Agent-based Collaborative Manufacturing

Liviu Cotfas, Andreea Diosteanu, Ion Smeureanu
Bucharest Academy of Economic Studies, Bucharest, Romania
liviu.cotfas@ase.ro, andreea.diosteanu@ie.ase.ro, smeurean@ase.ro

Abstract. The current rapidly changing market requires unprecedented levels of interoperability between enterprises. The framework presented in this paper allows the implementation of highly flexible distributed agent-based supply chain systems. Starting from the customer’s initial request an entire network of suppliers is generated automatically. Agents in our framework can change roles dynamically, collaborate and compete against each other to offer the best solution for the customer’s request. In order to evaluate offers, a rule based evaluation procedure is implemented. Use of location data is also introduced, enabling agents to determine transport costs, define specific delivery areas and evaluate risks based on the customer/supplier location.

Keywords. Software Agents, Geographic Information Systems (GIS), Enterprise Collaboration, Virtual Enterprise, Supply Chain

1. Introduction

The changing economic conditions and the emerging global economy create a context in which companies that quickly and accurately evaluate new market opportunities and create innovative products and services can perform. Fully integrated traditional enterprises are being replaced by business networks in which every company is specialized in certain services or products [1], [2], [3]. Taking advantage of the developments in IT&C technologies, supply chains can be more rapidly established allowing the involved companies to benefit from reduced costs and increased flexibility. Such supply chains operate as long as the market opportunity exists and are easily adaptable to take advantage of the new technologies that are continuously emerging.

The new economic behaviors are the main consequence of the transformations that affected the traditional types of needs and opportunities and are required by the globalization trend. The global market can be characterized by increased levels of interoperability that reflect in the integration of multiple and different information systems that are able to cooperate.

Nowadays, there are true businesses networks compose of independent partners that offer to the others their specific services or products. Therefore, we can assume that companies become product and customer oriented structures. In the context of dynamic business, maximizing and optimizing business performance is a critical requirement for profitability.

Through this paper we present a software agent based framework’s architecture for boosting performance in supply chain management applications. The framework is based on agent interaction is very flexible and enables enterprise interoperability.

In the first section of the article we place our research in the current international research trends by making a short literature review. The following section presents the main elements of the multi-agent supply chain framework. The last section describes a prototype solution implemented by using the proposed framework.

Innovative features of our proposal are both the flexible agent search, the use of location and semantic annotation.

2. Multi-Agent Systems for Collaborative Manufacturing

Software agents are complex autonomous software entities that behave and interact with each other in order to collaboratively fulfill the purpose of the entire multi-agent system. [4] identified 12 desirable characteristics of an agent, among which 4 are recognized as essential [5]: autonomy, reactivity, social, pro-activity. The major advantage of multi-agent systems consists of the fact that simple individual behaviors combine into complex ones. Furthermore, another important feature of multi-agent systems refers to their ability of

Proceedings of the ITI 2010 32nd Int. Conf. on Information Technology Interfaces, June 21-24, 2010, Cavtat, Croatia
decomposing complex problems into more easily manageable sub problems. This feature is particularly useful for representing supply chains as networks composed of autonomous business that negotiate, manufacture and distribute products and services [6]. Agent technology is considered appropriate for complex, distributed software systems [7].

Using multi-agent systems in collaborative manufacturing systems implies modeling communication and cooperation between agents in order to allow demand-supply processes. As shown in [8] agents can communicate and interact with each other through ontology language.

However, if agents are used without combining with semantic web services technology, they fail to respond to the continuously changing business environment in the nowadays knowledge driven society. According to [9], the failure is associated to the fact that they function on a predefined agreement without being flexible. Also, web service description offers only a technical presentation of the features offered and not a semantic one.

The demand for flexible, adaptive multi-agent behavior has increased the challenges inherent in the design of multi-agent systems [10]. As a result, the attempt to automate the collaboration process between enterprises has been amplified. The payoff for meeting these amplified challenges is the creation of more capable, more robust multi-agent systems.

In general, adapting the organization of a multi-agent system enables agents to overcome problems or improve performance by changing the pattern of information, control, and communication relationships among agents as well as the distribution of tasks, resources, and capabilities. For example, agents may be able to overcome agent failure (by restructuring collaborative decision-making to exclude failed agents), communication failure (by allowing agents awaiting orders to eventually take initiative), and under-performance (by allowing agents to establish new collaborations that may work better) [11].

Intelligent software agents have been used in enterprise independent software systems integration process, not only to assure an approach for functional integration, but also to facilitate the use of business intelligence and collaboration among enterprises for their communication, interaction, cooperation, pro-

activeness, and autonomous intelligent decision making.

In order to achieve the objectives of the current enterprise interoperability trend, we propose a framework that combines web services and software agents so that to provide an efficient service selection, retrieval, composition and integration.

Multi-agent systems are closely connected to web service technology because they represent interoperable, portable and distributed solutions. Agents and web services may be related in different ways: agents use web services, web services are in fact agents or agents are composed of, deployed as, and dynamically extended by web services [12].

3. Multi-Agent Supply Chain Framework

The proposed framework enables the implementation of multi-agent supply chain systems that allow customers to easily identify suppliers that meet their business needs. All participants in the supply chain are modeled using software agents.

3.1. Trading Agents

According to [13] a supply chain usually consists of suppliers, factories, warehouses, distribution centers, and retailers through which raw materials are acquired, transformed, and delivered to customers.

Although creating specialized agents for these categories like shown in [14] implies several benefits we consider that this approach limits flexibility and the overall capacity of the system to model the real business entities.

Taking into account that a participant in a supply chain might belong to several of the above mentioned categories and offer multiple services and products we have chosen to implement generic trading agents.

Similar to the real world business they represent on the internet, each agent has inputs and outputs. Outputs represent the services and products the agent offers, their price levels and technical features. Inputs consist in raw material, required-services or sub-assemblies needed in order to manufacture a specific output product or in order to provide a specific service.

In order to choose the best supplier for each input, both mandatory rules (ex: maximum distance between customer and supplier) and an evaluation function are used to compute a score
for each offer (ex: based on price and technical features). Every agent will have its own evaluation function for selecting the most suitable business partners. A sample evaluation function is presented below (1). The function is parameterized, so that each agent can choose different importance weights for the constraints according to its preferences.

\[
\text{bestOffer} = \max_{i=1..m} \frac{\sum_{j=1..n} p_{ij} \cdot M_{ij}(\alpha_{jk})}{\sum_{j=1..n} p_{ij}}
\]

(1)

Where:
- \(m\) = number of suppliers
- \(n\) = number of technical features
- \(o_j\) = number of possible values for the feature \(j\)
- \(p_{ij}\) = importance of feature \(j\) for trading agent \(l\)
- \(\alpha_{jk}\) = possible values of the technical feature \(j\)
- \(M_{ij}(\alpha_{jk})\) = function evaluating the optimality of characteristic \(j\) for trading agent \(l\)

Risk could also be taken into consideration based on supplier’s location, available transport networks, etc...

If one mandatory rule is not met, the offer is rejected and will no longer be taken into consideration during the evaluation process. Based on previous experience or user input, each agent maintains a list of trusted and mistrusted agents. Importance for the above mentioned parameters can be defined using the dedicated Supplier’s Web Interface.

![Figure 1. Agent trading](image)

All agents can and usually perform several roles in the supply chain. In Figure 1 the agent called Trader 1 is a supplier for the agent Customer from whom it receives a service or a product request. If Trader 1 cannot complete the request by itself he can request additional services or subassemblies to other agents becoming thus a customer. Trader 2, 3 and 4 are possible suppliers for Trader 1, but they also can become customers for other agents. If a trader agent cannot supply the requested service or product in the requested conditions, it returns a void offer.

Ontology Web Language (OWL) [15] was used to semantically describe the products and services and also their technical characteristics. All configurations are done using the trader agents’ web interface.

Although location is considered an important aspect [16], many existing frameworks fail to use it at the full potential or omit it completely. We consider location import as it might have a big influence on transport costs and risks involved. It also helps us narrow the search when the requested service or product can be delivered only in certain areas. GPS coordinates are stored for all agents involved. Besides, for agents offering distribution services, GPS coordinates for the delivery areas are also stored as geographic shapes (polygon) using Geographic Markup Language (GML) standard format as defined by Open Geospatial Consortium (OGC) [17].

### 3.2. Identifying the best offer

In order to find the best offer a series of steps are required. The communication between agents is based on OWL ACL messages compliant with FIPA [18] standards. The proposed system functions in a distributed manner as every agent follows his own goal in the negation process considered as a whole.

**Step 1:** All Domain Directory Facilitator (DDF) agents register themselves with the main Directory Facilitator (DF) agent. Domain Directory Facilitator agents will keep information about service from a certain domain. A trader agent first queries the DF agent in order to obtain the available DDFs and then, it registers its services and products with the corresponding ones.

![Figure 2. Distributed agent-based architecture](image)
As shown in Fig. 2, the framework is implemented in a distributed manner that allows having multiple DF agents running on different machines. Communication between the distributed components is implemented using web services for better interoperability across heterogeneous computing environments. The Web Service Agent (WSA) agent manages the translation between ACL messages and web service calls and constantly maintains a list of DF agents on other machines.

Step 2: As the supply chain is a customer demand-driven system, the flow starts from the customer’s incoming orders. The customer agent first queries the DF and DDF agents on all machines for the list of supplier agents that provide a certain service or product.

Step 3: The customer agent sends its demand to the seller agents discovered in the previous step. The demand specifies the requested delivery addresses using GPS coordinates, the requested products, their technical features, quantity and maximum delivery date.

Step 4: Each contacted supplier agent analyzes the customer’s demand and respond in case of match. First it compares the geographic position of the customer with its points of sales covered surface. Next, the seller agent compares the request products or services with his own data described using OWL. Using OWL proved to be a good solution to overcome data heterogeneity. Inference was implemented using Jena [19] to better understand the customer’s request. If the seller agent can’t provide the requested service or product by itself (ex: it needs to buy subassemblies or raw materials), it will need to find suppliers. Thus, the seller agent becomes a customer for other seller agents and the process repeats starting with Step 2 (Fig. 1).

Step 5: The customer agent continues to communicate and negotiate with the agents that positively respond. The offer evaluation takes into consideration both mandatory rules and the evaluation function (bestOffer). More than one seller agent might be selected if necessary in order to meet the required quantity. If the customer agent is not the initial customer, but a seller agent searching for necessary sub-assemblies or services, the algorithm continues with the selection of the best seller offer in the previous recursion step. The algorithm ends when the best seller offer evaluation is done by the initial customer agent.

As shown above, the algorithm is implemented in a recursive manner that allows each seller agent to become a customer agent if needed. This approach guarantees a high degree of flexibility in generating the supply chain.

The customer agents can take into consideration a set of predefined rules when selecting suppliers like shown in Fig. 3.

Figure 3. Agent search in supply chains

Additional reasoning capabilities can be added to seller agent as shown in [20] [21] in order to enable the seller agents to adapt prices according to market situation so it can maximize its revenues by selling at appropriate prices (Fig. 3).

4. Prototype Implementation

For our prototype implementation we used both Java Agent Development Framework (JADE) [22] and Jena Framework [19].

In order to make the prototype implementation accessible from a wide range of devices and platforms, the interface is web based. The application allows companies interested in network economy to configure their products and services and to define delivery areas. The delivery areas can easily be defined by drawing on the map (Fig. 3).

Figure 4. Adding a new delivery area
Great care has been taken when designing the user interface for mobile devices in order to make it accessible and easy to understand even on small screens. The proposed application is implemented in a distributed manner that enables agents running on different platforms to communicate over the internet. Agents contribute and communicate to each other in order to fulfill the main target of the system: mediating supply chain transactions.

Moreover, the implementation of the Web Service Agent (WS Agent) makes the application accessible using web service calls and therefore easier to incorporate in existing or new systems (Fig 5).

5. Conclusions

Our work presents a flexible and adaptable multi-agent supply chain framework. Innovative aspects include the bid evaluation method, the use of location data and OWL semantics. The proposed framework relays on standards recognized by W3C, FIPA and OGC in order to assure compatibility with other systems. Further development includes implementing a bidding model and reasoning capabilities for seller agents to adapt their prices based on supply and market demand.

6. Acknowledgements

This article is a result of the project „Doctoral Program and PhD Students in the education research and innovation triangle, DOCECI”).

7. References


