Workshop on Virtual Reality Proceedings - 3rd WRV, p.277-278. Gramado, Brazil. SBC, 2000.

Coordination Mechanisms for Collaborative Virtual Environments

Alberto Barbosa Raposo, Léo Pini Magalhães, Ivan L. Marques Ricarte

DCA – FEEC - UNICAMP
{alberto, leopini, ricarte}@dca.fee.unicamp.br

Abstract. *In order to be effectively used as collaborative work tools, developers of virtual environments should invest, among other aspects, in the coordination of users' activities. The goal of this work is to present coordination mechanisms that may be reused in different implementations of collaborative virtual environments (CVEs).*

1. Introduction

In CVEs, users are simultaneously present and can interact with objects and other users. Currently, the development of CVEs has been dominated by leisure activities, enabling basically navigation through virtual scenarios and communication with remote users [Frécon 98]. This kind of activity is well coordinated by the "social protocol", characterized by the absence of any coordination mechanism, trusting the participants' abilities to mediate interactions. However, activities related to cooperative work require sophisticated coordination mechanisms to avoid that participants get involved in conflicting or repetitive tasks.

This paper focuses on the coordination of activities in CVEs, defining a set of interdependencies that frequently occur among collaborative tasks and presenting coordination mechanisms for them. The idea is to separate activities from dependencies (controlled by the coordination mechanisms), enabling the use of different coordination policies in the same CVE by changing only the coordination mechanisms. Moreover, these mechanisms are generic and can be reused in other CVEs.

2. Coordination Mechanisms

Coordination is "the act of managing interdependencies between activities performed to achieve a goal" [Malone 90]. Therefore, this work started with the definition of a set of frequent interdependencies between cooperative tasks. The next step was the modeling of coordination mechanisms to guarantee that those dependencies will not be violated. The final step is the implementation of those mechanisms in CVEs. The idea is that the designer of a CVE be concerned only with the definition of tasks and their interdependencies, and not with the management of those dependencies.

Interdependencies were divided into two main classes: temporal and resource management. Temporal dependencies establish the execution order of tasks, while resource management ones deal with the distribution of resources needed to the execution of a task. A total of 20 dependencies were defined [Raposo 00].

Petri nets (PNs) were used to model the coordination mechanisms. The graphical representation of PNs is simple and offers an adequate hierarchical description to define the coordination structure in different abstraction levels. PNs, due to their support for modeling, simulation and analysis, are a powerful tool for verifying the correctness and validating the effectiveness of CVEs before their actual implementation. Using this approach, it is possible to predict the behavior of a CVE, avoiding undesired situations. A similar approach were used for computer animations [Magalhães 98].

After the environment has been tested with the PN model, it is necessary to implement the CVE by including the coordination mechanisms to control the tasks' execution. One way to do this is by a "manager program" that interacts with the environment. Blaxxun platform [Blaxxun 00], a system to create CVEs offers the resources for this implementation.

3. Example

To illustrate the use of the coordination mechanisms, consider the hypothetical situation depicted in Figure 1. The figure shows two PNs, each one representing the sequence of tasks of a specific CVE user. During the environment specification, the developer detects interdependencies among tasks. For example, task3B must occur during the execution of task2B and task5A shares a resource with task4B (mutual exclusion).

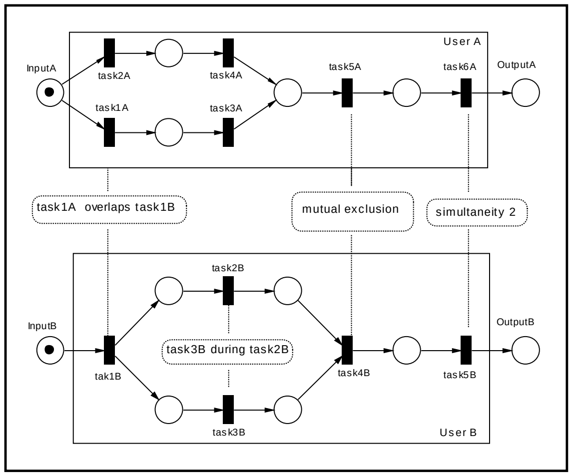

Figure 1. Tasks and interdependencies in a hypothetical CVE.

After the modeling phase, the PN model, with coordination mechanisms included, is tested to verify its correctness. In Figure 1, if user A follows, for example, the path that executes task2A instead of task1A, user B will be blocked in the execution of task1B, which depends on task1A. This indicates that the model should be revised.

When the model is approved, it should be "translated" to the CVE. This is achieved by indicating how tasks are mapped to the CVE (i.e., which events are associated to the tasks) and including the coordination mechanisms of a predefined library.

4. Conclusion

The coordination of activities in CVEs is a problem that should be addressed to facilitate the use of this kind of environment for the realization of tasks that cannot be controlled by the social protocol. The separation between activities and dependencies and the use of coordination mechanisms are steps towards this goal.

The developed coordination mechanisms also revealed generic enough to be used in a wide range of collaborative systems, including workflow systems [Raposo 00].

References

Blaxxun Interactive, Inc. (2000) "Blaxxun Community Platform 4", http://www.blaxxun.com.

Frécon, E. and Nöu, A. A. (1998) "Building Distributed Virtual Environments to Support Cooperative Work", *Proc. VRST'98*, p. 105-119.

Magalhães, L. P., Raposo, A. B. and Ricarte, I. L. M. (1998) "Animation Modeling with Petri Nets", *Computer & Graphics*, 22 (6): 735-743.

Malone, T. W. and Crowston, K. (1990) "What is Coordination Theory and How Can It Help Design Cooperative Work Systems?", *Proc. CSCW'90*, p. 357-370.

Raposo, A. B., Magalhães, L. P. and Ricarte, I. L. M. (2000) "Petri Nets Based Coordination Mechanisms for Multi-Workflow Environments", accepted for the *Int. J. of Computer Systems Science & Engineering*.