L. Aroyo, e A. Inaba, L. Soldatova, R. Mizoguchi, et al.: EASE: Evolutional authoring support environment, *Intelligent tutoring systems*, Lecture notes in computer science, 2004.

[4] H.J.S de Azevedo, E. Scalabrin, A Humam Collaborative Online Learning Enviroment Using Intelligent Agents, *Designing Distributed Learning Enviroment with Intelligent Software Agents*, Informatiom Science Publishing, pages 1-32, 2005.

[5] W. Chen, B. Wasson, Intelligent Agents Supporting Distributed Collaborative Learning. *Designing Distributed Learning Environments with Intelligent Software Agents*. IDEA Publishing Group, pg 146-161, 2005.

[6] E.B.Costa,., M.A Lopes, E. Ferneda.: Mathema: A learning environment based on a multi-agent architecture. In *LNAI - Advances in Artifical Intelligence*, v 991, 141-150, 1995.

[7] L.B. Frigo, Um modelo de Autoria para Sistemas Tutores Adaptativos, PGEEL, Tese de doutorado, Universidade Federal Santa Catarina, 2007.

[8] M. W. Goldberg, S. Salari, P. Swoboda: World Wide Web - Course Tool: An Environment for Building WWW-Based Courses. in *Fifth International World Wide Web Conference*, May 6-10, Paris, France, 1996.

[9] A. Inaba, T. Supnithi, M. Ikeda, R. Mizoguchi, J. Toyoda: How Can We Form Effective Collaborative Learning Groups? in *Proc. Intelligent Tutoring Systems*, pages 282-291, 2000.

[10] G. McCalla, The fragmentation od Culture, Learning, Teaching and Technology -Implication for the Artificial Intelligence in Education Research Agenda in 2010. in *International Journal Of AI In Education*, 177-196, 1999.

[11] R. Mizoguchi, J. Bourdeau, Using ontological engineering to overcome common AI-ED problems. *Journal of Of Artificial Intelligence In Education*, 11(2):107-121, 2000.

[12] J. Self.: The Defining Characteristics Of Intelligent Tutoring Systems Research: Itss Care, Preciselys. in *International Journal Of Artificial Intelligence In Education*, 1-10, 1999.

[13] M. Thibodeau, S. Belander, C. Frasson, White rabbit.matchmaking of user profiles based on discussion analysis using intelligent agents. *Proc of 5th International Conference ITS*, 2000.

[14] POZZEBON Eliane, CARDOSO Janette (IRIT, France), BITTENCOURT Guilherme (UFSC, Brazil), Chihab Hanachi (IRIT, France), A Group Learning Management Method for Intelligent Tutoring Systems, In: International Journal of Computing and Informatics, Vol 31, Number 2, pages 191-199, June 2007.