CO-OPERATIVE PLANNING IN DYNAMIC SUPPLY-CHAINS

Américo L. Azevedo[1,2], César Toscano[2], Jorge P. Sousa[1,2]
[1] *Faculdade de Engenharia da Universidade do Porto, Rua Roberto Frias, P-4200-465 Porto*
[2] *INESC Porto, Rua José Falcão, 110, 4050-315 Porto, Portugal*
email: {aazevedo,ctoscano,jsousa}@inescporto.pt

This paper describes an order planning system for dynamic supply-chains, addressing the requirements of a make-to-order *business environment. A distributed and decentralised information system, based on an architecture of agents and extensively using the internet, was designed and implemented to provide 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 CO-OPERATE European Project, and implements the functionality defined in the context of the 'request feasibility studies for the network' business solution.*

1. INTRODUCTION

The co-ordination and optimisation of complex and dynamic supply chains 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 realize significant returns through efficiency improvement, higher delivery reliability, better asset and capacity utilization, faster time to market and responsiveness. This is particularly important in the complex and highly dynamic environment of the automotive and electronics industries (Azevedo, 1999).

This paper describes part of a Decision Support System (DSS), for co-operative 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, that started on January 2000, aims at developing solutions to enhance the entire supply network, from the OEM manufacturer (automotive industry) to the original material supplier (semiconductor industry).

The DSS sits on top of the Enterprise Resource Planning (ERP) systems available in each company of the production network, and periodically pulls out information regarding forecast demand, backlog and the production schedule for each finished product in a given time horizon. A Capacity Model is available in each company of the network, for assessing the company's capabilities and for computing its production capacity, as well as for managing 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 regarding the evaluation of the customer inquiry 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 delivered to the customer for approval.

The paper is organised as follow. In the next section a briefly description of the business process is given. The architecture of the DSS is then presented and its functional model is described in the fourth section of the paper. Finally, in the fifth section some conclusions and current developments are briefly presented.

2. THE ReFS BUSINESS SOLUTION

The Co-OPERATE Project has defined a number of 'business solutions' or processes, in order to provide an answer to the 'global' needs of a network of companies, and focusing on the network aspects of the business processes between those companies.

It is one of these business solutions, called "Request feasibility studies for the network" (ReFS), that is dealt by the research described in this paper. Its main objective is to provide a fast response to incoming new orders (order promising) or to accommodate requests for large order changes which exceed the current availability allocation. When a request enters the enterprise network at a given node, its feasibility is checked internally taking into account the capacities and plans 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 where the initial request had occurred.

A number of "what if" questions related to the satisfaction of a customer request may be answered by the system. The following are examples of such questions:

- 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 to fully satisfy the customer's request?

In order to satisfy the requirements stated for this business solution, the response to customer inquiries should be given in useful time (i.e. much faster than traditionally) and 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.

3. SYSTEM ARCHITECTURE

3.1 Basic architectural model

The basic architectural principals behind the ReFS DSS system (which is the implementation of the ReFS business solution) were presented in (Azevedo et al., 2001). Basically, the supply-chain is modelled as a distributed system, without any central coordination of activities. A *software agent* (Nwana, 1996; Franklin et Graesser, 1996) representing each node in the network and automating tasks related to the *order promising* functionality, is based on an aggregate planning procedure. Interactions between companies in the network are modelled as 'speech acts' (Labrou et al., 1999) carried by the software agents (RefsAgent) that have the knowledge and expertise of their human counter parts in the company.

Figure 1 presents the main elements of the ReFS system, in UML - the Unified Modeling Language (Booch, 1999). The *Planner* symbol represents a human being in the company, with 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 Remote Method Invocation (RMI) protocol over the Internet Inter-ORB Protocol (IIOP) forms the protocol on which the connection mechanism was built.

According to the basic principles of Co-OPERATE, there can only be direct communication of information between companies with direct customer-supplier business relationships. Interaction between different instantiations of the RefsAgent follow this principle.

The *LegacySystem* type of components depicted in the right part of figure 1 represents any type of data source in the company from which the data needed by the *RefsAgent* should be obtained. Interactions between instances of a *RefsAgent* and a *LegacySystem* are accomplished through the periodic exchange of XML-based data.

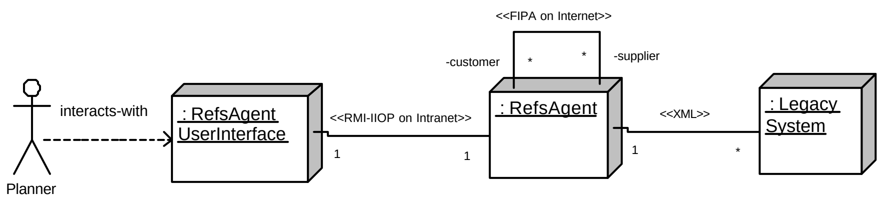

Figure 1 – Architecture of the ReFS system (UML deployment diagram).

3.2 Protocols and conversations

Interaction between the several software agents forming the multi-agent system of a given network of companies, was implemented according to the rules defined by the standardisation work of FIPA - the Foundation for Intelligent Physical Agents (Aparicio, 1999 and O'Brien, 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, which exhibit typical patterns of message exchange.

The RefsAgent implements the FIPA-request protocol (see figure 2). The Initiator entity corresponds to the RefsAgent that initiates the conversation and the Recipient entity corresponds to 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;

- check the feasibility of a new order request.

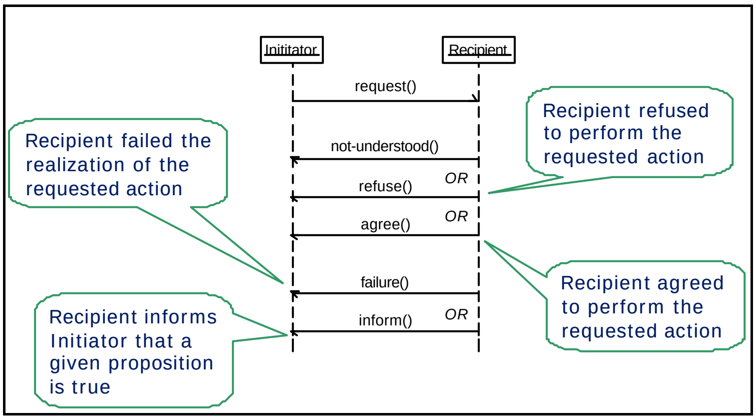

Figure 2 – FIPA-request protocol (UML sequence diagram).

3.2.1 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. The input to the protocol is composed by three items:

- *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:* identification of the 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. Calculation of the ATP and CTP values for the product referred by "productID" and for the week referred by "DeliveryWeek" is then accomplished.

3.2.2 Feasibility checking of new order requests

The protocol is instantiated by any RefsAgent in situations where the feasibility of a new order request is required to be analysed. 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 are the input of the instantiation of 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 the feasibility of a new order request is demanded;
- *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.

3.3 Internal organisation of the RefsAgent

The internal organisation 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 forms 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, which 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 visualisation tools.

The RuleEngine component gives the RefsAgent the ability to "reason" using knowledge supplied in the form of declarative rules and heuristics. Decision making to support internal processes is accomplished via this inference mechanism. A third-party tool, the Jess rules engine, 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.

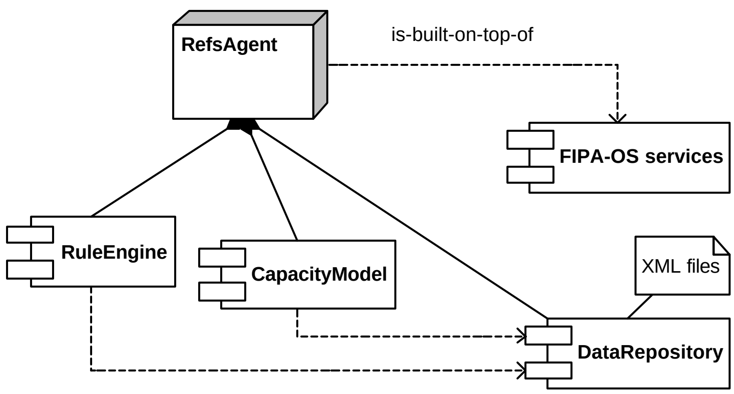

Figure 3 – RefsAgent internal organisation (UML component diagram).

The CapacityModel component is a full Java application. Its intended functionally and purpose are detailed in the next section of this paper.

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

4. FUNCTIONAL MODEL

4.1 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 measure of the production unit capacity, supports the creation of medium-long term plans, performs material management, and evaluates the local implications of a given customer order. Several key concepts form the basic elements of a Capacity Model:

- *Meta-Product* - Group of products (finished products or product families) that have similar characteristics concerning their production processes (here similarity is considered at the level of aggregation dealt by the model).
- *Macro-Operation* - A concept aggregating several sequential operations with similar characteristics. These operations are reduced to one basic macro-operation, with a single capacity. A macro-operation is a high level conceptual operation that characterises a set of real production operations.
- *Resource-Centre* - Aggregation of a set of shop floor work-centres. It represents 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.

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

The construction of an aggregate data model describing 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.

The following items form the basic data the model requires from the company's ERP system:

- *Product identification:* unique reference of each product (finished product or product family) supplied by the company to its suppliers;
- *Product bill-of-material:* required component parts for each product; 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:* quantity to be delivered by the production unit on each time period of the considered planning horizon.

4.2 Available-To-Promise and Capable-To-Promise

Available-to-Promise (ATP) information (APICS 1995, Azevedo 1999) is used for responding to new order feasibility requests. The ATP value is available for each product produced by the company in each time period of the planning horizon. Its value gives the uncommitted portion of the company's inventory and planned production of a specific product in the considered time period. Basically, it is 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 (where 0 denotes the first period of the horizon):

$DiscreteATP_0 = InitialStock + MasterProductionSchedule_0 - Backlog_0$

$DiscreteATP_i = MasterProductionSchedule_i - Backlog_i$

$\text{MasterProductionSchedule}_i$ $(i > 0)$ 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. Backlog_i corresponds to the quantity already committed to customer orders for time period i.

Cumulative ATP results from accumulating the above values, period by period along the planning horizon, as follows $(i >= 0)$:

$CumulativeATP_i = DiscreteATP_0 + \ldots + DiscreteATP_{i-1} + DiscreteATP_i$

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 correspondent resource-centres. These values are denoted by the expression Capable-To-Promise. The algorithm used to compute this sequence of values is just the inverse of the algorithm described in the previous section (considering quantities of the Master Production Schedule in computing capacities).

4.3 Master Production Schedule

A basic assumption is that the company's Master Production Schedule for each final product or product family it is available (ERP system generates this plan). This plan 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 done in the simple following way. The plan indicates that Q units of product P are going to be delivered in period DP. The model starts by identifying what is the meta-product of product P. Then, its production route and the sequence of macro-operations and resource-centres are identified. Following that routing, each resource-centre will then be loaded with the quantity Q, in a sequence of time periods: DP, DP-1, DP-2, … . 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 loads all the identified resource-centres with the same quantity Q divided by the process lead-time of the product.

5. CONCLUSIONS

The DSS presented has been installed and tested in three main nodes of a dynamic supply-network involving different sectors (automotive and the semiconductor industries). The evaluation process has clearly proved the potential of the approach based on multi-agent technology supported by the Internet infrastructure. Concerning the business solution briefly described in the paper, a real-time customer order planning functionality is 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.

6. REFERENCES

Aparicio, Manuel; Chiariglione, L.; Mamdani, E; McCabe, F.; Nicol, R.; Steiner, D and Suguri, H (1999). "FIPA – Intelligent agents from theory to practice", Telecom 99, Geneva.

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

Azevedo, Américo L.; Toscano, César and Bastos, João (2001). "An Intelligent Agent-based Order Planning for Dynamic Networked Enterprises", International Conference on Enterprise Information Systems ICEIS'2001, Setúbal, Portugal.

Azevedo, Américo Lopes (1999). "Apoio à Decisão na Negociação de Encomendas em Redes de Empresas", PhD dissertation thesis, Faculdade de Engenharia da Universidade do Porto, Portugal.

Booch, Grady; Rumbaugh, James and Jacobson, Ivar (1999). "The Unified Modeling Language User Guide", Addison-Wesley.

Franklin,Stan and Graesser, Art (1996). "Is it an Agent, or just a Program?: A taxonomy for Autonomous Agents", Proc. 3rd Int. Workshop On Agent Theories, Architectures, and Languages, Springer-Verlag,

Labrou, Yannis; Finin, Tim and Peng, Yun (1999). "Agent Communication Languages: the current landscape", IEEE Intelligent Systems, March/April.

Nwana, Hyacinth S. (1996). "Software Agents: An Overview", Knowledge Engineering Review Journal, v 11, no 3.

O'Brien, P. D., and Nicol, R. C (1998) "FIPA - towards a standard for software agents", BT Technology Journal, vol. 16, no. 3, 51-59 (July 1998).

Poslad, Stefan; Buckle, Phil; Hadingham, Rob (2000). "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, 355-368.