

Applying Petri Nets to Model SCORM Learning Sequence Specification in Collaborative Learning

H. W. Lin, Wen-Chih Chang, *George Yee, Timothy K. Shih, **Chun-Chia Wang and Hsuan-Che Yang

*Department of Computer Science and Information Engineering
Tamkang University, Tamsui, Taiwan, R.O.C.*

nbird@mail.mine.tku.edu.tw

**Institute for Information Technology, National Research Council Canada, Ottawa, Canada*

***Department of Information Management*

Northern Taiwan Institute of Science and Technology, Peitou, Taipei, Taiwan, R.O.C.

Abstract

With the rapid development of Internet technology and Web-based education, distance learning provides a novel learning style, which is different from traditional education. In order to adapt different teaching strategies in accordance to individual students' abilities in a distance learning environment, system directed navigation of students was proposed in a distance learning standard called SCORM (Sharable Content Object Reference Model). We introduce the Distance-learning Color Petri Net (DCPN), applying the features of Petri nets, to decrease the complexity of the sequencing definition model in the SCORM 2004 specification. We thus construct a sequencing framework for various instructional strategies by piecing DPCN subnets together.

1. Introduction

Distance learning allows students to learn anytime in any location. In general, budget and time for traveling are saved. Distance learning can be categorized into two styles, according to the facility used. One is synchronous learning. Another is asynchronous learning. In synchronous learning, learning content and the instructor's presentation are delivered in real-time. On the other hand, asynchronous distance learning allows students to look at CD ROM or Web-based instruction. The instructor and students can be located in different positions in both the spatial and the temporal domains. With this advantage of asynchronous distance learning in mind, Learning Management Systems (LMS) are able to provide on-line content, member accounts, discussion board and other help facilities. With the increase in the variety of Learning Management Systems, inconsistent representation of content databases results in a time consuming process to move a course from one LMS to another.

A few committees have proposed infrastructures or specifications of standards for asynchronous distance learning, such as the Learning Technology System Architecture (LTSA) [1], the Sharable Content Object Reference Model (SCORM) [2], and the IMS Simple Sequence Specification [3]. It is very important to have standardized representation of contents, which can promote the development of e-learning in industry, academia, and government.

In order to support collaborative learning in an e-learning specification, it is necessary to understand the navigation behavior of students. We use Petri Nets to model the behavior of the simple sequence specification of SCORM. By applying learning sequencing behavior modeling, we have determined new approaches to support collaborative learning in SCORM, which are not yet defined in the latest version of the standard specification.

In this paper we first discuss some important issues of SCORM. In the related work section, we briefly describe the role of SCORM in distance learning. Then, we explain why we use Petri Nets to model the SCORM sequence. In section 3, we give a formal definition of our proposed model – the Distance-learning Color Petri Net (DCPN). Furthermore, we take examples of SCORM from the Advanced Distributed Learning organization to illustrate our DCPN. The results show that the proposed DCPN framework can fully model the SCORM sequence. As a main contribution, we simplify the DCPN to support collaborative learning in section 4. The last section is the conclusion, where we summarize our contributions.

2. Related Work

We divided this section into two parts to describe the background of this research. We begin with the introduction of SCORM. Then, Petri Nets are briefly described.

2.1 SCORM

SCORM establishes a technical foundation of e-learning via standardization. It is focused on defining reusable learning objects, developing new content models, developing learner assessment models, and creating new models for sequencing content. The three main portions are the Content Aggregation Model (CAM), the Run-Time Environment (RTE) and the Sequencing and Navigation (SN).

CAM defines the responsibilities and requirements for building content aggregations (e.g., course, lessons, modules, etc.). CAM contains information on creating content packages, applying metadata to the components in the content package and applying sequencing and navigation details in the context of a content package. The SCORM Run-Time Environment communicates with its underlying Learning Management System (LMS) that is both machine independent and operating system independent. Sequencing and navigation define the learning sequencing behavior and the learner navigation.

2.2 Petri Nets

The Petri Net is a graphical and mathematical modeling tool of information flow that was proposed by Carl Adam Petri [4, 5], in 1962. A Petri Net is represented by a directed, bipartite graph in which nodes are either places or transitions, where places represent conditions (e.g., parasite in blood stream) and transitions represent activities (e.g., invasion of host erythrocytes).

A Petri Net can also be evaluated and analyzed by a simulation tool. Here are some reasons why we adopt Petri Nets to map SCORM Sequences:

- **Graphic Characteristic:** Graphical representation helps the designer to understand the big picture.
- **Formal Analysis:** Petri Nets are a famous process modeling technique with precise definitions to support the dominant factors of workflow management [4]. Because of its powerful analysis ability, it can be used to verify the workflow.
- **Computerized Support:** Petri Nets are process oriented and are an explicit representation tool that can be run using a software tool. In fact, in the development of our collaborative learning model, we simulated user behavior using a Petri Net tool.

3. Mapping Petri Nets onto SCORM Sequences

In this section, we introduce our Distance Learning Color Petri Net (DCPN). We apply the DCPN to construct various sequence controls in the SCORM specification. We also show examples using a sequencing sub-module. We

implemented a prototype based on DCPN, which is shown below.

3.1 Distance-learning Color Petri Net (DCPN)

We adopt the traditional Petri Net, with refinement and additions. Our DCPN model is defined as a directed graph $PN = (P, T, F, W, M_0)$; where

$P = \{p_1, p_2, \dots, p_m\} \cup \{cp_1, cp_2, \dots, cp_n\}$ is a finite set of places that consist of two subset, ordinary place subset (circle) and control place subset (double circle), respectively. This is different from an ordinary Petri Net where there is only one type of place.

$T = \{t_1, t_2, \dots, t_K\}$ is a finite set of transitions that are drawn using bars.

$F : \{P * T\} \cup \{T * P\}$ is a finite set of arcs representing the flow relations .

$W : F \rightarrow I$ is a weight function, $I = \{1, 2, \dots\}$ representing set of nonnegative integers. The function W can be extended to incorporate a weight k . Therefore,

$W_k : F \xrightarrow{k} I$ represents a set of k parallel arcs with the same source and destination.

$M_0 : P \rightarrow \{IC_1, IC_2, IC_3, \dots\}$ is the initial marking (represented using dots), that assigns color tokens to each place in the net, $IC = \{n_1, n_2, n_3, \dots\}$, is a nonnegative set of integers representing numbers of color tokens.

Generally speaking, there are some typical interpretations of places such as preconditions, input data, conditions, and resources needed. As indicated above, the ordinary places represent learning material including lessons, assessments, or courses; the other type of place is a control place that assists in model management. T is a finite set of transitions that are events, computation steps, or state change operators. According to [6], an arc directs the information flow outgoing from input place as either a point of departure or a temporary pause. The model's dynamic behavior is simulated by the firing rules. A transition will fire if the number of color tokens in its input places is equal to or greater than the weight of each color on the input arcs. If the transition fires, the tokens in the input places will move to the output places according to the weight of each output arc.

3.2 Sequence Construction

The learning behaviors from different learners will lead to different learning paths. This is stressed in the SCORM 2004 sequencing definition model. In order to provide the proof of a feasibility study of sequences and a preview of SCORM sequence features, ADL released the Photoshop sequence example [7]. It illustrates sequence in several ways based on the same course content using various instructional strategies, such as linear, linear choice, and knowledge paces, etc., using original content to create textual and visual representations of learning design and sequencing frameworks. Finally, the design maps these

frameworks to an activity tree for sequencing. To probe into the patterns of instructional strategies, we can get the categories of possible routing constructs as follows:

Routing Constructs – Flow and Choice. The routing construct is the atomic structure being able to control the sequencing behavior of a cluster.

Possible Constructs – Skip, Limit Condition, Suspend and Roll up. Skip is one of actions of set [if [condition_set] then [action/behavior]] in the sequence rule; Limit Condition is based on the Tracking Model of SCORM. Limited conditions override Sequencing Rules. Suspend indicates when the learner want to exit the course temporarily. Roll Up is the process of evaluating the Objective and Attempt Progress data for a set of child activities of a parent activity.

The above concepts follow the specification of SCORM 2004, which has good examples in [7].

3.3 Map DCPN onto SCORM Sequence

The DCPN model is able to accomplish the SCORM routing constructs easily, including flow and choice. Flow displays a straight linear learning path. It ensures that the learner progresses through the content aggregation in a pre-determined order. Choice allows the learner to jump and select other lessons in an arbitrary order. In Figure 1, and Figure 2, we illustrate these routing constructs of flow and choice, respectively. In Figure 2, the learner can choose “Lesson1” or “Lesson 2” by firing transition 1 or transition 2, after he has completed “Introduction”.

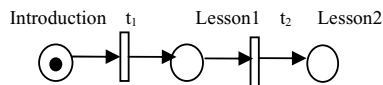


Figure 1. Flow Construct.

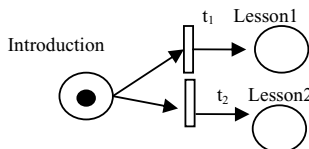


Figure 2. Choice Construct.

We identify other SCORM constructs as follows. Skip is the action when the learner wants to ignore some learning material temporarily in a linear structure. We add the control place with an initial token and use the choice structure to allow a learner to either go through the content in a linear manner, or skip one of the learning materials. In Figure 3, the double circle represents the control place. If transition t4 is fired then the learner will skip “Lesson2” and learn “Lesson3” immediately. The reason of the input arc of t4 to be two-way is to preserve the learning opportunity of skipped learning material (i.e., Lesson2). After a learner learned Lesson 1, Lesson 3 and Lesson 4, the learner has an opportunity to study the skipped Lesson 2.

Limit Condition describes a condition under which an activity is not allowed to deliver. Since SCORM does not requires the evaluation of any time-based limit conditions, our model focuses on the maximum number of attempts for the activity. By the number of token in the control place, we limit the times that the learning material can be read. In Figure 4, because each number of tokens in input places (cp1 and Lesson1) is equal to ($M(\text{Lesson1})=1$) and greater than ($M(\text{cp1})=2$) each weight of the directed arcs connecting the places to transition t1 simultaneously, transition t1 is said to be enabled. If we fire t1, the number of tokens in input places decrease ($M(\text{Lesson1})=0$, $M(\text{cp1})=1$) and the number of token in output place increases ($M(\text{Lesson2})=1$). In other words, we utilize the number of tokens in a control place to limit the number of times the transition can be fired.

Suspend describes the situation where the learner needs to terminate learning activities temporarily, in which case the LMS should record the break point in order to restart learning. In our model, one type of arc can represent Suspend by labeling the arc with “s”. This can distinguish whether the source place of an outgoing arc terminates or is temporarily paused when the token leaves that place.

The above concepts can be mapped onto the features of Petri Nets to construct a framework. An entire activity tree may be replaced by a single place or transition for modeling at a more abstract level, or places and transitions may be replaced by subnets to provide more detailed modeling. In our model, we make use of the token number and the weight of an arc to determine whether the Roll Up condition is satisfied.

We illustrate using the Exam model in Figure 5. When transition t1 is fired, the token is moved from Question1 to cp1 to judge whether the answer is correct. If so, the net fires t5 and moves a token from cp1 to cpcorrect; otherwise, the net fires t4 and move token from cp1 to cpererror. After all questions are finished, the number of tokens in $\text{cp}_{\text{correct}}$ and cp_{error} represents the total number of correct and error answers respectively. And, we use the weight of the arc connecting to t10 to set the filter of whether the exam is passed. In order to calculate the sum of the correct questions, we didn't divide the module into individual question. In the example, we simulate the real operation of an exam.

If the learner passes an exam (pretest or posttest), the module's learning objective has been satisfied and the module becomes disabled – the learner is unable to select any of its lessons. In Figure 6, we draw the Abstract Module 1 in the SCORM example. We can observe that the arc connects from Ppretest to two transitions t1 and t2. Ppretest is the abstract place that can contain Question1, Question2, and Question3 from Figure 5, and thus has a hierarchical relationship with Figure 5.

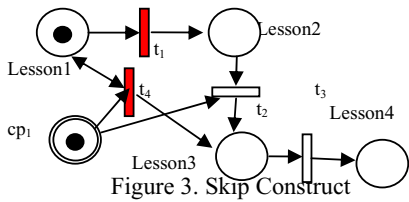


Figure 3. Skip Construct

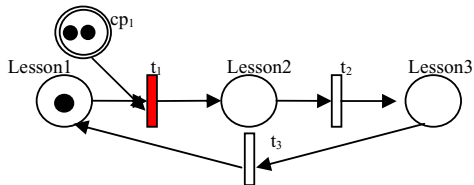


Figure 4. Limit condition (attempt=2) Construct

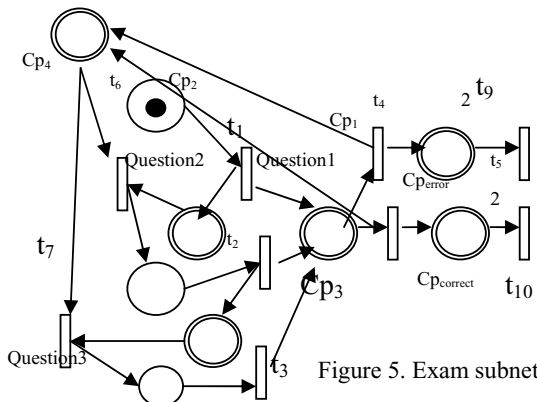


Figure 5. Exam subnet

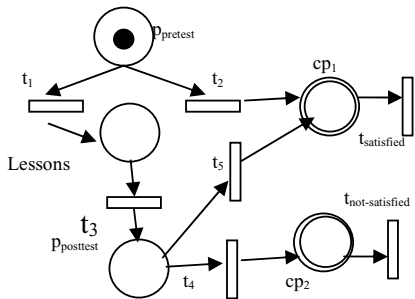


Figure 6. Abstract Module 1 subnet

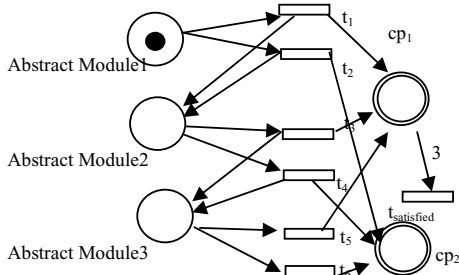


Figure 7. Modules subnet

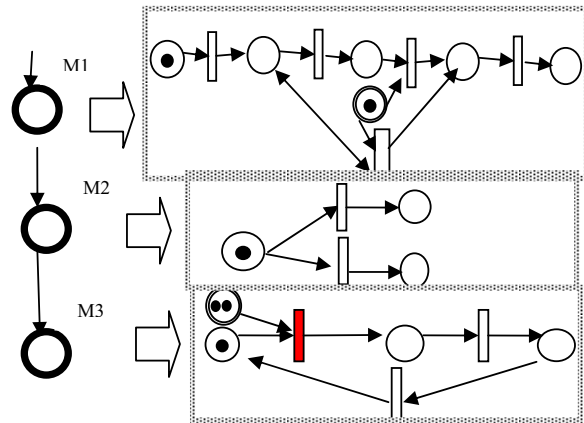





Figure 8. An example with Flow, Skip, Choice and Limit condition

The output of Ppretest is assessed by t_1 and t_2 . If the learner entered into the Abstract Model 1 and passed the Pretest, he/she can ignore the immediate Lessons and the Posttests to experience the next Module. If the learner does not pass the Pretest, he/she is directed to the Lessons, and once completed, must take the Posttest. The Posttest is not selectable by the learner. It is only encountered after 'flowing' through the Lessons.

The same operation to construct the Modules is demonstrated in Figure 7. The subnet replacements for Abstract Module 1, Abstract Module 2 and Abstract Module 3 each have output transitions that are reached depending on which module is satisfied.

We use examples to show the compositionality of the constructs. In Figure 8, the example is comprised of three major nodes. Node M1 shows the Flow with Skip construct. Node M2 shows the Choice structure. Node M3 shows the Limit condition construct with a maximum of 2 attempts.

3.4 The DCPN Modeling tool in MINE Authoring Tool

In [8, 9], we proposed a SCORM-based course content object authoring system called MINE Authoring Tool. The system provides a friendly authoring interface. Users can just drag and drop to generate a SCORM compatible and learning sequence supported exchangeable file. The authoring interface in Figure 9 has two levels of design windows (on the left side) and two preview windows (on the right side). On the first level, there are three types of Assets, Private Lecture Component , Public Lecture Component , Extracted Lecture Component , and two buttons to load or delete the Components. However, the sequence rules are too complex to trace. As a result, the system generates a new window to show the SCO sequence within DCPN according to the above fill-in actions. Figure 9 shows the SCO sequence rule 'flow.' The left side consists of

two parts. The part above shows the Petri Net graph and the part below is the browser interface for users to preview the course content.

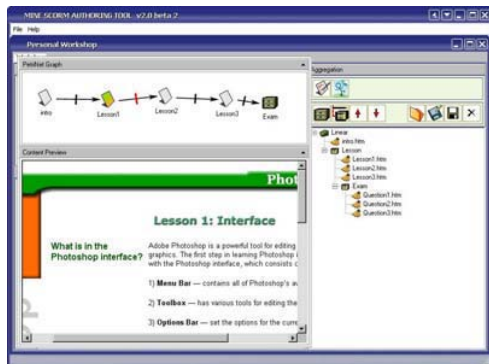


Figure 9. The sequence representation interface

We compare, in Figure 9, the structure in the right side preview window with the Petri Net graph and find that they seem to be the same structure. In fact, one is a flow sequence and the other is a choice sequence. But we cannot observe this fact from the structure in the preview window. This is why we need the interface to help the course designer and the learner to trace the course sequence easily and unequivocally.

In the literature, the above authoring tool is the first of its kind to visually enable sequences. Several content designers in local industry use the authoring tool to create content. Preliminary feedback is in general satisfactory. Suggestions received include enhancing the editor to incorporate additional media types, such as video and audio.

4. Supporting Collaborative Learning

The main contribution of this paper is in the newly proposed technology of collaborative learning in SCORM. In several studies [8, 10], collaborative learning was shown to be effective in distance education.

Our DCPN can model all the flows of collaborative learning that are key to distance learning, as we explain in the following sections.

4.1 Construct Collaborative Learning with DCPN

The most important feature of collaborative learning (CL) is to form a group with heterogeneity such as having different backgrounds, various learning paths, and diverse instruction styles. According to this feature, collaborative considerations should be addressed before we design tools for collaborative learning. One issue is what capability is necessary and the other is whether the number of groups is satisfied. The solution is DCPN utilizing token color through an exam to verify the learner capability and the token number to validate the specified number of groups. The learner should take an

exam to tell the system what role he can play before entering into collaborative learning. As shown in Figure 10, taking the Exam and dividing into two categories (black token and white token), p_3 collects the results and waits until the condition is satisfied. That is, if there are two black tokens and one white token, the transition can be fired and then three learners make up a team to enter into the CL mode.

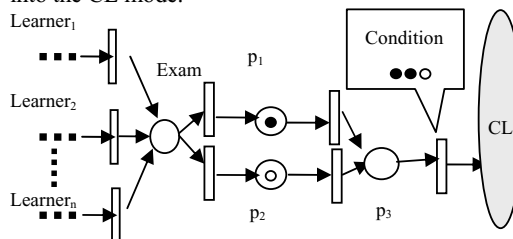


Figure 10. Conditions before collaborative learning mode

After entering CL mode, a team member has to learn some knowledge or skill before working together to solve a problem as their mission. In our model (shown in Figure 11), the “s” above the arc represents the LMS break point of suspending the lesson to enter into chartroom mode to facilitate the learner getting assistance from other partners. When the learner gets the necessary background information, he/she can go back to the break-point and keep going. In order to confirm that the learner not only achieved the global mission but also the personal mission of a collaborative learning, we identify individual accountability using places.

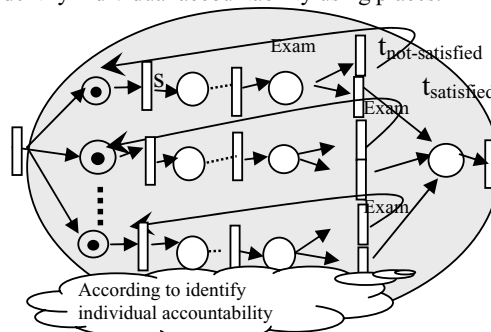


Figure 11. CL Construct

4.2 Simplified DCPN graph to support collaborative learning

In this section, we will introduce the simplified DCPN for SCORM as the learning activity tree. The learning activity tree is a tree structure showing the learner's status (e.g., pass, fail, completed, not completed, satisfied, not satisfied, attempted, or not attempted). It is important for the learners to know their learning status and learning history. Learners can know their failed courses, their successful courses, and the course structure. A learning activity tree is like a driver's map. Figure 12 shows a

learning activity tree example. This is a linear structure, having three modules with 9 lessons and one exam. We simplify the DCPN and define some icons as follows. There are some sequencing behaviors which show simple learning sequence. ↓ means the learner can only go

forward. ↕ means the learner can go both forward and

backward. ⬆ means the learner can choose any course

node. These icons are designed to fit the requirements of a SCORM sequence specification. However, the collaboration can be achieved using a LMS, with networked computers for different users. With the support of displayed learning activity trees, learners can know the visiting course node and learning history. In collaborative learning, learners can cooperate with other learners when visiting the same course node.

In our system, the learning activity tree shows the learning statuses. ● means the visiting course node.

○ means the course nodes which have not been visited. ● means the visited course nodes. ● means

the cooperative learning course nodes. Figure 12 shows the monitor status for a learning group. The instructor can see all the learners' learning status and the cooperative learning status. The number in each course node showed the learner number. The pair (8, 11) in the cooperative learning node shows (cooperative learning number, number of learners in the cooperative learning course node).

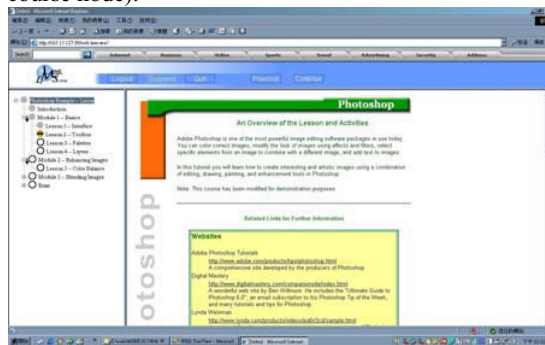


Figure 12. Collaborative Learning Monitoring (instructors)

5. Conclusion

We propose the Distance-learning Color Petri Net for the SCORM Sequence Specification using subnet oriented aggregation and firing rules. The advantage of mapping DCPN on the SCORM sequence is to make all instructional elements traceable, via a visualized activity tree. Authors using can preview the learning content after authoring, understanding the sequence and navigation behavior through the help of a visualized DCPN. We believe that collaborative

learning and learning content management are important to e-Learning. Research results in collaborative learning using SCORM are few in number. As a contribution, the newly proposed DCPN provides a starting point for the formal definition of navigation and collaboration behavior of users using SCORM compliant courseware. With this model incorporated into a visual tool, we have found that instructors are able to understand and control instruction flow easily.

Reference

- [1] IEEE P1484.1/D9, *Draft Standard for Learning Technology — Learning Technology Systems Architecture (LTSA)*, 2001, <http://ieeeltsc.org/wg1>
- [2] ADL Technical Team, *Sharable Content Object Reference Model (SCORM) Version 1.3 Working Draft 1*, Advanced Distributed Learning (ADL), October 22, 2003
- [3] IMS Global Learning Consortium, Inc., *IMS Simple Sequencing Specification*, 2002, <http://www.imsglobal.org>
- [4] Aalst, W. M. P. v. d., The application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers* 8, 1998, 21-66.
- [5] J. L. Peterson, *Petri Net Theory and the Modeling of Systems* (Englewood Cliffs, NJ:Prentice-Hall, 1981).
- [6] C.-H. Lin, "Distribution network reconfiguration for load balancing with a coloured Petri net algorithm", *Transmission and Distribution IEEE Proceedings*, Vol. 150, No. 3, May, 2003, 317-324
- [7] ADL Technical Team, *SCORM Version 1.3 Photoshop Examples Version 1.0*, Advanced Distributed Learning (ADL), November 19, 2003.
- [8] T. K. Shih, J. C.-S. Hung, W.-C. Ko, W.-C. Chang, and N. H. Lin, "Collaborative Courseware Authoring Based on SCORM Metadata," in *Proceedings of the International Conference on Computers in Education*, Hong Kong, China, Dec, 2003
- [9] T. K. Shih, W.-C. Chang and W.-C. Ko, "SCORM Sequence and Template Authoring System", *Proceeding of Resources Management Association Annual Conference*, New Orleans, Louisiana, USA, May 23-26, 2004
- [10] C. Bouras, E. Giannaka and T. Tsiatsos, "Virtual Collaboration Spaces: The EVE Community", *The 2003 Symposium on Applications & the Internet*, Orland, Florida, January 27-31, 2003, 48-55