24 AGENT COMMUNICATION FOR SCHEDULING IN THE EXTENDED ENTERPRISE

Joaquim Reis [1♦], Nuno Mamede [2], Henrique O'Neill [1]
[1] ISCTE, [2] IST, Portugal

In previous work we have proposed how to model enterprise facilities (like factories, warehouses, etc.) in a multi-product production/distribution network, capacity management at those facilities, and agents which act as enterprise managers, taking decisions that affect the available capacity. In this article we describe a model for the internal activities of an agent for communication for multi-agent coordination in task scheduling in an integrated logistics environment.

INTRODUCTION AND EARLIER WORK

According to [Thomas 1996], investigation on coordinated planning in the areas of Management Science and Operations Research exists since the sixties. From then on, many researchers have studied multi-echelon inventory/distribution and production/distribution systems, although coordination in production/distribution scheduling has had less attention.

In most of these studies, which develop from the idea of the Economic Order Quantity, use mathematical models for a multi-echelon lot-sizing problem, and view the problem as one of optimization. They are more useful in a strategic/tactical approach, but not so useful for the short-term horizon of multi-agent activity scheduling. Their typical limitations include (see [Muckstadt 1993], or [Williams 1981], for instance): i) production is considered to be instantaneous, and lead times are zero, ii) final product demand is constant in time, iii) the network must follow pre-specified policies as a whole, iv) there is only one final product, v) there is no accommodation for unexpected events, vi) the network is considered a centralized system and there is only one objective function, vii) an optimal global solution is searched, viii) no preferences of each node of the network are considered.

Modern methods for scheduling include techniques like Constraint Satisfaction

♦ ISCTE, Dpt. Ciências e Tecnologias de Informação, Avenida das Forças Armadas, 1600 Lisboa, Portugal, tel. + 351 1 7903261 , fax + 351 1 7964710 , e-mail: Joaquim.Reis@iscte.pt

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: 10.1007/978-0-387-35577-1_37

L. M. Camarinha-Matos et al. (eds.), *Infrastructures for Virtual Enterprises*

[Kumar 1992], [Prosser 1991], and concepts like Constrained Heuristic Search [Fox 1989], Micro-Opportunistic Scheduling [Sadeh 1994]. These methods involve a mix of backtracking search and constraint propagation techniques, using specific heuristics at specific choice points and they have helped to build large scale and complex scheduling systems (see [Atabakhsh 1991] or [Zweben 1994], for instance).

Today, challenges are posed to enterprises by competitive pressure and market globalization. They are forced to form supply chains with short response times to customer needs, and to remain competitive they must reduce operating costs while continuously improving customer service [Thomas 1996]. As a result we are moving towards a more coordinated and integrated design and control of the actors of the supply chain. Also, with recent advances in communication and information technology, enterprises have the opportunity to reduce operating costs by coordinating planning and scheduling across stages in the supply chain. Concepts like Quick Response, Accurate Response, Integrated Supply Chain Management, Agile Scheduling, Virtual or Extended Enterprise (see, for instance, [Ross 1996], [Fisher 1994], [Fox 1993], [Rabelo 1996] and [O'Neill 1996], respectively), are being used to describe management models for cooperative supply chain networks. For instance, in the Extended Enterprise (EE) paradigm [O'Neill 1996], [Sackett 1994], a group of interdependent highly specialized enterprises agree in cooperating to make available, at the right times, the demanded quantities of final products. In this scenario, the entire *business process* (including production, storage, transportation) of a product is performed by a network of geographically distributed *business units*, owned and managed by enterprises. These are connected by an electronic network that supports the information interchange needed for the effective coordination. A *supervision unit* assumes the role of integration among business units. It is a team, where business units are represented, with some authority to define global policies and medium/long-term planning and forecasting. The EE paradigm tries to respond to challenges posed. We felt it could give some conceptual support in modeling the cooperative environment of logistics task scheduling, and we borrowed some ideas from this paradigm, and from some experience of its practical application, namely in the Esprit project AITEAR [AITEAR 1997a], [AITEAR 1997b].

Also, in our work, we view the production/distribution network scheduling activity in the EE framework through the Distributed Artificial Intelligence Multi-Agent paradigm (see, for instance, [Bond 1988], [ICMAS 1996] or [O'Hare 1996]), as a *multi-agent coordination* problem. The agents, while cooperating to avoid violation of hard scheduling constraints (capacity constraints and temporal constraints like predefined task precedences and agreed due-dates), compete for their own interests to satisfy their scheduling preferences.

BASIC ONTOLOGY

We propose a two-level model for the EE scheduling environment including: the *physical level*, respecting to the production/distribution network, and the *virtual*, or *agent level*, respecting to agents which own and manage the physical resources of the physical level. In the present section we summarize previous work that describes these modeling aspects (for details see [Reis 1998 and also Reis 1999]).

The Physical Level

The *physical network*, or simply *network*, models a multi-product production/distribution network, which is an acyclical network composed of physical renewable resources, referred to by *physical nodes*, or *nodes*. The *physical arcs* of this network are client-supplier relationships between pairs of nodes. The nodes are capable of executing logistic *tasks* (production, transportation, storing), making available *output products* for other nodes, or for the outside of the network, if they are *retail nodes*. There are three types of nodes, according to the type of tasks they can execute: the *store* (S), the *producer* (P) and the *transporter* (T) node. The set of output products of the retail nodes defines the set of network *end products* (with

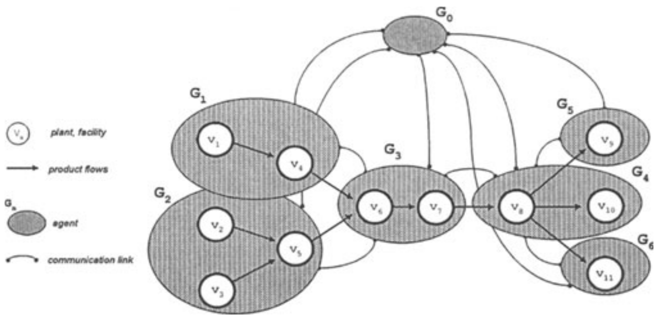


Figure 1- An EE network example.

demand from the outside). Nodes consume *input products* (or *materials*) made available by other nodes in the network, or by the outside of the network if they are *raw-material nodes*. Except for raw-material and retail nodes, every node has a limited *capacity* and maintains an internal capacity state by recording the available capacity and the capacity being used by, and shared amongst, different tasks.

In Figure 1 we graphically represent a simple EE network. The physical network is composed of nodes labeled v_1 to v_{11}, with raw-material nodes v_1, v_2 and v_3, and retail nodes v_9, v_{10} and v_{11}. Arrows represent physical arcs.

Two types of capacity exist: *storing capacity* (available in store nodes), and *processing capacity* (available in producer, and transporter nodes). The former is constrained in space, and is defined by a quantity, the second latter constrained in time and is defined by a rate.

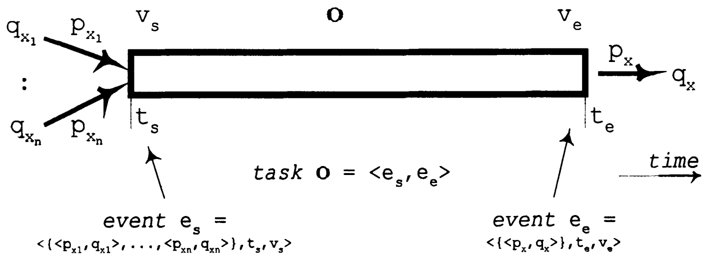
The Agent Level

In the *agent level*, a network of *manager agents*, or *virtual nodes*, exist. Each physical node is owned and managed by one agent and each agent owns and manages one or more nodes and the client-supplier relationships among nodes are extended to the agents. Agents are actors in the environment, taking scheduling decisions resulting in tasks being executed in the near future. Agents are asked to schedule tasks in their nodes with the purpose of making available products to clients (or to the outside). A special virtual node, the *supervision agent*, assumes an integration role and supplies global demand forecasts and proposes plans in a medium/long-term perspective. Bi-directional *communication links*, the *virtual arcs*, connect pairs of agents allowing the communication necessary to coordination. This all makes the *agent*, or *virtual*, *network*.

In the EE network example of Figure 1, agents, labeled G_1 to G_6, are represented as shaded areas, with the respective nodes inside. G_0 represents the supervision agent and virtual arcs are represented by curved arcs linking agents.

Requests, Events and Tasks

We now focus in the internal activity of an agent for inter-agent communication for short-term temporal planning, *i.e.*, scheduling, and develop a little further from work already published (see [Reis 1999]). We assume that coordination among agents is to



p=*product*, q=*quantity*, t=*time*, v=*place/node*, e=*(internal) event*, o=*task*
indexes: s=start, e=end

Figure 2- A task and its internal (private of the agent) representation.

be achieved through communication between pairs of client-supplier manager agents and the supervision agent won't be involved.[1] Each agent will interact with other, supplier and client, agents by sending and receiving *messages*. For a product request (*i.e.*, an agent order) a message typically contains product, quantity, time and place (*i.e.*, node) information.

We refer to *events* as elementary records of information containing product, quantity, time and place information. In the logistics space of multi-product production/distribution environment, and for multi-agent scheduling purposes, everything that happens, or could possibly happen, can be, at least in part, described by one, or more, events. For instance, a product request, a task start or a task end, have associated product, quantity, time and place information, and they can be represented through the event concept. More formally, we define an event, e, as a triple of the form:

$<\{\ldots,<p_i,q_i>,\ldots\},t,v>$ with $p_i \in \boldsymbol{P}$, $v \in \boldsymbol{V}$

where the first element of the triple is a set of finite product-quantity pairs — p_i denotes a product ($\boldsymbol{P}$ is the set of all products in the network) and q_i the quantity of product p_i (any number) —, t is a temporal instant (an integer) and v is the a network node identifier ($\boldsymbol{V}$ is the set of all network node identifiers).

For requests, events only have one product-quantity pair, the quantity will be a

[1]The supervision agent is to be involved in medium/long-term planning and forecasting activities only.

positive number[2] and the node information will be the identifier of the supplier node. For instance, suppose agents G_a and G_b own nodes v_a and v_b, respectively, that G_a is a client of G_b and that product p_x is consumed by v_a and produced by v_b. The content for a request message from G_a to G_b, could be an event like $<\{<p_x, q_x>\}, t, v_b>$. This conveys the essential pieces of information of *what* (product p_x), *how much* (quantity q_x), *when* (time t), and *where* (place/node v_b).

We assume here each agent owns only one physical node of the network,[3] with no alternative suppliers for the same input product. For a capacity agent (neither a raw-material agent nor a retail agent) each request from a client will call for a task to be executed on the agent node, during the execution of which the capacity of the node is lowered by an amount proportional to the amount of product requested.

Tasks can also be represented using the event concept. A task, in its *internal representation* (private of the agent), is defined as a pair of events $<e_s, e_e>$, where e_s is the task *start event* and e_e is the task *end event*. For instance, task O, graphically represented in Figure 2 delivers q_x units of product p_x, in time t_e, in place v_e, and absorbs $q_{x_1}, \ldots, q_{x_n}$ units of products $p_{x_1}, \ldots, p_{x_n}$, respectively, in time t_s, in place v_s. An agent would internally represent task O by using the event pair $<<\{<p_{x_1}, q_{x_1}>, \ldots, <p_{x_n}, q_{x_n}>\}, t_s, v_s>, <\{<p_x, q_x>\}, t_e, v_e>>$.

Scheduling a task calls for one or more requests of materials to the suppliers. If the agent owns a producer node, and this node produces a product p_x, which is made of products $p_{x_1}, \ldots, p_{x_n}$ ($n \geq 1$) then, for each request of product p_x from a client, the agent will generate one request of each product, $p_{x_1}, \ldots, p_{x_n}$, to the respective suppliers. If the agent owns a store node, or a transporter node, and the node delivers product p_y, then, for each request of product p_y from a client, the agent will generate a request of product p_y to the respective supplier. A task, in its *external representation* (shared with other agents), has associated information which is also described by the event concept, namely the event of the request that came from the client agent (which caused the task to be scheduled) and the events of the requests of materials that the agent sent to the suppliers. These events are termed *external events* (they are shared by the agent with other agents), as opposed to the events of the internal representation of the related task, which are termed *internal events*.

Figure 3 represents the internal and external events of task O in a time line. e_{ee} and $e_{ss,i}$ ($i=1, \ldots, n$) are the external end event and the external start events

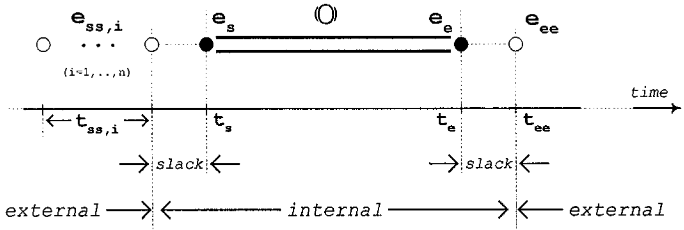


Figure 3- Events and temporal information associated to task O.

[2]Negative numbers are reserved to specify a demand that waits to be satisfied at a retail node.

(one for each input product p_i); e_e and e_s are the internal end event and the internal start event of the task, respectively (also represented in Figure 2).

THE AGENT ACTIVITY

Agents take scheduling decisions by allocating capacity of their nodes to tasks for future execution. The activities that an agent performs can be: a) *external* (or *communicative*), through which sequences of communicative acts (message exchanging) take place amongst agents, and b) *internal*, through which the internal state of the agent is maintained.

External Activities

In external activities an agent acts:
1) as a supplier by accepting supply requests from clients;
2) as a client by sending supply requests to its suppliers;
3) by exchanging rescheduling request, cancellation or satisfaction messages with clients and suppliers;
4) by exchanging supply query and reply messages with clients and suppliers.[4]

Coordination amongst manager agents is achieved through communication between pairs of client-supplier agents by using sequences of messages (KQML like [Finin 1994]), with some expected structure in the sequence, termed agent *conversations*, or *communication threads*. Through 1), 2) and 3) a pair of client-supplier agents can engage in a *supply request* conversation for a product supply, establish a local schedule by agreeing on shared (external) events, and eventually agree on rescheduling or cancel the supply request. Scheduling, or rescheduling, is accomplished by local agreements on the (time) values of shared events which, for each agent, impose temporal limits on tasks. Through 4), which is optional, a pair of client-supplier agents can engage in a *supply query* conversation. This is the way an agent can ask about the possibility of the supplier satisfying a supply request (but without any commitment between the two), or ask if a client or a supplier can accommodate for a rescheduling request of a previously accepted supply request.

Internal Activities

Agent internal activities are:
1) maintaining an up-to-date state of the interactions with the outside world;[5]
2) scheduling, unsche-duling and resche-duling tasks when requested, and when necessary and possible.

Internal activities 2) are accomplished by allocating (or deallocating) capacity on the nodes, for which the agent is responsible, for the necessary tasks, and

---

[3] The most frequent situation, which is equivalent in terms of inter-agent requests, is each agent owning a node with processing capacity (a P or T type node), and a pair of nodes with storing capacity (S type nodes), one at the upstream side and the other at the downstream side of the first node.

[4] Additional external activities, not discussed here, include exchanging long/medium-term planning and forecasting global information for extended temporal horizons with the supervision agent.

[5] Additional internal activities (not discussed here) include maintaining up-to-date information of long/medium-term capacity reservation for forecasted global demand, according to information from the supervision agent and actual supply requests from clients.

maintaining an internal representation of the capacity used in each node output product as well as capacity available along time. Internal activities 1) are accomplished by maintaining an internal representation of the state of the presently active conversations (see ahead) in which the agent is involved with other agents.

COMMUNICATING FOR SCHEDULING AND RESCHEDULING

We now propose a simple protocol for multi-agent scheduling by defining the structure of the agent conversations. Without considering rescheduling, for an agent acting as a supplier, for each supply request from a client, the agent communication state will go through the conversation represented by the state diagram in Figure 4. In this figure, the arc labels correspond to types of messages. For the agent acting as a client the state diagram is similar (just exchange the S and R prefixes in the arc labels). The states in Figure 4 are briefly described:

1) initial state;
2) a supply request from a client was received;
3) the request was accepted by the agent;
4) final state: the request was rejected, or satisfied, or canceled by the agent, or canceled by the client.

Rescheduling

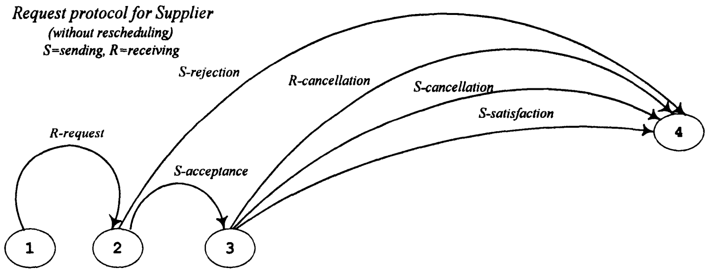


Figure 4- Conversation for a supply request (from a client), for the agent as a supplier (without rescheduling).

For the rescheduling part, we have, additionally, the state diagram in Figure 5 (complementary to the one in Figure 4), which states we describe:

5) a rescheduling-request (for a previously accepted supply request) was sent by the agent;
6) or 8) the rescheduling-request is pending;
7) a rescheduling-request (respecting to a previously accepted supply request) was received by the agent.

Arc traversing is triggered by the exchanging of a message. For instance, state 2 (in Figure 4) is attained because a request message was received from a client. Also, some activity goes on. For instance, in state 3, the agent has scheduled the necessary task for the request satisfaction, by having available capacity allocated on its nodes, and possibly has sent the necessary supply request messages for materials needed to suppliers.

In state 5, a rescheduling request was made by the agent to a supplier or to a client; in state 7 a rescheduling request was made by a supplier or a client to the agent. Rescheduling requests can occur when event times don't fit in the temporal horizon, and they can be accepted (if there is enough temporal/capacity slack), or stay pending (to be accepted if necessary rescheduling requests to be made to other clients or suppliers downstream or upstream are accepted), or rejected (otherwise). In tight scheduling situations rescheduling will become necessary (*e.g.*, to avoid rejections or cancellations) and islands of rescheduling can propagate and spread over all the network. This can result in agents trying to reduce flow times by relaxing

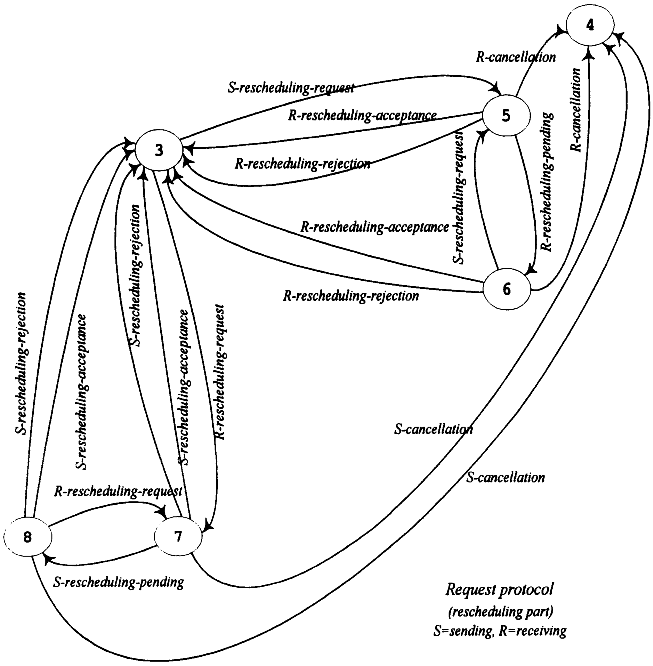


Figure 5- Conversation for a supply request, for the agent (rescheduling part only).

progressively their scheduling preferences, or agreeing locally in task splitting and, in more extreme situations, in canceling accepted requests.

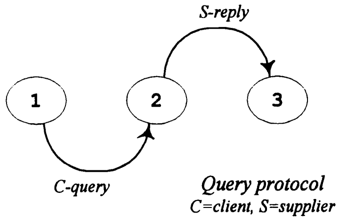


Figure 6- Conversation for a supply query.

Querying

Anytime, a supply query conversation can be triggered by the reception of a supply query from client agent, see state diagram in Figure 6. The corresponding supply reply message from the supplier can contain one or more event alternatives, listed in (supplier) preference order. If we view the scheduling problem as Distributed Constraint Satisfaction Problem (DCSP, see [Yokoo 1992] and [Yokoo 1993]), with events corresponding to the DCSP variables, and assuming agents are cooperative, in a rescheduling situation the client agent can, if possible, propose rescheduling to the supplier with event time and quantity values according to this preference information, possibly minimizing future backtracking. The same can apply for supply requests.[6]

Agent Internal Structures

The data structures needed for keeping track of internal activities, of an individual agent, related to inter-agent communication, are represented in Figure 7 (except for supply queries) in the part labeled 1. This is divided in two almost symmetrical sides: one, the downstream side, for the communication with client agents (labeled *output of products*), and the other, the upstream side, for communication with supplier agents (labeled *input of products*). Part labeled 2, in Figure 7, respects to maintaining a representation of capacity in use and available capacity along time in the agent nodes.

The component labeled *scheduling behavior* is responsible for the way the agent takes its scheduling decisions (and this includes its scheduling preferences). This component can merely contain some dispatch rules used in capacity allocation for tasks (in that case we will have a more reactive agent) or can involve a more sophisticated scheduling system, with planning and reasoning capabilities (in that case we will have an agent of a more predictive, or cognitive, kind).

Agent Internal State

For an individual agent, any incoming supply request message from a client agent triggers the creation of a new client supply request conversation, with the respective event information being stored in the structure *requests from clients made* (in Figure 7, see the part labeled 1). If the agent accepts the request,[7] that information goes to *requests from clients accepted* and the respective task is scheduled (with the appropriate information updated in the capacity state of the *physical nodes*). Then, the necessary event information for requests of materials to suppliers is created and put in *requests to suppliers to make.*[8]

When task priorities are involved and there is no enough capacity, new requests with higher priorities can be accepted, even at the cost of canceling previously accepted ones, and for this the agent uses its *requests from clients accepted and in conflict* and *requests from clients to cancel* structures. This latter structure is also used for storing to-be canceled client requests, whenever the respective task cannot be executed because of a cancellation, or rejection, of a material request from a supplier.[9] The structures *agent rescheduling requests* (*made* and *pending*) and *client rescheduling requests* (*made* and *pending*) accommodate for rescheduling of the agent, or the client initiative, respectively. Whenever a request is satisfied or canceled, the respective event information is removed and the corresponding satisfaction message sent to the client.

---

[6] However, for very high problem dynamism, this information can become rapidly out-of-date.

[7] The agent must obviously reject the requests with past due-dates.

[8] Depending on the temporal values chosen by the agent for the internal end and start events for a task and for the corresponding external start events, there can be temporal slacks inserted in the task network schedule by the agent (see Figure 3). This can be good because it can avoid too much rescheduling in the future, or bad because it will inflate network flow-time.

[9] From the short-term scheduling perspective, and for now, we are not worried with the economic aspects associated to cancellations (*e.g.*, unused materials).

For each event in *requests to suppliers to make*, the agent creates a new supplier supply request conversation, sends the appropriate supply request message to the respective supplier and puts the event information in *requests to suppliers made.* The use of the remaining structures (in the *input of products* side) in the rest of the process, when following a supplier supply request conversation, is rather symmetrical to the one followed in a client supply request conversation.

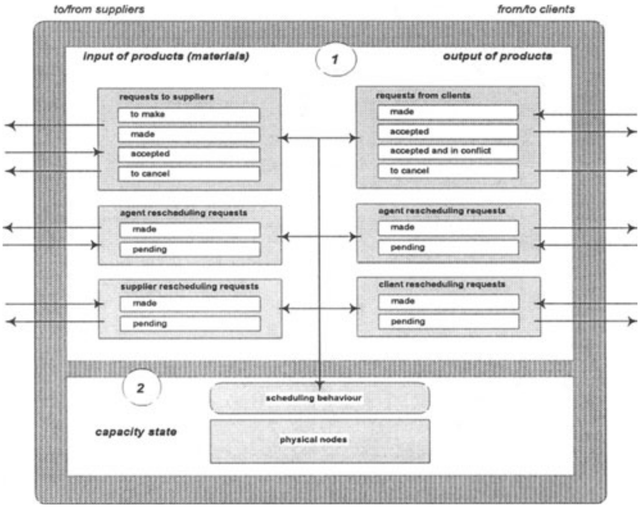


Figure 7- Internal agent structures for internal activity support.

MULTI-AGENT SCHEDULING DECISIONS AND PERSPECTIVES

Some decisions constitute choice points for an individual agent in the network scheduling. These can be, for instance:

1) when facing a group of requests from clients, which request to consider first;
2) when considering a request from a client, decide to accept the request or not. This primarily depends on whether, or not, there is enough capacity to allocate for a task to satisfy the request, and in the case of enough capacity slack, agent scheduling preferences will govern the choices made;
3) for a set of requests from clients causing capacity conflicts, choose which ones are to be canceled, or to be asked for rescheduling.

In the multi-agent EE network scheduling environment, the way scheduling decisions are taken, make up the scheduling behavior of each agent. We identified two dimensions, or perspectives, which can be used to bias the scheduling behavior and constitute degrees of freedom to explore in future work. These are:

Local vs. global perspective (information in the space/place dimension):

1) local perspective - each agent acts individually, by responding to its clients requests and by sending requests to its suppliers;
2) global perspective - each agent has access to global network information and can use it for more coordinated network scheduling.[10]

Short vs. medium/long-term perspective (information in the time dimension):

1) short-term perspective - each agent reacts to its clients requests;

---

[10]*E.g.*, information propagated by agents about unexpected events can be used to adjust predictively agent individual schedules accordingly; or the medium/long-term forecasting and planning information for the whole network supplied by the supervision agent for each network agent.

2) medium/long-term perspective - each agent makes use of information for planning its tasks for periods of time in the future, and the answers it gives back to the requests made by its clients (namely with respect to dates) are biased by this information.[11]

Additionally, other aspects also seem interesting, like: a) scheduling only (no rescheduling) vs. scheduling and rescheduling; b) without vs. with task splitting; c) without vs. with batching (task aggregation) and d) without vs. with task priorities. Although our work has focused in the local, short-term perspectives (with rescheduling and without task splitting, batching and priorities) we plan to extend it in the near future.

CONCLUSION AND FUTURE WORK

We described a model for the activities of an agent, with emphasis on inter-agent communication, for multi-agent coordination in task scheduling in the EE integrated logistics environment.

A computational system is being developed, based on the ideas exposed, in an object oriented computer language. Current work is also being done on model refinement (namely inter-agent protocols and agent scheduling behavior) and on applying DCSP techniques to the dynamic multi-agent scheduling problem. We are also elaborating the internal "mental" model of the individual agent and structuring it according to the so-called Belief-Desire-Intention architectures (see, for instance, [Georgeff 1998]), which are appropriate for agent operation in dynamic and unpredictable environments (which is the case of the EE scheduling problem).

Future work will involve: i) completing the model by including medium/long-term planning and involving the supervision agent; ii) building a simulator of the EE scheduling environment and make some experimental work, in particular with respect to the perspectives and aspects mentioned in the previous section, and to network scheduling with heterogeneous agent scheduling behavior iii) build a decision support system to support scheduling and planning activities in the context of the EE.

REFERENCES

AITEAR 1997a CIMI, 1997, *Final Report on Assessment of State-of-the-Art in Accurate Response*, Project AITEAR, CIMI/Cranfield Univ., Cranfield, Beds, MK43 0AL, UK, April 1997.

AITEAR 1997b Benz, Harald, 1997, *Working Definition of EE and ARM for the AITEAR Project*, Discussion paper, AITEAR Project, FhG-IAO, Stuttgart, August 1997.

Atabakhsh 1991 Atabakhsh, H., 1991, *A Survey of Constraint Based Scheduling Systems Using an Artificial Intelligence Approach*, Artificial Intelligence in Engineering, 6 (2) 1991.

Bond 1988 Bond, Alan H.; Les Gasser (eds.), 1988, *Readings in Distributed Artificial Intelligence*, Morgan Kaufman Publishers, Inc., San Mateo, California, 1988.

Finin 1994 Finin, Tim; Fritzson, Richard; McKay, Don; McEntire, Robin, 1994, *KQML as an Agent Communication Language* (from the Web site http://www.cs.umbc.edu/kqml/papers/ kqml-acl-html/root2.html).

Fisher 1994 Fisher, Marshall R.; Hammond, Janice H.; Obermeyer, Walter R.; Raman, Ananth, 1994, *Making Supply Meet Demand*, Harvard Business Review, May-June 1994.

Fox 1993 Fox, Mark S.; Chionglo, John F.; Barbuceanu, Mihai, 1993, *The Integrated Supply Chain Management System*, Department of Industrial Engineering, University of

[11]This information can come from a local perspective (*e.g.*, the agent builds it based on its product outputs in past periods) or from a global perspective (supplied by the supervision agent).

Toronto, Canada (from the Web site http://www.ie.utoronto.ca/EIL/iscm-descr.html).

Fox 1989 Fox, Mark S.; Sadeh, Norman; Baykan, Can, 1989, *Constrained Heuristic Search*, Proceedings of the 1989 Int. Joint Conf. Artificial Intelligence (IJCAI89) 309-315, 1989.

Georgeff 1998 Georgeff, M.; Rao, A., 1998, *Rational Software Agents: From Theory to Practice*, In Agent Technology, Foundations, Applications, and Markets, Jennings, Nicholas R.; Wooldridge, Michael J. (eds.), Springer Verlag, 1998, 139-160.

ICMAS 1996 ICMAS 1996, 1996, *Proceedings of the Second International Conference on Multi-Agent Systems (Contents)*, ICMAS-96, AAAI Press, Menlo Park, California, 1996.

Kumar 1992 Kumar, Vipin, 1992, *Algorithms for Constraint-Satisfaction Problems: A Survey*, AI Magazine, 13 (1) 1992

Muckstadt 1993 Muckstadt, John A.; Roundy, Robin O., 1993, *Analysis of Multistage Production Systems*, In Handbooks in Operations Research and Management Science (Nemhauser, G.L.; Kan, A.H.G. Rinnooy, eds.), V. 4: Logistics of Production and Inventory, Graves, S.C.; Kan, A.H.G. Rinnooy; Zipkin, P.H. (Eds.), North-Holland, 1993,Ch.2,59-131.

O'Hare 1996 O'Hare, G.M.P.; Jennings, N.R., 1996, *Foundations of Distributed Artificial Intelligence*, John Wiley & Sons, Inc., 1996, New York, USA.

O'Neill 1996 O'Neill, H.; Sackett, P., 1996, *The Extended Enterprise Reference Framework*, Balanced Automation Systems II, Luís M. Camarinha-Matos and Hamideh Afsarmanesh (Eds.), 1996, Chapman & Hall, London, UK, pp 401-412

Prosser 1991 Prosser, Patrick, 1991, *Hybrid Algorithms for the Constraint Satisfaction Problem*, Technical Report AISL-46-91, Univ. Strathclyde, Glasgow, Scotland, UK, Sept. 1991

Rabelo 1996 Rabelo, R.J.; Camarinha-Matos, L.M., 1996, *Towards Agile Scheduling in Extended Enterprise*, Balanced Automation Systems II, Luís M. Camarinha-Matos and Hamideh Afsarmanesh (Eds.), 1996, Chapman & Hall, London, UK, pp 413-422

Reis 1998 Reis, J.; Mamede, N.; O'Neill, H., 1998, *Ontologia para um Modelo de Planeamento e Controlo na Empresa Estendida*, IBERAMIA'98.

Reis 1999 Reis, J.; Mamede, N., 1999, *What's in a Node, Nodes and Agents in Logistic Networks*, 1st Int. Conf. Enterprise Information Systems (ICEIS'99), Mar 1999, Setubal, Portugal.

Ross 1996 Ross, David F., 1996, *Designing an Effective Quick Response System*, APICS - The Educational Society for Resource Management, 1996 (from the Web site http://www.apics.org/SIGs/Articles/designin.htm).

Sackett 1994 Sackett, P.; Wortmann, H.; Brown, J., 1994, *Manufacturing Business Challenges in the late 1990's*, Proc. 1st. SCMA Conf. Outstanding Business Success in Manuf., London.

Sadeh 1994 Sadeh, Norman, 1994, *Micro-Oportunistic Scheduling: The Micro-Boss Factory Scheduler*, Intelligent Scheduling, Morgan Kaufman, 1994, Cap. 4

Thomas 1996 Thomas, Douglas J.; Griffin, Paul M., 1996, *Coordinated Supply Chain Management*, European Journal of Operational Research, 94 (1996) 1-15.

Williams 1981 Williams, Jack F., 1981, *Heuristic Techniques for Simultaneous Scheduling of Production and Distribution in Multi-Echelon Structures: Theory and Empirical Comparisons*, Management Science, Vol. 27, No. 3, March 1981, 336-352.

Yokoo 1992 Yokoo, Makoto; Durfee, Edmund; Ishida, T.; Kuwabara, K., 1992, *Distributed Constraint Satisfaction for Formalizing Distributed Problems Solving*, Proceedings of the 12th IEEE International Conference on Distributed Computing Systems, 614-621, 1992.

Yokoo 1993 Yokoo, Makoto, 1993, *Dynamic Variable/Value Ordering Heuristics for Solving Large-Scale Distributed Constraint Satisfaction Problems*, Proc. 12th Int. W. on Distributed Artificial Intelligence, Hidden Valley, Pennsylvania, May 19-21, 1993, 407-422.

Zweben 1994 Zweben, Monte; Fox, Mark S., 1994, *Intelligent Scheduling*, Morgan Kaufmann Publishers, Inc., San Francisco, California, 1994.