

Multi-Agent Cooperation, Dynamic Workflow and XML for E-Commerce Automation

Qiming Chen, Meichun Hsu, Umeshwar Dayal, Martin Griss

HP Labs

1501 Page Mill Road, MS 1U4
Palo Alto, California, CA 94303, USA
+1-650-857-3060

{qchen,mhsu,dayal,griss}@hpl.hp.com

This work focuses on providing a multi-agent cooperation infrastructure to support E-Commerce automation. Before discussing our solutions, we first present a typical E-Commerce scenario and the requirements for E-Commerce automation.

E-Commerce applications operate in a **dynamic and distributed environment**, dealing with a large number of heterogeneous information sources with *evolving contents* and *dynamic availability*. They typically rely on distributed and autonomous tasks for information search, fusion, extraction and processing, without centralized control.

E-Commerce typically involves the following activities: identifying requirements; brokering products; brokering vendors; negotiating deals; or making purchase and payment transactions. Today, these activities are initiated and executed by humans. In the future, we see them being conducted by software agents.

Software agents are personalized, continuously running and semi-autonomous, driven by a set of beliefs, desires and intentions (BDI). They can be used to *mediate* users and servers to automate a number of the most time-consuming tasks in E-Commerce, with enhanced parallelism. Agents can also be used for *business intelligence*, such as discovering patterns (e.g. shopping behavior patterns or service providing patterns) and react to pattern changes. For example, suppose the sales of VCRs had been strongly associated with the sales of TVs, but this association has recently weakened as TV buyers turn to buying DVDs instead of VCRs. Such a change in the association helps to explain or predict the slow down of VCR sales. Moreover, agents can selectively preserve data and themselves become *dynamic information sources*.

E-Commerce is also a **plug and play environment**. Business processes and agent cooperation are embedded in each other. Services need to be created dynamically on demand. Business partnerships (e.g. between suppliers, resellers, brokers, and customers) need to be created dynamically and maintained only for the required duration such as a single transaction.

The dynamic nature of E-Commerce requires multi-agent cooperation to be based on **dynamic ontology**. By dynamic ontology we mean that the concepts, rules and facts underlying agent interaction, are different from domain to domain, and vary from time to time. In order to automate agent cooperation, it is necessary to provide a standard format for encoding messages with meaningful structure and semantics, as well as domain ontology that agents can readily exchange and interpret. This format should be common for agent communication as well as for E-Commerce data exchange in general. The extensible markup language, XML, is becoming the standard for data interchange on the Web. We use XML for the above purpose.

Business processes, or workflows, may be considered as a kind of multi-agent cooperation, in the sense that software agents may be used to perform tasks of business processes, and workflow may be used to orchestrate or control the interactions between agents. We envisage the need for dynamically plugging them into each other.

1. DYNAMIC AGENTS

At HP labs, we have developed a Java based **dynamic agent** infrastructure for E-Commerce which supports **dynamic behavior modification** of agents, a significant difference from other agent platforms. A dynamic agent does not have a fixed set of predefined functions, but instead, it **carries application-specific actions**, which can be loaded and modified on the fly.

A dynamic-agent has a *fixed part* and a *changeable part*. As its fixed part, a dynamic-agent is provided with light-weight, built-in management facilities for distributed communication, object storage and resource management. A dynamic agent is capable of carrying data, knowledge and **programs** as objects, and executing the programs. The data and programs carried by a dynamic agent form its changeable part. All newly created agents are the same; their application-specific behaviors are gained and modified by dynamically loading Java classes representing data, knowledge and application programs[1].

These capabilities allow a dynamic agent to adjust its capabilities and play different roles to accommodate changes in the environment and requirements. Through messaging, dynamic agents can expose their knowledge, abilities and intentions, present requests and exchange objects. They can move to the appropriate location for high-bandwidth conversation. They can also manage their own resources across actions. Such an infrastructure supports dynamic service

construction, modification and movement, and allows a dynamic agent to participate in multiple applications and dynamically formed partnerships. With these features, dynamic agents fit well into the dynamic E-Commerce environment.

A multi-agent cooperation infrastructure is developed for E-Commerce automation, where dynamic agents perform various market activities, cooperating through exchanging data as well as programs, switching roles and forming *dynamic partnership* that exists only when needed. For example, the agents reselling products, the agents supplying products and the agents providing brokering services for negotiating service terms, etc, may form dynamic partnership for a specific business application. In this way, dynamic agents cooperatively support *plug-and-play commerce*, allowing businesses to be built on one another's services.

2. XML MESSAGING

Autonomous agents cooperate by sending messages and using concepts from a domain ontology. A standard message format with meaningful structure and semantics, and a mechanism for agents to exchange ontologies and message interpreters, have become key issues. Furthermore, the message format should be accepted not only by the agent research community, but also by all information providers.

Dynamic agents send and receive information through XML encoded messages. We use a KQML/FIPA ACL-like format, encoded in XML.

In fact, an XML document is an **information container** for reusable and customizable components, which can be used by any receiving agent. This is the foundation for document-driven agent cooperation. By making Web accessible to agents with XML, the need for customer interfaces for each consumer and supplier will be eliminated. Agents may use XML format to explain their BDI, explaining new performatives by existing, mutually understood ones. Based on the commonly agreed tags, agents may use different style DTDs to fit the taste of the business units they mediate. Further, a dynamic agent can carry an XML front-end to a database for data exchange, where both queries and answers are XML encoded.

The power of XML, the role of XML in E-Commerce, and even the use of XML for agent communication, have been recognized. Although XML is well structured for encoding semantically meaningful information, it must be based on an ontology. As ontology varies from domain to domain, and dynamic for dynamically formed domains, The more significant issue is to exchange the semantics of domain models, and interpret messages differently in different problem domains.

Generally speaking, domain ontology provides a set of concepts, or meta-data, that can be queried, advertised and used to control the behavior of agent cooperation. These concepts can be marked using XML tags, and then a set of commonly agreed tags, underlie message interpretation. The structures and the semantics of the documents used in a particular problem domain are represented by the corresponding DTDs and interpreters.

We use different languages, all in XML format, for different problem domains, such as product ordering, market analysis,

etc. Accordingly, we use an individual interpreter for each language. Dynamic agents can exchange those DTDs together with documents, and exchange those interpreters as programming objects, in order to understand each other in communication.

These approaches allow us to provide a unified application carrier architecture, a unified agent communication mechanism, a unified way of data flow, control flow and even program flow, but flexible application switching capability, for supporting E-Commerce.

3. PLUG-IN WORKFLOW SUPPORT

Workflow systems provide flow control for business process automation. Business processes often involve multilevel collaborative and transactional tasks. Each task represents a logical piece of work that contributes to a process. A task at leaf-level is performed by a *role*. A role is filled at run-time with a user or a program. A process and its tasks are handled at separate layers. At the process layer, centralized coordination is supported; and at the task layer, location distribution, platform heterogeneity and control autonomy, are allowed .

Business processes may be considered as a kind of multi-agent cooperation, in the sense that a software agent can be used to fill a role for performing a task in a workflow, and workflow can be used to orchestrate or control the interactions between agents. However, many related activities in E-Commerce automation do not form synchronized, traditional workflow, but requires more dynamic agent cooperation. In order to combine the strength of workflow and agent cooperation for supporting E-Commerce, it is necessary to understand their relationship and difference.

We have developed the mechanisms for **plugging workflow in agent cooperation**, and **plugging agent cooperation in tasks** of business processes, are introduced. In particular, **dynamic workflow service provisioning** is supported, allowing workflow servers to be built on the fly.

E-Commerce is a dynamic, distributed and a plug and play environment for which we expect software agent based technologies to become increasingly important. However, since agents with static capability cannot dynamically load new functions, cannot change their predefined behavior, and cannot exchange programs with others, they are unable to switch roles, to participate in multiple applications, or to be involved in dynamically formed partnerships. Therefore, static agent frameworks are not really suitable for the highly dynamic E-Commerce applications. In this paper we presented our solutions for E-Commerce automation using a dynamic agent infrastructure. We plan to further explore a conceptual business model of plug and play E-Commerce.

4. REFERENCES

- [1] Q. Chen, P. Chundi, Umesh Dayal, M. Hsu, "Dynamic-Agents", International Journal on Cooperative Information Systems, 1999, USA.

