Chapter XVI
Software Agent Technology for Supporting Ad Hoc Virtual Enterprises

Jarogniew Rykowski
The Poznan University of Economics, Poland

ABSTRACT

This chapter introduces a new idea of using software agents for supporting ad hoc virtual enterprises and similar forms of temporal business-to-business collaboration. It seems that current information and telecommunication technologies, based on information interchange and local data processing, are not flexible enough to deal with modern business requirements, especially dynamic and temporal business relations, heterogeneity of hardware, software and communication means, and data complexity. The proposed approach differs in the distribution of both data and programs for data treatment at-the-place and just-in-time. The distributed and remotely executed programs, software agents, are autonomous entities, targeted on obtaining preprogrammed goals, and working in the name and under the authority of their owners. The authors hope that the proposed techniques for agent preparation, distribution, and execution make the whole system safe and secure, providing an efficient environment for a wide spectrum of temporal and ad hoc business collaboration.

INTRODUCTION

The idea of a virtual enterprise (VE), a value-added business built upon different, distributed, autonomous units, resulted in the rapid evolution of traditional business models. As the technical possibilities grow, stable business units are becoming less and less geographically- and timely-restricted, and relations among these units are more and more dynamic and case-oriented. It looks like the VE evolution is going to change traditional enterprises into a set of autonomous business units, able to establish a virtual enterprise to achieve given business goals at a given place and time with minimum efforts and costs, and maximum profits.

Imagine such pool of autonomous enterprises, both real and virtual, ready to be in a business.

The enterprises are heterogeneous, taking into account both their internal organization, and computer/telecommunication (IT) infrastructure. Suddenly, a business opportunity appears, for example, a possibility to organize a jubilee of a famous pianist. The cooperation is scheduled for two to three weeks only, and the coordinated activities, for example, a philharmonic concert and an official reception in a palace, are occasional, as they will probably never happen in the future. Moreover, the situation is changing all the time; there are new limitations and requirements coming that cannot be identified from the beginning, for example, a need for transportation for some disabled guests. In the future, another jubilee, for example, for a Nobel prizewinner, would require quite different activities, such as the organization of a scientific congress. Even if at the first view the jubilees are similar, the business activities and relations are completely different, and the business partners to be involved cannot be determined in advance.

Current, fixed, and costly IT technologies are not well-suited to deal with such temporal and evolving VEs. Existing proposals for building a virtual enterprise, mainly distributed applications based on such technologies as CORBA (2002) and Voyager (SOA platform, 2005), multi-databases (Hurson, Bright, & Pakzad, 1993), Web services (2002), and Semantic Web (DAML, 2006), are built upon two basic assumptions: (1) The relations between different VE units are quite stable and long-lasting; and (2) the client-server model is a dominant way of interaction among VE units. As a consequence, each unit implements a set of well-defined services (interfaces, programs, etc.) to be used at request by other business parties. This approach seriously limits an efficient implementation of evolving relations among cooperating units for at least two reasons. First, the information flow must be initialized by the client party in the online mode (information polling). Due to this poll-only mode, automatic detecting of server-side information changes and "pushing" them to clients is hard to maintain. Second, it is quite difficult to adopt services and interfaces of a single unit to the specificity of another unit; the client must be ready to adopt itself to the server-side standards. Server-side parameterization is usually limited, due to both technical and economical reasons.

To solve the problem of supporting ad hoc VEs (ahVE), we propose autonomous software agents (Caglayan & Harrison, 1997; Franklin & Graesser, 1996; Nwana, 1996; Wooldridge & Jennings, 1995). The role of an agent is twofold. First, an agent is used as an information broker and wrapper, to adjust data format (both syntax and semantics) to the specificity of communicating units. The brokerage/wrapping algorithm may be programmed either by one or by both parties. Second, an agent may act as a monitor and asynchronous notifier about important data changes. What is "important" is programmed in agent code and variables. The agents may interact with local IT systems of the VE units, as well as with other agents, and with humans. The agents are executed in the scope of agent computing environment (ACE) framework, being a set of agent servers (Rykowski, 2003a). An agent server may be located at any host, including dedicated hosts belonging to unit's local area networks. The agents may be moved among agent servers (Rykowski, 2005a), and according to the situation each agent may be executed at client-side (i.e., in unit's local area network), at server-side (i.e., in a network of another unit), or at an external network host (i.e., on a separate host outside units' networks).

How do the proposed agents differ from (for example) Web services and other competitive technologies? There are three main advantages to using the agents in comparison with the classical approaches: substantially reduced amount of work needed to establish a connection between two cooperating VE units, unrestricted individualization of relations among units, and a possibility of off-line, server-side monitoring of critical information changes.