S. L. Wong et al. (Eds.) (2010). Proceedings of the 18th International Conference on Computers in Education. Putrajaya, Malaysia: Asia-Pacific Society for Computers in Education.

Modeling and Personalizing Curriculum Using Petri Nets

A.J. ARMSTRONG[a*] & Fuhua LIN[b]

[a] *School of Information Communication and Engineering Technology, Northern Alberta Institute of Technology, Canada*

[b] *School of Computing & Information Systems, Athabasca University, Canada*

*aja@nait.ca

Abstract: We introduce and discuss our ongoing research project using a multi-agent system to model, personalize and track individual students' curriculum within the standards of a given course while monitoring and maintaining opportunities for intra-student collaboration. A Petri Net data structure models the relationships amongst learning objects within the course. A particular marking of the net indicates the current state of a given student or entire class of students. Agents use the net to represent their beliefs and to make plans that maximize curriculum personalization opportunities without compromising collaboration between students.

Keywords: Multi-Agent Systems, Curriculum, Collaboration, Petri Nets, BDI

1. Introduction

Our primary research goal has been to present an approach to constructing adaptive learning systems that exploit multi-agent systems (MAS), machine learning, and novel course modeling to the problem of generating and coordinating student-centered study plans that maximize learners' outcomes while simultaneously reducing instructor workload. A problem in instructional design is the tension between providing personalized learning programs and balancing the workload of instructors. One approach is to marshal technology to provide learners with individual e-learning experiences without requiring the detailed attention of a human instructional designer. Significant research has been conducted into adaptive hypermedia education systems (Henze et al., 2004; Brusilovsky & Peylo, 2003).

Adaptive learning systems provide the promise of improved pedagogical integration of technology for large classes of students with dissimilar goals and preferences. One issue is that a high degree of personalization necessarily limits the opportunities for collaboration. Hamilton & Jago (2010) emphasize the importance of *modeling* and *connectedness* to promote customized and connected learning experiences. *Modeling* involves creating structured representations of systems for exploring a complex knowledge domain. *Connectedness* refers to socialization in learning, including rich multilayered connections between individuals.

We have developed our *eInstructor* platform as a system that depends on interaction between agents to balance two antithetical goals: personalized curriculum for a student and collaborative learning amongst groups. We have implemented a platform that supports the implementation, testing and comparison of several algorithmic approaches. This includes the agent infrastructure, course progression and status modeling system. Of particular interest is the Petri Net data structure used to model the relationships amongst course elements, document student states and inform student and system plans.

S. L. Wong et al. (Eds.) (2010). Proceedings of the 18th International Conference on Computers in Education. Putrajaya, Malaysia: Asia-Pacific Society for Computers in Education.

2. Modeling Course Progressions Using Petri Nets

We consider a course as some set of learning objects (LOs) and define course completion as completion of some subset of those objects, with several possible sets being available. For example, a course requirement might be met by one of several distinct learning objects, each catering to a different learning style. Learning objects can have pre-requisite skills or knowledge that might be fulfilled by other learning objects. Thus, learning objects can be ordered within a course, with early objects fulfilling the requirements of later objects. As a set of learning objects might fulfill the requirements of another set of objects, this progression is better envisioned as a complex directed graph than a simple path through the objects. For a given course, there might be multiple completion paths that represent a distinct set and ordering of learning objects.

We determined that the Petri Net (PN) (Reisig, 1992) would be an ideal data structure to represent this view of a course. The PN is a graph composed of three elements: places, transitions and arcs. We model a learner’s particular cognitive state as a place (circle). An LO is a transition (rectangle). The “firing” of a transition (completion of an LO) causes the learner to transit from an earlier cognitive state to a more advanced one. Arcs connect places to transitions to places. A transition can only fire (an LO can only be completed) if all immediately preceding places have been filled. When the learning object is completed, all immediately subsequent places are filled. We mark states with some sort of token to indicate that they are filled.

This simple set of rules permits us to model quite complex relationships amongst learning objects within a course, including conjunctions and disjunctions of predecessors. We define an initial state, representing starting the course or completing all course pre-requisites, and a final state representing completion of the course requirements. Any sequence of transitions starting from the initial state to the final state is a valid completion path for the course.

One of the most important parts of the background logic is the construction of PNs for a course based on metadata of the LOs contained within that course. We consider the places in the PN to be competencies, build fragments around each LO using these competencies, and then join together competencies that are the same. Thus, the PN essentially build itself by matching input competencies with output competencies.

3. Use of the Petri Net to Model and Track Student Status

Once the course network has been constructed, a student’s current status can easily be determined, as can their immediately available learning objects. The system marks the course graph by firing the transitions representing learning objects that have been completed for that student. This graph can be calculated by any student agent, representing that agent’s beliefs about its state and that of the course.

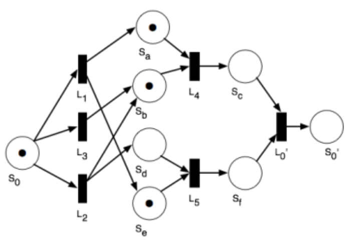

Fig 1: Student Status Net

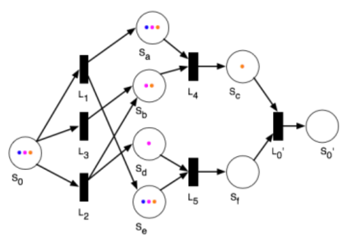

Fig. 2: Class Status Colored Net

The graph in Figure 1 shows a student that can currently attempt only learning object L_2 or L_4 (L_1 and L_3 have either already been completed or would not advance the student). L_5 could only be attempted after L_2. Not only is the set of learning objects that can be undertaken immediately available, but all paths from the current state to the end of the course can be readily determined. In addition, even if the graph changes (perhaps by the addition of new learning objects), the student's current state can be reconstructed in time linear with the number of learning objects that have already been completed.

By expanding our model to a Colored Petri Net (Jensen, 1997), we obtain the ability to model the current state of all the students in a course. It also offers the facility that the system can quickly determine immediate and future opportunities for collaboration. A Colored Petri Net adds the facility that a given state may be marked with tokens of different "colors". Each student is assigned a token color, and the Petri Net state for the entire class can be calculated in time linear with the product of number of students and number of learning objects. This allows easy assessment of opportunities for collaboration. For example, the network in Figure 2 shows that if students are encouraged to complete learning object L_2, more opportunities for collaboration by multiple students in L_4 and L_5 will be engendered. In addition, student agents can be informed that a particular learning object requires a minimum number of collaborators, and they can negotiate paths that allow a sufficient number to arrive at that point in the graph at the same time.

5. Conclusion

Innovative coordinated planning and control mechanisms from multi-agent systems, computational intelligence, as well as game theory and dynamic optimization will be needed to integrate modeling, control, communication and computing concerns into a single architecture. The proposed system is being realized through a Moodle learning management system (LMS) in which a variety of students are connected to each other and to a JADE and Jason-based multi-agent system.

MAS have been identified as one of the most suitable technologies to contribute to this domain due to their appropriateness of modeling autonomous individual students as agents in a flexible way and therefore take advantage of algorithms and techniques for group decision making such as voting and coalition formation that is proposed in the MAS literature. The architecture we have developed provides the Petri Net data structure for courses, which is a valuable tool for representing agent beliefs and for providing agents with a consistent framework for generating and representing plans that allow them to realize their desires. When coupled with the framework for agent interaction, a powerful tool for exploring adaptive collaborative learning systems is provided.

References

[1] Henze, N., P. Dolog, W. Nejdl. (2004). Reasoning and ontologies for personalized e-learning in the semantic web, Educational Technology & Society, 7(4), 82-97

[2] Brusilovsky, P. and Peylo, C. (2003). Adaptive and Intelligent Web-based Educational Systems. *Int. J. Artif. Intell. Ed.* 13, 2-4 (Apr. 2003), 159-172.

[3] Hamilton, E. and M. Jago, (2010), Toward a theory of Personalized Learning Communities, in Design for Learning Environments of the Future, M. J. Jacobson and P. Reimann (eds.), Springer, 263-281

[4] Jensen, Kurt (1997). *Coloured Petri Nets*. Springer Verlag. ISBN 3-540-62867-3

[5] Reisig, Wolfgang (1992). *A Primer in Petri Net Design*. Springer-Verlag. ISBN 3-540-62867-3