Appeared in IJCAI '95, Workshop on Intelligent Manufacturing
N. Sadeh (editor), Montreal, 1995.

The Architecture of an Agent Based Infrastructure for Agile Manufacturing

- Extended abstract -

Mihai Barbuceanu and Mark S. Fox
{mihai, msf}@ie.utoronto.ca
Enterprise Integration Laboratory, Department of Industrial Engineering,
University of Toronto, 4 Taddle Creek Rd, M5S 1A4, Toronto, Ontario, Canada.
Phone: 1-416-978-0910, 1-416-978-6823

1.0 Introduction

The supply chain of a manufacturing enterprise is a world-wide network of suppliers, factories, warehouses, distribution centres and retailers through which raw materials are acquired, transformed and delivered to customers. Supply chain management is the strategic, tactical and operational level decision making that optimizes supply chain performance. The strategic level defines the supply chain network, i.e. the selection of suppliers, transportation routes, manufacturing facilities, production levels, warehouses, etc. The tactical level plans and schedules the supply chain to meet actual demand. The operational level executes plans. Tactical and operational level decision making functions are distributed across the supply chain.

In order to optimize performance, supply chain functions must operate in a coordinated manner. But the dynamics of the enterprise and the market make this difficult: exchange rates change overnight, materials do not arrive on time, production facilities fail, workers are ill, customers change or cancel orders, etc. causing deviations from plan. In some cases, these events may be dealt with locally, i.e. they lie within the scope of a supply chain function. In other cases, the problem can not be "locally contained" and modifications across many functions are required. Consequently, the supply chain management system must coordinate the revision of plans/schedules/decisions across supply chain functions. The agility with which the supply chain is managed at the tactical and operational levels in order to enable timely dissemination of information, accurate coordination of decisions and management of actions among people and systems, is what will ultimately determine the efficient, coordinated achievement of enterprise goals.

Our research addresses coordination problems at the tactical and operational levels. It organizes the supply chain as a network of cooperating, intelligent agents, each performing one or more supply chain functions, and each coordinating their actions with other agents. The focus of our research is in supporting the construction of supply chain intelligent agent systems in a manner that guarantees that agents use the best collaboration/coordination mechanisms available with minimal programming effort on the developers' side. We are achieving this goal:

- by developing communication and coordination theories and tools allowing agents to cooperatively manage change and cooperatively reason to solve problems,
- developing ontologies that semantically unify agent communication,
- developing intelligent information infrastructures that keep agents consistently aware of relevant information,
- developing constraint-based reasoning as the fundamental agent reasoning technology,
- packaging the above theories into agent development tools that ensure that agents are able to reuse standardized coordination and reasoning mechanisms, relieving developers from the tedious process of implementing agents from scratch.

The TOVE virtual enterprise [13] is our unified testbed used by the agents we built for the major supply chain functions: Logistics, Transportation Management, Order Acquisition, Resource Management, Scheduling and Dispatching. These are *functional agents* - in charge with carrying out well defined organizational functions. They rely on ontologies for activity, state, time, resources, cost, quality and organization as a common vocabulary for communication and use constraint satisfaction reasoning to perform their tasks. We have also built *information agents* - information brokers and mediators - that keep functional agents aware of any relevant events and changes by automatically distributing information and managing the consistency and evolution of enterprise information.

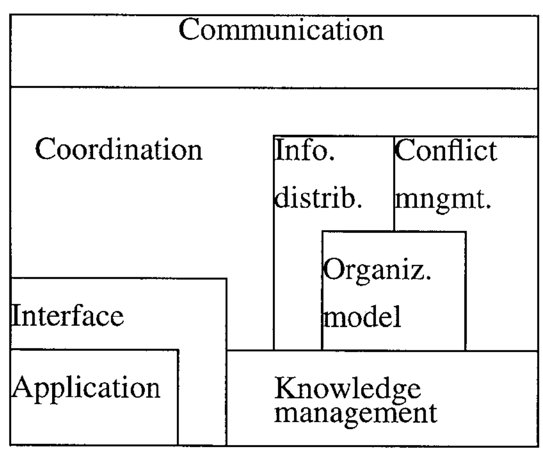

FIGURE 1. Generic agent architecture

2.0 An Agent Development Tool

We are now building agent development tools integrated in a Generic Agent Shell that will provide "off-the-shelf" but customizable services for communication, coordination, reasoning and problem-solving, relieving developers from the tedious process of implementing agents from scratch.

The GenericAgent Shell provides several layers of reusable services and languages. They are concerned with agent communication, specification of coordination mechanisms, services for conflict management and information distribution, reasoning and integration of legacy application programs (figure 1). The glue that keeps all layers together is a common knowledge and data management system on top of which these layers are built. Clearly distinguishing between an agent's social know-how - consisting of agent communication services, coordination mechanisms, information distribution services and others - and its domain level problem solving means that the approach is both flexible and open, imposing few constraints on the application designer and yet providing many useful facilities. Purpose built application programs can make use of this agent architecture to enhance their problem solving and to improve their robustness through coordination with other agent based applications. Pre-existing (legacy) application programs can also be incorporated with little adaption and can experience similar benefits. This latter point is important because in many cases developing the entire application afresh would be considered too expensive or too large a change away from proven technology. Here is brief review of the current layers of our generic agent architecture:

1. *Agent Communication Language.* This is the language agents use to communicate. The language consists of speech acts, describing the communicative actions carried out, and an embedded content layer that describes the actual information communicated. We have adopted as the ACL the KQML/KIF language produced as a result of the ARPA effort on knowledge sharing. KQML/KIF supports a declarative approach to knowledge communication, as opposed to procedural approaches (e.g. TeleScript). A major advantage of the declarative approach is that it permits the explicit and declarative specification of various coordination mechanisms agents use. These coordination mechanisms can be modeled as shared conventions governing the exchanged speech acts during coordinated action. These shared conventions allow participating agents to understand the intentions of the other agents and thus to coordinate effectively in solving the common problem.

2. *Coordination models and language.* Coordination models - shared conventions about exchanged messages during cooperative action - are described in a special purpose coordination language. We are building on the assumption that coordination models can be generically defined in terms of rules about cooperation and situation assessment that are applicable in many industrial applications.

The purpose of the coordination language is to allow the explicit, declarative specification of coordination models. Our implemented coordination language, named COOL [2], allows the representation of coordination models in terms of structured conversations with a finite state semantics, conversation rules managing the state transition within conversations, continuation rules managing the selection of the next conversation to work on and recovery rules managing communication problems (delays, lost messages, etc.). In this language we are building coordination libraries that are imported and extended by applications. The implemented system provides a graphic user interface allowing users to manipulate and animate visual representations of coordination models and their execution.

3. *Information distribution.* The information distribution service is a generic service whose major purpose is to be able to distribute (voluntary or at request) information of

interest to other agents, in a manner that relies on the content of the information. This essentially requires deductive information processing capabilities. Another capability of the service is performing multi-agent belief revision functions. When any knowledge or information used by the agent is invalidated, the service determines if and what communicated information is invalidated and sends denial messages to the recipients of that information [6,7,9].

4. *Organizational model.* Agents can not operate autonomously unless they have a model of the organization(s) they are part of. This model tells the agent what other agents exist, which are the roles they have, what goals agents pursue, what sort of communication can take place amongst them, etc [14]. The organization model contains a model of the agent itself, representing the agent's roles, goals and capabilities.

5. *Conflict management.* The conflict management service enables the agent to make decisions when confronted with contradictory information derived or received from other agents. We assume that in real enterprises contradictory information occurs quite often and being able to cope with it increases the robustness of agents. We have designed a new approach to conflict management, based on organization modeling and on the concepts of agent credibility and information retraction costs [8].

6. *The knowledge management system.* The mental state of agents, consisting of generic concepts and actual beliefs about the world, is maintained in a powerful description logic representation system, extended with temporal reasoning [11] and several other processing mechanisms.

7. *Generic interface to applications.* An agent may control a number of non-agent applications (legacy or purpose built). For this purpose it must provide an interface allowing data, parameters and control specifications to be transmitted to/from the applications. We are assuming that even if the integrated applications will be diverse, the interface need not be ad-hoc. In consequence, we have devised ways to construct a systematic interface able to accommodate diversity at one end and consistent manipulation mechanisms at the other.

3.0 Enterprise Information Architecture

The Enterprise Information Architecture (EIA) consists of a distributed environment of Information Agents that support functional agents with services for relevancy based information distribution and consistency management. Functional agents register their capabilities and needs with an Information Agent. Information Agents receive information from functional agents, reason about its relevance to other agents and distribute it to those agents for which they consider it relevant, and in the form that is easiest to understand. If distributed information ever becomes inconsistent, the Information Agent mediator will alert all receivers. If functional agents supply contradictory information, the mediator will apply strategies for solving the conflict and reinstall consistency. Communication takes place using the KQML / KIF.

We have fully designed and implemented an Information Agent mediator based on description logic representation technology, with capabilities for content and relevance-based information routing, conflict management based on our own "credibility-utility"-conflict removal model and information translation to bridge the gap caused by different ontologies employed by agents [8].

4.0 The Integrated Supply Chain Management System

We address coordination problems at the tactical and operational levels of the supply chain. The supply chain is organized as a network of cooperating, intelligent agents, each performing one or more supply chain functions, and each coordinating their actions with other agents. Having an agent building toolset allows the construction of agent systems in a manner that guarantees that agents use the best collaboration/coordination mechanisms available with minimal programming effort on the developers' side.

In this vision, we have functionally decomposed the supply chain and built agents for the major resulting functions: Logistics, Transportation Management, Order Acquisition, Resource Management, Scheduling and Dispatching. These are functional agents, in charge with carrying out well defined organizational functions. They rely on ontologies for activity, state, time, resources, cost, quality and organization as a common vocabulary for communication and use constraint satisfaction reasoning to perform their tasks. We also employ information agents, information brokers and mediators that keep functional agents aware of any relevant events and changes by automatically distributing information and managing the consistency and evolution of enterprise information.

Here is a brief review of the functional agents existing in the supply chain (figure 2):

Order Acquisition agent. This agent is responsible for acquiring orders from customers, negotiating with customers about prices, due dates, etc., and handling customer requests for modifying or canceling respective orders. This agent captures the order information directly from customers and communicates these orders to the logistics agent. When a customer order is changed, it is communicated to the logistics agent. When schedules violate constraints imposed by the customer (such as due date violation), the order acquisition agent participates in negotiating with the customer and the logistics agent for a feasible schedule.

Logistics agent. This agent is responsible for coordinating multiple-plants, multiple-supplier, and multiple-distribution center domain of the enterprise to achieve the best possible results in terms of final deliveries to customers. It manages the movement of products or materials across the supply chain from the supplier of raw materials to the customer of finished goods. The inputs to the logistics agent are customer orders, deviations in factory schedules which affects customer orders, and availability of resources. The outputs of the agent are customer orders (assigned to respective plants), and purchase orders for "critical" components.

Scheduling agent. This agent is responsible for scheduling and rescheduling activities in the factory, exploring hypothetical "what-if" scenarios for potential new orders, and generating schedules that are sent to the dispatching agent for execution. The inputs to the scheduling agent are requests for new orders from the logistics agent and the deviations of the current schedule from the dispatching agent. Its output is an executable schedule.

Resource agent. This agent dynamically manages the availability of resources so that the schedule can be executed. It estimates resource demand and determines resource order quantities. It is responsible for selecting suppliers while minimizing costs and maximizing delivery. The inputs to the resource agent are the schedule from the scheduler, the availability or unavailability of resources from suppliers, the arrival of resources from the factory floor, and the consumption of resources from the dispatcher. The outputs of the resource agent include the arrival of resources, the availability of resources, and the orders sent to suppliers.

Dispatching agent. This agent implements the schedule created by the scheduling agent. It monitors the factory operations and decides what to do next. Its decisions are constrained by the schedule. The inputs to the dispatching agent are the schedule from the scheduling agent, the status of the factory floor, and the availability of resources. The outputs are the deviations from the current schedule and the starting of activities.

Transportation agent. This agent is responsible for the assignment and scheduling of transportation resources in order to satisfy inter-plant movement requests specified by the Logistics Agent. It will be able to consider a variety of transportation assets and transportation routes in the construction of its schedules.

Plant Management Agent. This agent distributes work among the plant level agents, based on the orders received from Logistics. It also conducts negotiations when orders can not be fulfilled as requested or when plant level agents can not reach an understanding.

Our development of the supply chain system is stepwise. First, we have developed the local problem-solving functions of agents. Most of them now use constraint heuristic search techniques and frame based representations. Then, we have introduced KQML-based communication with user driven decision making for each functional agent The Information Agents, providing content based distribution, translation, consistency management, etc. operate autonomously from the beginning. Next, coordination scenarios were designed, prototyped and validated using our tool COOL. They allow for mixed initiative negotiations in which the system automates what it can and calls for the human decision maker when its abilities are exceeded. We are now in an experimentation phase that allows us to exercise the various features of the system.

This application is part of the TOVE Virtual Enterprise Project [14]. The TOVE Project includes two major undertakings: the development of an Enterprise Ontology, and a Multi-Agent System Testbed. The TOVE Enterprise Ontology provides a generic, reusable ontology for modelling enterprises, currently spanning knowledge of activity, state, time, causality, resources, organizations, cost and quality. The TOVE Multi-Agent System Testbed provides an environment for "execution" of the virtual enterprise as a multi agent system. The Testbed provides a model of an enterprise - a lamp manufacturing plant - and tools for browsing, visualization, simulation and deductive queries.

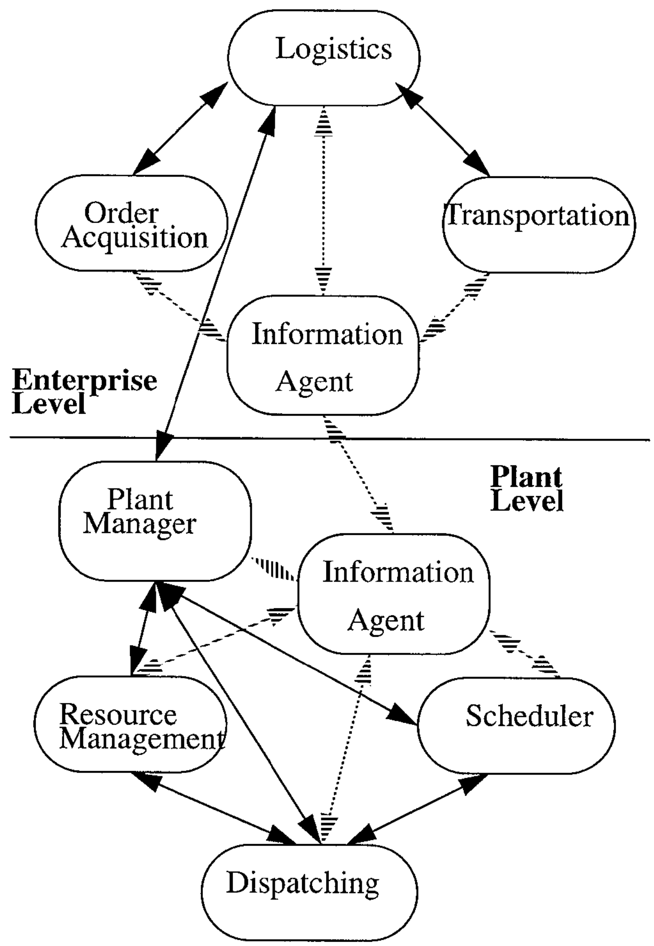

FIGURE 2. ISCM Agents.

5.0 References

1. M. Barbuceanu and M. S. Fox, The Architecture of a Generic Agent for Collaborative Enterprises, EIL working paper, nov. 1994, submitted for publication, available at http://www.ie.utoronto.ca/EIL/public/generic-agent-shell.ps

2. M. Barbuceanu and M. S. Fox, COOL - A Language for Describing Coordination Models in Multi-Agent Systems, To appear in International Conference on Multi-Agent Systems, SanFrancisco, june 12-14, 1995.

3. J.C. Beck, A Schema for Constraint Relaxation with Instantiations for Partial Constraint Satisfaction and Schedule Optimization, MSc. Thesis, University of Toronto, 1994. (also released as a technical report EIT-94-3)

4. J.C. Beck and M.S. Fox, Mediated Conflict Recovery by Constraint Relaxation, Working Notes of AAAI Workshop on Conflict Management, Seattle, United States, 1994.

5. J.C. Beck and M.S. Fox, Supply Chain Coordination via Mediated Constraint Relaxation Proceedings of the First Canadian Workshop on Distributed Artificial Intelligence, Banff, AB, May 15, 1994.

6. M. Barbuceanu and M.S.Fox, The Information Agent - An Infrastructure Agent Supporting Collaborative Enterprise Architectures, Proc. of Third Workshop onEnabling Technologies: Infrastructure for Collaborative Enterprises, Morgantown, West Virginia, IEEE Computer Society Press, 1994.

7. M. Barbuceanu and M.S.Fox, The Information Agent: An Infrastructure for Collaboration in the Integrated Enterprise, in Proc. of Cooperative Knowledge Based Systems, University of Keele, Staffordshire, UK, M. Deen (ed), DAKE Centre, Univ. of Keele, 1994.

8. M. Barbuceanu and M.S.Fox, Conflict Management with the Credibility/Deniability Model, in S. Lander (ed) Proc. of AAAI Workshop on Models of Conflict Management in Cooperative Problem Solving, Seattle, WA, AAAI Technical Report, AAAI Press 1995

9. M. Barbuceanu and M. S. Fox, The Information Agent: Building Intelligent Information Infrastructures for Collaborative Enterprises, EIL working paper, oct 1994, submitted for publication. http://www.ie.utoronto.ca/EIL/public/info-infrastructure.ps

10. Peng Hu, Mark S. Fox, Managing Transportation by Constraint Heuristic Search, Technical Report, Enterprise Integration Laboratory, University of Toronto, Dec. 1994.

11. M. Barbuceanu and M. S. Fox, Temporal Recognition in Description Logics, EIL working paper, july 1994, submitted for publication. http://www.ie.utoronto.ca/EIL/public/temp-recogn.ps

12. M. Barbuceanu and M. S. Fox, The Architecture of an Agent Building Shell, EIL working paper, jan 1995, submitted for publication.

13. Fox, M.S., The TOVE project: A Common Sense Model of the Enterprise, in Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Belli, F. and Radermacher, F.J. (eds) Lecture Notes in Artificial Intelligence #604, Berlin, Springer Verlag, pp 25-34.

14. Fox, M.S., M. Barbuceanu and M. Gruninger: An Ontology for Organization Modeling: Preliminary Steps Towards Linking Structure and Behavior, To appear in Proceedings of 4-th Workshop on Enabling Technologies - Infrastructures for Collaborative Enterprises, Coolfont Resort, West Virginia, april 1995.