REQUIREMENTS FOR AN AGENT BASED INFORMATION SYSTEM SUPPORTING VARIABLY COUPLED NETWORKED ENTERPRISES

Américo L. Azevedo, Jorge P. Sousa, António L. Soares
*Faculdade de Engenharia da Universidade do Porto and
INESC Porto / Manufacturing Systems Engineering Unit,
Rua José Falcão 110, 4050-315 Porto, PORTUGAL
Email: {azevedo, jsousa, asoares}@inescporto.pt*

This paper presents the requirements for an information system supporting the co-operation in networks of enterprises. First, these networks are characterised, focusing on the variable coupling of the participating enterprises and on the consequent needs of co-operation. Departing from those needs, the concept of an information system based on an architecture of agents is presented. A generic infrastructure to implement the system is outlined, including the main interaction scenarios. The next steps of this research work end the paper.

1. INTRODUCTION

Co-OPERATE is a project supported by the European Commission, that started in January 2000, and involves several industrial companies, a consultancy firm and two academic groups. The pilot companies are part of rather complex and dynamic networks or supply-chains, and will be used in the project to define and develop distributed, but co-operatively inter-operating business processes and information infrastructures.

The project aims at providing an advanced information and communication infrastructure to support co-operation in distributed and virtual enterprises. A distributed and decentralised information system, based on an architecture of agents, and extensively using the internet is being designed and implemented, as a means to provide new and more powerful decision support tools to be used in this type of environments. Solutions will be developed to enhance the entire supply network, from the final product manufacturer to the original material suppliers. An advanced information and communication infrastructure will be provided, to support co-operation in dynamic enterprise networks, as well as a methodology for co-operative planning, and a set of mechanisms to enable real-time monitoring of the supply-chain. Moreover, tools to support a fast and effective reconfiguration of manufacturing networks will be made available.

The project has the following strategic objectives aimed at improving the performance of networked enterprises: higher responsiveness; lead time reduction

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35399-9_52](https://doi.org/10.1007/978-0-387-35399-9_52)

L. M. Camarinha-Matos et al. (eds.), *E-Business and Virtual Enterprises*

© IFIP International Federation for Information Processing 2001

throughout the supply chain; improved flexibility in the supply chain; reduced inventory levels; and an increased capability to solve unexpected problems in the supply chain.

In order to achieve these goals a number of “business solutions” have been defined, that will focus on the network aspects of the business processes and will try to accommodate and support the current internal tools and processes of companies. These business solutions cover the following issues [Collings and Loeh (2000)]: long term business planning for the network; standard operational order and planning processes; feasibility studies for new order or change requests across the network; exception handling; multi sourcing co-ordination; process visibility; performance management information.

2. GENERAL FEATURES OF NETWORKED ENTERPRISES

New and more demanding needs in terms of product innovation and market responsiveness raised the concept of agility and led many enterprises to improve their organisation and structure. This is often achieved through the establishment of temporary production networks, as an attempt to satisfy both customer expectations and long term enterprise goals and viability. Agile, market responsive supply-chains must not only deal with a rather volatile demand, but also cope with this variability through frequent chain reconfigurations and the implementation of new methodologies or business processes.

Within a given network, the needs of each entity will vary depending on the primary focus of the collaboration and on the nature of the product or service provided. Three main types of network arrangements may be identified [Miers and Hutton (1997)]:

Dominant organisation with a network of suppliers - This model is often used within large organisations, where one dominant player sets the main rules and uses a network of small firms to meet its needs and those of its customers. In this type of relationship, processes are usually focused on the repetitive supply of a service or product. The terms of the relationship are in general well defined in advance.

Co-operative network - This model involves networks of small businesses working together to support larger projects or services (which in turn may be supplied to larger or more dominant players). Processes are project oriented and require a trust relationship between all co-operating entities. This type of networks has an enormous potential for growth. Currently there are no suitable technology solutions to support this kind of model.

Virtual teams - The trend in most large businesses is towards having teams that are managed on a project by project basis (seeking for a more agile structure).

3. REQUIREMENTS FOR CO-OPERATION IN VARIABLY COUPLED NETWORKED ENTERPRISES

In this context it is clear that we are talking of physically distributed networks of

units (several sites) that are possibly owned by different companies. These units are separately managed entities (plants, subcontractors, transport companies, warehouses, “customers”, suppliers, ...) that have some kind of relationship with the business around which the Virtual Enterprise has been set up. Within the same network, the degree or strength of these node-to-node relationships may be quite different, and it has not only to do with ownership or contractual links, but also with informal agreements or even with some subjective commitments. Hopefully we should be able to explicitly take into account these differences and adequately manage them.

It should also be noted that networked enterprises are dynamic by nature and that the degree of “coupling” between units (nodes of the network) may change very rapidly. To have the capability to handle this issue is obviously a key requirement of information systems that is not trivial to design or implement. In such systems we might say that the main issues are the logistic processes and co-operative planning. However, in a slightly different but complementary perspective, special attention has to be given to the life cycle of the whole co-operation processes i.e., requirements are related not only to the operation of the network but also to its creation and reconfiguration.

3.1 Some basic assumptions

Along this line of concerns, Co-OPERATE aims at addressing a comprehensive set of supply chain management activities (in an environment of Extended or Virtual Enterprises), by focusing on the co-ordination of tasks involved in manufacturing planning and control.

In this context, some important issues are: logistic aspects; global visibility (real-time monitoring of all links); order promising; co-operative planning; plan repairing. The network (or supply chain) is managed by a set of units, each responsible for one or more activities and for interacting with other units. Co-OPERATE will provide an infrastructure able to support a community of co-operating logical entities belonging to the units and providing: a) at the planning level, support to co-operative planning based on negotiation; b) at the execution level, real-time monitoring of the supply chain status and early detection of problems and repair services.

4. A CONCEPT FOR THE CO-OPERATE INFORMATION SYSTEM

Given these general requirements, we have been investigating and designing general methodologies and tools for improving supply chain management through some kind of overall distributed production planning. In contrast to centralised planning systems, a decentralised approach naturally partitions global complex planning problems into smaller specific problems, which are tackled by (local) independent organisational units. Along this line, one key objective of our research is to perform a comprehensive evaluation of the benefits of a distributed approach, based on a framework of multiple agents that communicate through ‘speech acts’ of advanced languages. In fact, *multi-agent systems* seem to be well-suited for modelling

networks, as they naturally provide a set of features that make the implementations and the operation of complex distributed systems easier (Azevedo and Oliveira, 1999).

We therefore propose a global approach to these problems that includes a collaborative planning methodology, along with an information system architecture based on several autonomous units (one for each site or network node) that are able to perform a set of activities, and to anticipate the consequences of their actions, within the scope of their commitments. Agents own a set of key features (as usually identified in the literature) that are of major importance for the effective management of a co-operative and distributed manufacturing environment: autonomy, proactivity, sociability and responsiveness

In fact, by having a set of agents communicating and co-operating with each other, we can address richer and more complex problems, and can increase both the functionality and the efficiency of planning processes. Moreover, a natural support is provided for the whole life-cycle of the enterprise network, especially in what concerns: its set-up, in a static or in a dynamic way; propagation of information through the chain (e.g. the announcement of customer orders, etc.); negotiation between units, leading to a global production plan; and distributed control and "traceability" of manufacturing orders throughout the chain

Agents are software entities with an high degree of autonomy and loosely coupled to other entities (other agents). This means that when the roles of an agent in a community of agents are well known, the fact that an agent enters or leaves the network results in an increase or decrease of the overall capabilities of the community without major disturbances.

In the Co-OPERATE system each agent in a node plays one or more roles. Therefore adding or removing an agent results in the reconfiguration of the roles played by a node. This leads to an easiness of addition or deletion of roles in each node without any disturbances in the network (no changes are needed for the other nodes). The basic functionality for effective agent co-operation will be directly implemented in an infrastructure that provides the basic services for that co-operation. The so-called agent frameworks are in an advanced stage of development and can be implemented on top of state-of-the-art application servers.

In summary we might say that, concerning order negotiation and planning in a highly dynamic supply-chain or enterprise network, the agent-based approach should allow us to set the right ground for: higher levels of autonomy and proactiveness; new communication and expression paradigms, to promote delegation and co-ordination; new planning methodologies and algorithms, based on negotiation and in co-operation; and some level of "intelligence" integrated into management applications.

The adoption of agent technology in the Co-OPERATE project involves the type of risks associated with the adoption of a not fully matured technology. Although there is almost a decade of R&D in this field, the IT evolution only now enables cost-effectiveness in this kind of tools.

In despite of this possible drawback, the adoption of agent technology by major IT vendors is a good indicator of its potential and that it should soon become a sound approach. Agent technology already proved to be effective in conceptual and technical terms. We can say that the major risk involved with the adoption of agent

technology lies mainly on its rate of evolution. Nevertheless, being a natural choice for a major part of e-business/e-commerce applications, agent technology will be rapidly adopted by major IT vendors and web application developers.

4.1 Basic architectural principles

Co-OPERATE consists of a number of business solutions, each one tackling a specific area of network co-ordination. To generically support this, we consider a software agent based system, structured around the following basic architectural principles:

- each business unit in the network of companies will be served by a set of agents;
- each of these sets will form a "node" in a community of agents distributed by the several "units" in the network;
- in each node, the agents co-operate to achieve local goals;
- in each node, each agent performs one or more functions, and co-ordinates its decisions with the other agents in the node;
- different nodes in the network co-operate to achieve global or local network goals; the node co-operation is carried out by the co-operation of individual agents in the different nodes;
- the functionality across the network or supply-chain is achieved through the interaction of the different nodes;
- the number and types of agents present in each node is variable, i.e. we can have all kind of agents or only some of them.

These principles define a direction for the design of the agent based system, and they adopt a "corporate" social interaction between agents. This means that local goals have priority over the network goals, i.e., when co-operating with other nodes, the agents in a node will act in a way that do not conflict with local goals. The alternative would be a "global" social interaction where each agent acts according to the network goals that are decomposed into individual goals. This means that local goals (roughly, the business unit goals) are secondary when compared with the network goals.

4.2 Types of agents

The software agents supporting the Co-OPERATE business solutions will perform different functions to achieve given goals. Following a basic principle of division of work, each agent will perform a related set of functions, i.e. each agent will play a *role* in the node where it belongs.

We identify here the basic *roles* that different types of agents in Co-OPERATE should play: information provider, facilitator, co-operative planner, capacity manager and monitor (see Table 1). This role descriptions result from a preliminary analysis and clearly further work has to be done in this area. The descriptions are not exhaustive, and in some cases there are even not fully consistent.

Table 1 – Types of agents in Co-OPERATE

<i>Information provider</i>	provides information services; manages information within the node; collects and aggregates information supplied from different sources; answers queries from other roles and nodes; builds up data views; propagates information required by other roles.
<i>Facilitator</i>	acts as an organisational agent and is responsible for providing support to other agents; it provides, for example, services to find the location of external agents of a given type or having certain specific skills or properties.
<i>Co-operative planner</i>	co-operates with other planners in other nodes, in order to develop plans that achieve the network goals (basically each planner has a planning algorithm and a reasoning mechanism to negotiate and collaborate); interacts with the local capacity manager; co-ordinates activities at a local (node) level.
<i>Capacity manager</i>	dynamically manages the availability of the business unit resources; interacts with the applications for Materials Management and with the Manufacturing Execution Systems.
<i>Monitor</i>	acquires and aggregates the network or supply-chain state variables; compares current state variables with target values; identifies deviations or disruptions, e.g. resource failure (loss of production capacity), logistics problems (transport delays), quality issues; propagates warnings to the other network nodes.

5. GENERIC INFRASTRUCTURE FOR THE CO-OPERATE IMPLEMENTATION

5.1 Agents internal architecture

Each agent is composed by three fundamental subsystems: communication and interaction; business logic; and sensor/actuator with the external context.

Communication and interaction subsystem - This subsystem deals with the interaction between agents of the same or different types. It allows the agent to receive messages from other agents, expressed in the KQML language. It is also used to prepare and send messages to one or more agents. The communication between agents can be performed through different transport mechanisms, namely protocols (http, https, smtp), the CORBA infrastructure, etc.

Business logic subsystem - This subsystem implements the agent operation and behaviour. It is formed by a knowledge base and by a reasoning mechanism based on simple rules and heuristics, to support decision making during the negotiation process. The functionality of this subsystem strongly depends on the agent under consideration. It may for example implement a mechanism for negotiating and

propagating constraints; for querying other agents; for building data views; for acquiring and aggregating network state variables; for comparing results against target values; for propagating warnings to the network, etc.

Interface with the world - This subsystem deals with the bi-directional interaction of the agent with its environment. For example, in this case, this means interactions with ERP and legacy systems, as well as with decision makers.

5.2 Interaction scenarios

In a supply-chain formed by business units that may have quite different types of relationships (according for example to their relative position within the supply chain), several interactions between the nodes may take place at the same time. In this context, nodes may be viewed as passive or active, and they may require the intervention of the "human planner" or just respond automatically to external actions or requests.

To interact and collaborate, some nodes use an infrastructure based on an *extranet* (external intranet), allowing a high active level of manual and automatic communication. At another level, there may be some passive interactions through the *Internet* (access to WEB servers) or active interactions through *push* technology or other subscription or notification mechanisms (for example, *e-mail*). Moreover, some other traditional communication mechanisms may be considered, such as EDI, phone, fax, etc.

5.3 Software infrastructures for agent-based systems implementation

The more important features that a software infrastructure for agents can provide are the implementation of an high-level language and communication protocol, and message routing services, which include the support to naming and registration services, and asynchronous communication between agents. Agent functionalities (as defined above) can be implemented by using state-of-the-art technology such as application servers based on EJB (Enterprise Java Beans) (e.g., variants of the J2EE architecture model), and making use of concepts such as components.

These features are made available by some frameworks aimed at building and supporting software agent based systems. Regarding the first feature, the language and communication protocol, there seems to be a consensus in the use of KQML. For the second feature, message routing services, there are several possible approaches. In any case, the available infrastructures, while fully functional, are still under development. However a particular reference should be made to the two more promising approaches: JKQMLⁱ and JATLiteⁱⁱ.

The former is a framework and API for constructing Java-based, KQML-speaking software agents that communicate over the Internet, supporting KTP (KQML transfer protocol), ATP (agent transfer protocol) and OTP (object transfer protocol). The later (Java Agent Template, Lite) is a package of programs written in the Java language that allow users to quickly create new software "agents" that communicate robustly over the Internet, providing a basic infrastructure, in which agents register, connect/disconnect from the Internet, send and receive messages, transfer files with FTP, and generally exchange information with other agents on the

various computers where they are running.

6. CONCLUSIONS AND FURTHER WORK

In this paper, some general requirements of distributed and decentralised information systems for companies operating in networked environments have been identified. We have also presented a global approach to these problems, based on an architecture of agents, and extensively using the internet, that is being designed and implemented in the currently running European project Co-OPERATE. The types of agents used, their basic functionality, and the main processes of co-operation they use were highlighted.

Preliminary results obtained with a system prototype prove the potential of the approach, in particular by showing it is rather flexible and robust. But obviously further research is needed in understanding the mechanisms of decision-making for networked enterprises. On another direction, we are currently working on collaborative and negotiation planning procedures for improving network co-ordination, as well as on the complex multi-criteria decision-making processes involved in this context.

Finally it should be noted that the capability of networked enterprises to dynamically change the type and strength of relationships between companies is a key requirement of information systems. But this is not a trivial issue in terms of design or implementation, and deserves therefore further research. A theoretical framework based on the "social actors networks" approach is being worked out in order to be used as a tool to describe and analyse the interplay between the different actors in networked companies. The approach is potentially well integrated with the description and analysis of requirements of networks of software agents, constituting a framework for the requirements analysis and specification of techno-organisational structures of networked companies.

7. REFERENCES

1. Bradshaw, J. M. 1997. An Introduction to Software Agents. In *Software Agents*, ed J. M. Bradshaw. Menlo Park, Calif.: AAAI Press.
2. Petrie, C, 1996, Agent-Based Engineering, the Web, and Intelligence. *IEEE Expert*, December.
3. Stan Franklin and Art Graesser, 1996, Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*, Springer-Verlag, 1996.
4. IBM Aglets workbench. <http://www.trl.ibm.co.jp/aglets>.
5. Finin, T. et al. 1993. DRAFT Specification of the KQML Agent-Communication Language plus example agent policies and architectures. The DARPA Knowledge Sharing Initiative, External Interface Working Group.
6. Co-OPERATE Project, Solution Scope. Project IST-1999-12259.
7. Miers, D. and G. Hutton (1997). Technology Futures For The World Wide Web - Business Needs Something Better. <http://www.cutter.com/consortium>. Cutter Consortium.

¹ See <http://www.alphaworks.ibm.com/tech/jkqml>

² See <http://java.stanford.edu/ProcessLink/papers/JATL.html>