Taylor & Francis
Taylor & Francis Group

Cooperative planning in dynamic supply chains

A. L. AZEVEDO*†‡, C. TOSCANO‡ and J. P. SOUSA†‡

†Faculdade de Engenharia da Universidade do Porto
‡INESC Porto, Rua Roberto Frias, 378, 4200-465 Porto, Portugal

This paper describes an order planning system for dynamic supply chains, addressing the requirements of a make-to-order business environment. A distributed and decentralized information system based on an architecture of agents and extensively using the internet was designed and implemented, so as to enable new and more powerful decision support. The system aims at responding to the basic requirements of cooperativeness, integration and configurability. It was developed under the scope of the IST European Project Co-OPERATE and implements the functionality defined in the context of the 'request feasibility studies for the network' (ReFS) business solution.

1. Introduction

The coordination and optimization of complex and dynamic supply chains (or networks) consisting of independent companies require management systems able to link the most important business processes and make them inter-operable. By adopting new approaches for supply chain integration and collaboration, companies can significantly improve their returns through improvements in efficiency, higher delivery reliability, better asset and capacity utilization, faster time to market, and responsiveness. This is particularly important in the complex and highly dynamic operating environment of the automotive and electronics industries (Azevedo 1999).

This paper describes part of a Decision Support System (DSS) for cooperative planning in dynamic supply chains addressing the requirements of a *make-to-order* business environment. The system resulted from research and development activities pursued under the framework of the Co-OPERATE European Project (IST-1999-12259). The project aimed at developing solutions to enhance the entire supply network, from the original material supplier (semiconductors) to the OEM manufacturer (automotive) by providing an advanced information and communication infrastructure to support cooperation in networked and distributed organizations (Collings and Loeh 2000).

The DSS sits on top of the Enterprise Resource Planning (ERP) systems of each company of the production network and periodically pulls out information regarding forecasted demand, backlog, and the production schedule for each finished product. In each company of the network, a capacity model is used to assess the company's capabilities and to compute its production capacity, as well as to manage the availability of the suppliers for the required materials. Unplanned demand coming from any of the company's customers is immediately analysed, by triggering a feasibility check all over the network of suppliers. Information supporting decision making is then passed to the person who is negotiating the request with the customer. A plan of partial deliveries, based on Available-to-Promise (ATP) and Capable-to-Promise (CTP) information, is finally produced and presented to the customer for approval. The ATP quantity is the uncommitted production—inventory and planned—maintained in the master production schedule (MPS) to support customer order promising. The ATP is normally calculated for each period in which an MPS receipt is scheduled. The CTP denotes the capability of a production system or resource to produce a quantity of output in a particular time period (APICS 1995).

The paper is organized as follows. Sections 2 and 3 give a brief description of the business solutions (processes) characterized by the Co-OPERATE project. Particular

*Corresponding author. Email: ala@fe.up.pt

International Journal of Computer Integrated Manufacturing
ISSN 0951-192X print/ISSN 1362-3052 online © 2005 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/09511920500081304

emphasis is given to the 'ReFS' business solution for which a DSS has been designed and implemented. Section 4 presents the architecture of this DSS and its functional model is described in section 5. Finally, some conclusions and current developments are briefly presented in section 6.

2. The Co-OPERATE project

The Co-OPERATE project has developed solutions to enhance the entire supply network, by providing an advanced information and communication infrastructure to support general cooperation, as well as particular methodologies for cooperative planning and for network setup and support. The project started in January 2000 and finished in April 2002. Two academic groups, a consultancy firm and several industrial companies were involved in the project (Collings and Loeh 2000).

The project is based on the vision of delivering improved collaboration across the supply chain, leading to improved customer service and significant cost reductions. Enhanced visibility of information, early warning of disturbances, synchronized production, and collaborative planning in the supply chain are the goals to attain. Seven business solutions have been identified and analysed, forming a conceptual framework and a set of methodologies to improve aggregate planning. A brief description of each business solution is presented here.

(1) *Long-term business planning for the network*
This business solution improves the long-term planning process along the whole supply network. The goal is to deliver guidelines for the network operation, by synchronizing forecasts and strategic plans, and by promoting early communication of changes and feedback about feasibility.

(2) *Standard operational order and planning processes*
This business solution supports the short-and medium-term operational order and planning processes (from weeks to months). This includes order generation and transmission, synchronized planning, monitoring, and status information. This business solution provides a basis for real-time order promising and exceptions handling.

(3) *Feasibility studies for new order or change requests across the network*
This business solution supports the request for new orders or large order changes across the network and enables the coordination of the feasibility checks at the individual companies within the shortest possible time. This includes checking of capacity and availability of materials from suppliers.

(4) *Exception handling process*
This business solution includes methodologies to detect, as early as possible, problems in the network such as peak orders, capacity shortages, or problems in the availability of parts.

(5) *Multi sourcing coordination*
This business solution optimizes the allocation of work through the different suppliers. This is based on a defined set of rules and a fast negotiation process on capacities, involving a strong interaction with the suppliers.

(6) *Process visibility*
This business solution provides mechanisms to assess the progress of orders in the supply network.

(7) *Performance management information*
This business solution provides management information about the performance of the network processes. It includes the identification of key performance indicators and provides the methodologies to collect and present those indicators.

Business processes have been modelled and analysed and a conceptual framework and methodologies for aggregate planning have been tested and tailored in the pilot companies. The prototype developed so far, based in an advanced multi-agent architecture (Sycara 1998), is able to plan orders across the entire production network using capacity feasibility checking through local capacity models (Leach and Ristic 1998). These models use algorithms that are customized, configured, and optimized for each production unit implementation. Hopefully they will be easy to adapt to other industrial sectors.

3. Problem scope

The Co-OPERATE Project has defined a number of 'business solutions' or processes to fulfil the 'global' needs of companies, focusing on the network aspects of their business processes.

It is one of these business solutions, called 'Request feasibility studies for the network' (ReFS), that the research described in this paper deals with. Its main objective is to provide a fast response to incoming new orders ('order promising') or to handle requests for large order changes that exceed the current availability of some of the suppliers. When a request comes to the enterprise network at a given node, its feasibility is checked internally taking into account the capacities and planned production of the node, and externally by forwarding the request to the next relevant nodes (suppliers in the supply network). The final answer to the customer is then fed back to the node in which the initial request occurred.

Several types of 'what if' analysis related to the fulfilment of a customer request are provided by the system. The following are examples of questions supported by the system:

- What quantity of a given product can be delivered to the customer by the requested due date?
- On what date could the entire request be satisfied?
- What additional resources would be needed fully to satisfy the customer's request?

One major requirement for this business solution is that the response to a customer inquiry should be given in useful time, i.e. much faster than traditionally. Moreover, the scheduling of large order changes should be reliable and be consistent with the other supply network processes. The focus of this particular process is on situations in which the reaction time frame allows regular operational changes and adaptations, as opposed to emergency handling (dealt with by another business solution of the Co-OPERATE project).

4. System architecture

4.1. Basic architectural model

The basic architectural principles behind the ReFS DSS (intended to implement the ReFS business solution) have been presented by Azevedo *et al*. (2001). Basically, the supply chain is modelled as a distributed and decentralized system, with no central activity coordination. A *software agent* (Nwana 1996, Franklin and Graesser 1996) representing each node in the network automates the execution of the tasks related to the *order promising* functionality. That agent implements an aggregate planning methodology. Interactions among companies in the network are modelled as 'speech acts' (Labrou *et al.* 1999) carried by software agents (RefsAgent) that have the knowledge and expertise of their human counterparts in the company.

Figure 1 presents the main elements of the ReFS system using UML (Unified Modeling Language) (Booch *et al.* 1999). The *Planner* symbol represents a human in the company responsible for the planning functions. The *RefsAgent-UserInterface* component provides the user interface of the RefsAgent, made available in the Intranet of the company through any Internet browser. The connection mechanism was built on the Remote Method Invocation (RMI) protocol over the Internet Inter-ORB Protocol (IIOP). RMI is a distributed object model for the Java programming language that retains the semantics of the Java platform's object model, making distributed objects easy to implement and use (Sun Microsystems 2002).

A strategic choice in Co-OPERATE was that there can only be direct exchange of information between companies with direct customer–supplier business relationships. Therefore this principle is followed in the interaction between different instantiations of the RefsAgent.

The *LegacySystem* components depicted in the right part of figure 1 represent any type of data source in the company from which the data needed by the RefsAgent can be obtained. Interactions between instances of a RefsAgent and a LegacySystem are accomplished through the periodic exchange of XML-based data. In this context, XML provided the tools to describe the structure of those data elements and to implement a basic, but very flexible mechanism, that allows the exchange of XML data elements between a RefsAgent and a Legacy System, in a standard way (W3C 2004).

4.2. Protocols and conversations

The interaction between the several software agents forming the multi-agent system of a given network of companies was implemented following the rules defined by the standardization work of FIPA—the Foundation for Intelligent Physical Agents (Aparicio 1999, O'Brien and Nicol 1998). Accordingly, a conversation is an ongoing sequence of communicative acts exchanged between two or more agents relating to some ongoing topic of discourse. As such, a protocol is defined as a set of conversations that exhibit typical patterns of message exchange.

The RefsAgent implements the FIPA-request protocol (see figure 2). The 'Initiator' is the RefsAgent that initiates the conversation and the 'Recipient' is the destination RefsAgent (in a one-to-one connection). The protocol is instantiated twice, in order to implement the following two types of conversation between any two RefsAgents:

- query the ATP and the CTP information (see section 5.2 for an explanation of these concepts);
- check the feasibility of a new order request.

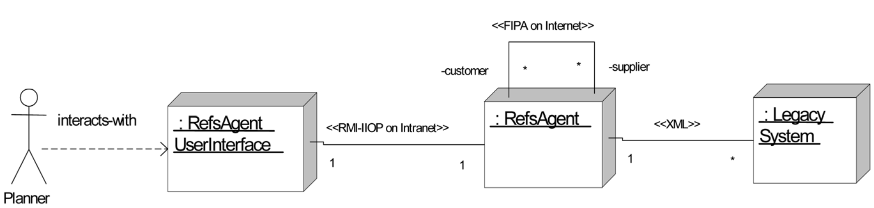

Figure 1. Architecture of the ReFS system (UML deployment diagram).

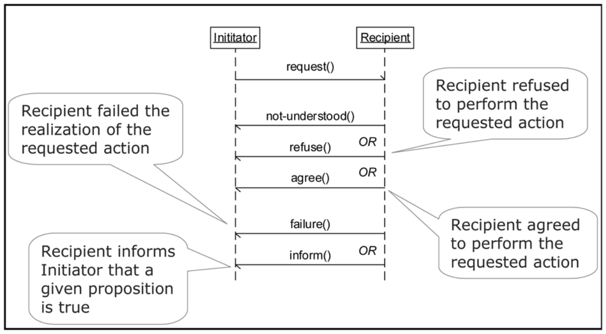

Figure 2. FIPA—request protocol (UML sequence diagram).

4.2.1. ATP and CTP information query. The protocol is instantiated by any RefsAgent in situations where the ATP and CTP information is required from the direct suppliers of the company represented by that RefsAgent. Three items form the input to the protocol:

- *SupplierID:* identification of the RefsAgent that is the target of the inquiry process;
- *ProductID:* reference code of the product supplied by the target company and for which ATP and CTP information is required;
- *DeliveryWeek:* delivery week for which ATP and CTP information is required.

In normal situations (see figure 2) the reception of a 'request' message triggers the transmission of an 'agree' message. The ATP and CTP values are then computed by the recipient ReFS agent, for the product referred by 'productID' and for the week referred by 'DeliveryWeek', and provided to the initiator RefAgent through the 'inform' message.

4.2.2. Feasibility checking of new order requests. This protocol is instantiated by any RefsAgent whenever it is required to analyse the feasibility of a new order request. The recipients of this type of conversation can only be the companies that act as direct suppliers of the company represented by the RefsAgent initiating the conversation.

The following items form the input to the protocol:

- *SupplierID:* identification of the RefsAgent that is the target of the inquiry process;
- *ProductID:* reference code of the product ('component') to be supplied by the target company, as part of the order request under analysis;
- *Quantity:* number of units of the product referred by productID;
- *DeliveryWeek:* due date of the order request.

The reception of a 'request' message by a RefsAgent triggers its internal analysis process, eventually resulting in the transmission of an 'agree' message (if the agent agrees to analyse the request), followed by an 'inform' message, describing the result of the inquiry process.

4.3. Internal organization of the RefsAgent

The internal organization of the RefsAgent is represented in figure 3.

The FIPA–OS services component represents the whole set of services provided by the FIPA–OpenSource (FIPA–OS) framework (Poslad *et al*. 2000) which is a reference implementation of the FIPA open standards for agent interoperability. This framework, available in the Java computational platform, provides support for:

- different types of agent shells for producing agents, that can then communicate with each other using the FIPA-OS services;
- multi-layered support for agent communication;
- message and conversation management;
- dynamic platform configuration, multiple types of persistence, and multiple encodings;
- abstract interfaces and software design patterns;
- diagnostic and visualization tools.

The RuleEngine component gives the RefsAgent the ability to 'reason' using knowledge supplied in the form of declarative rules and heuristics. Decision making intended to support internal processes is accomplished via this inference mechanism. A third-party tool, the Jess rules engine (JESS 2004), implements this component. Basically, Jess supports the development of rule-based expert systems that can be tightly coupled to code written in the portable Java language. The different rules and heuristics defined by the human user are kept in an XML file inside the DataRepository component.

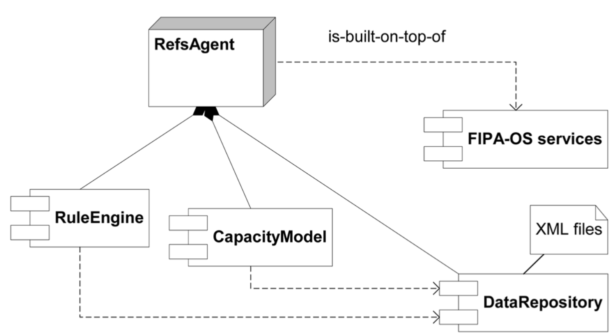

Figure 3. RefsAgent internal organization (UML component diagram).

The CapacityModel component is a full Java application module. Its purpose and functionally are detailed in the next section.

Data persistency of the whole RefsAgent is assured by the DataRepository component through a set of XML files. Moreover, these XML files act as the interface between the RefsAgent and the available legacy systems within the company.

5. Functional model

5.1. Capacity model

The ReFS DSS deals with rather aggregate information adequate for the global planning of the whole supply network. In this context each capacity model provides ReFS with a way to evaluate and measure the capacity of the associated unit, supports the creation of medium–long-term plans, performs material management, and evaluates the local implications of a given customer order. A capacity model is built-up on the following key concepts:

- *Meta-product*: group of products (finished products or product families) that have similar characteristics concerning their production processes (similarity considered here at the level of aggregation dealt by the model).
- *Macro-operation*: a concept aggregating a close sequence of real operations or a set of functionally similar operations; these operations are viewed as one basic macro-operation with a single capacity (i.e. a macro-operation is a high-level conceptual operation).
- *Resource-centre*: aggregation of a set of shop floor work-centres; it imposes the limiting capacity resource for a specific macro-operation.
- *Production-routing*: the sequential set of macro-operations that have to be carried out in order to manufacture a single meta-product.

An effective (probably not constant) capacity along the planning horizon is associated with each resource-centre.

The construction of an aggregate data model for the production process of a set of finished goods involves two main actors inside the company: a human planner and the company's own ERP system. The human planner is responsible for creating and managing the capacity model. This process takes place through the user interface of the RefsAgent, available through an Internet browser of the company's Intranet.

The following items are the basic data required by the model from the company's ERP system:

- *Product identification:* unique reference of each product (finished product or product family) supplied by the company;
- *Product bill-of-material:* required component parts for each product plus the identification of the companies that supply those parts;
- *Forecast demand:* estimated quantity demanded by the customers of the company in each time period of the planning horizon;
- *Backlog:* orders already committed by the customers of the company along the planning horizon; backlog orders partially consume forecast orders;
- *Master production schedule:* quantities to be delivered by the production unit in each time period of the planning horizon.

5.2. Available-to-Promise and Capable-to-Promise

Available-to-Promise (ATP) information (APICS 1995, Azevedo 1999) is used for feasibility assessing and responding to new order requests. The ATP value is available for each product produced by the company in each time period of the planning horizon. It is the uncommitted portion of the company's inventory and planned production of a specific product in the considered time period, being therefore the difference between the total amount of production for each product (according to its master production schedule) and the already committed customer orders (the company's backlog). Two types of ATP information are considered here:

- *Cumulative ATP:* total number of available units of a certain product taking into account the previous time periods: the quantity available for promise in period i is used in the calculation of the quantity available for promise in period $i + 1$.
- *Discrete or non-cumulative ATP:* total number of available units of a certain product without taking into consideration past periods.

For a given product, and letting i denote a time period, ATP values can be easily computed as follows (with 0 denoting the first period of the horizon):

$$Discrete\ \ ATP_0 = InitialStock + MasterProductionSchedule_0\text{-}Backlog_0$$

$$Discrete\ ATP_i = MasterProductionSchedule_i\text{-}Backlog_i$$

The $MasterProductionSchedule_i\ (i > 0)$ is the total amount of the product under consideration to be delivered by the production unit of the company in the beginning of the period. The $Backlog_i$ is the quantity already committed to customer orders for time period i.

Cumulative ATP results from the accumulation of the above values period by period, along the planning horizon as follows:

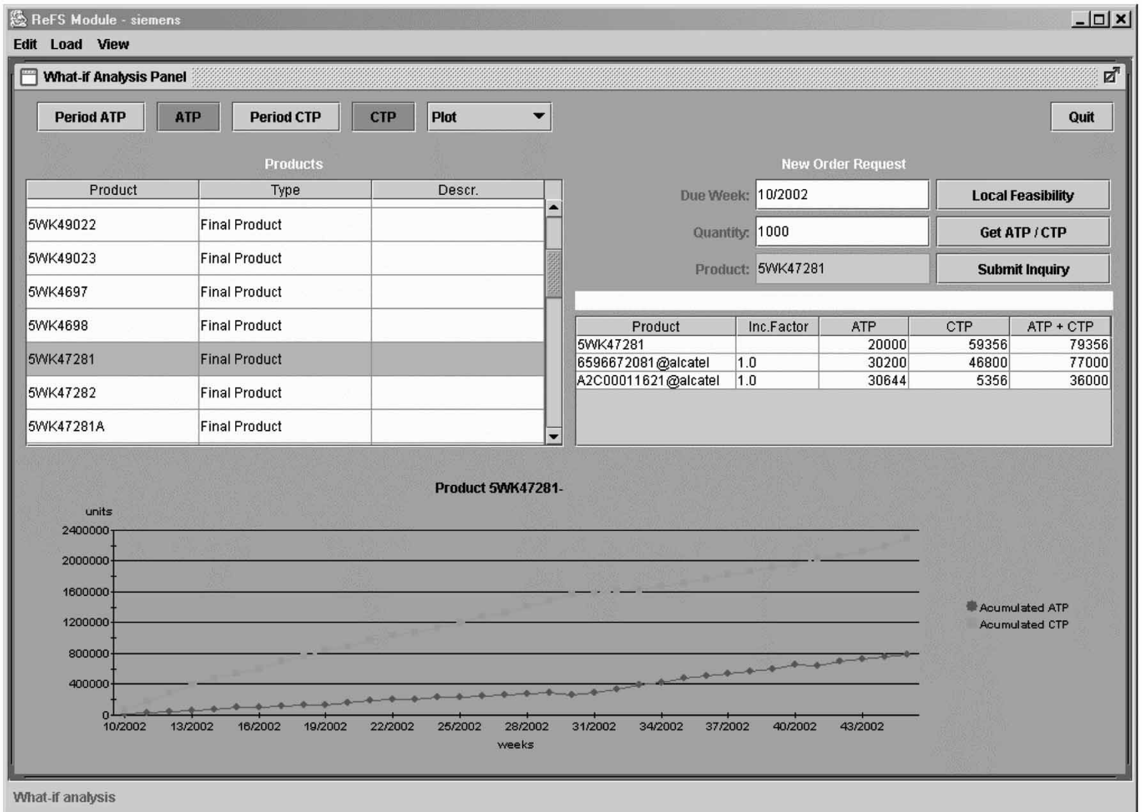

Figure 4. 'What-if Analysis' provided by the RefsAgent.

Cumulative $ATP_i = DiscreteATP_0 + \ldots + DiscreteATP_i - 1 + DiscreteATP_i$

Based on the way the human planner has configured the capacity model in his/her RefsAgent a simple algorithm (see next section) assigns the workload to each resource-centre. The difference of values between the resource-centre's effective capacity and its workload in a given period is the available capacity of the resource-centre for that period.

Another simple procedure is used to compute for each period a value that expresses the capability of the unit for future production, according to the way the production process of each meta-product was modelled, and taking into account the production-route of each meta-product, its process lead-time, its macro-operations, and associated resource-centres. The result of this computation is a set of values that represents the concept of Capable-To-Promise. The algorithm used to compute this (time) sequence of values is just the 'inverse' of the algorithm previously described [by considering quantities of the Master Production Schedule (MPS) in computing capacities].

ATP and CTP information is displayed by the ReFS agent to the company's Planner. Figure 4 shows one screen of the ReFS interface, used to perform some 'what-if' analysis and therefore support the decision-making process.

5.3. Master Production Schedule

A basic requirement for ReFS is that the MPS of each final product or product family is available (it is assumed that the company's ERP system is able to generate this plan). The MPS identifies, for each product, the quantity to be delivered by the production unit at the beginning of each period of the planning horizon.

Assigning loads to the production units, i.e. allocating quantities to the available capacities, is performed in the simple following way. The plan indicates that Q units of product P are going to be delivered in period D. The model starts by identifying what is the meta-product associated to P. Then, its production route and the sequence of macro-operations and resource-centres are identified. Following that routing, each resource-centre will be assigned the load equivalent to the quantity Q, starting at period D and proceeding backwards in time. The number of periods the model considers is equal to the process lead-time of the correspondent meta-product (which is an input to the model). For each period the model assigns a load to all the identified resource-centres with the same quantity Q divided by the number of periods in the process lead-time of the product.

6. Conclusions

The DSS presented in this paper has been installed and tested in three main nodes of a dynamic supply-network involving different sectors (automotive and semiconductors). The evaluation process has clearly proved the potential of the developed planning approach based on multi-agent technology supported by an Internet infrastructure.

A real-time customer order planning functionality was made available for each enterprise node of the supply chain, improving the global response time for new order requests and enhancing order promising. Furthermore, the developed solution leads to a better management of the real production capacity considered along the whole supply-network.

The growing importance of supply networks and networking activities obviously justifies further work in this area. We intend therefore to explore this line of research, by developing new approaches for the co-ordination among heterogeneous entities, and by investigating new strategies for decision making in complex supply chains and customer-driven manufacturing environments.

References

Aparicio, M., Chiariglione, L., Mamdani, E, McCabe, F., Nicol, R., Steiner, D., and Suguri, H., FIPA—Intelligent agents from theory to practice. 1999, *Telecom 99*, Geneva.

APICS, APICS Dictionary: eighth edition, 1995 (APICS).

Azevedo, A., Toscano, C., and Bastos, J., An intelligent agent-based order planning for dynamic networked enterprises. *Proceedings of the International Conference on Enterprise Information Systems ICEIS'2001*, Setúbal, Portugal, 2001.

Azevedo, A. L., Apoio à Decisão na Negociação de Encomendas em Redes de Empresas. PhD thesis (in Portuguese), Faculdade de Engenharia da Universidade do Porto, Portugal, 1999.

Booch, G., Rumbaugh, J., and Jacobson, I., *The Unified Modeling Language User Guide*, 1999 (Addison-Wesley: New York).

Collings, N. and Loeh, H., Co-OPERATE Project, Solution Scope. Technical Report, European Project IST-1999-12259, 2000.

Franklin, S. and Graesser, A., Is it an agent, or just a program?: a taxonomy for autonomous agents. *Proceedings of the 3rd International Workshop On Agent Theories, Architectures, and Languages* 1996, (Springer-Verlag).

JESS home page. Jess, the Rule Engine for the Java Platform, 2002. http://herzberg.ca.sandia.gov/jess.

Labrou, Y., Finin, T., and Peng, Y., Agent Communication Languages: the current landscape. 1999, *IEEE Intelligent Systems*, March/April.

Leach, N. and Ristic, M., *Capacity models for scheduling the virtual enterprise*, Technical Report. European Project X-CITTIC (IST EP 20544), 1998.

Nwana, H. S., Software Agents: An Overview. *Knowledge Engineering Review Journal*, 1996, **11**(3), 1–40.

O'Brien, P. D. and Nicol, R. C., FIPA—towards a standard for software agents. *BT Technology Journal*, 1998, **16**(3), 51–59. .

Poslad, S., Buckle, P., and Hadingham, R., The FIPA–OS agent platform: open source for open standards. *Proceedings of the 5th International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents*, 2000, pp.355–368.

Sun Microsystems Inc, Java Remote Method Invocation Specification, 2002, http://www.java.sun.com.

Sycara, K. P., Multiagent Systems. *AAAI AI Magazine*, 1998, **19**(2), 79–92.

W3C, The Extensible Markup Language (XML) specification, 2004. http://www.w3c.org/TR/REC-xml.