9 Multiagent Communication for e-Business using Tuple Spaces

H.F. Li, T. Radhakrishnan, and Y. Zhang

Department of Computer Science and Software Engineering
Concordia University, Montreal, Canada H3G 1M8

9.1 Introduction

9.1.1 Motivation

The growth in Web-based applications, distributed computing, and agent-based software technologies has created abundant interest in various aspects of e-business. One such application supported by these technologies is e-commerce in which online transactions between a buyer and a seller are supported in various stages of their trading (Gutman et al., 1998). E-commerce application systems have to deal with a large number of interacting autonomous tasks using heterogeneous information sources. These interactions have a need to be well coordinated and coordination requires efficient communication among the entities. The dynamic and complex nature of e-commerce requires a flexible technological infrastructure to support business processes more easily and effectively.

Agent-based software technology has become the subject of much research in a wide range of fields, especially in distributed system design. Multiagent systems are often used in a dynamic environment with autonomous problem solving entities cooperating and coordinating with each other. A typical software agent has the characteristics of autonomy, reactivity, proactivity and sociality, or structured interactions with other software agents. This makes the multiagent systems a natural candidate choice for implementing e-commerce applications.

Business processes in e-commerce may be considered as a kind of coordinated multiagent system in which the software agents perform various market activities under dynamic partnerships. With the increasing complexity of these applications, we need programming models to deal with the coordination of a large number of concurrently active entities. Thus, an infrastructure for coordination is needed to meet the complex and dynamic requirements. Tuple space that supports inter-agents coordination is an attractive solution.

For this purpose, our research focuses on tuple space-based agent coordination and tuple space-based agent programming framework (TSAF) that effectively supports building agent applications in e-commerce. In the rest of this chapter, we discuss multiagent interactions through tuple space, presenting how tuple space facilitates the dynamic couplings among agents. We also present an

agent programming model called TSAF, which provides not only agent architecture in abstraction, but also an easy-to-use programming environment for implementing the resulting design. The incorporation of role models derived from object-oriented methodologies into the design of agent behaviors supports the analysis and design of multiagent systems from the perspective of agent-oriented software engineering. In the implementation level, TSAF also supports tuple space-based agent coupling mechanism. The benefits of this framework are illustrated through a case study in e-commerce.

9.1.2 Agent Coordination in Multiagent System

Agents in a multiagent system have a general need to communicate amongst themselves, coordinate their activities, and negotiate once they find themselves in conflict. Coordination models for multiagent systems can be broadly classified into direct coordination model and indirect coordination model. Direct coordination means that agents explicitly initiate a communication via message passing and explicitly name the involved partners in their messages. In indirect coordination, agents interact via a shared space, like blackboard, where messages are posted and retrieved. A blackboard approach uncouples agent-to-agent interactions in time and space. This suits many application scenarios where agents do not know or care about the address of the collaborators. The tuple space-based coordination model promotes dynamic information sharing, so that information is available to any intended agent and every authorized agent can modify the information on the tuple space. The tuple spaces free the designer from the burden of keeping track of explicit or at least implicit addressing knowledge in agent couplings. Finally, the "reactive tuples" [1] of tuple space can support event-driven coordination among agents by triggering the reactions associated with such tuples. These reactions are normally defined based on the different roles that an agent plays. In addition, reactions can adapt the semantics of the interactions to the specific agent environment, thus simplifying the agent programming.

9.2 Computation and Tuple Spaces

A typical approach in designing multiagent systems is the use of "role models" (Bauer, 2001) [2] that capture the different functionalities in the underlying application. The roles are assigned to different agents. An agent can be assigned multiple roles so that it can cooperate with other agents to complete different missions simultaneously or at different times in its lifeline. Cooperation requires interactions. The traditional form of interaction is through message communication. An agent sends mission-specific messages to other agents involved in the same mission at "opportune times" in order to complete the common mission properly. When viewed globally, a mission involving multiple agents can be represented by the set of interaction protocols executed by these agents together. Generic agent