

Scheduling the Supply Chain by Teams of Agents

J. Sauer, H.-J. Appellrath

Universität Oldenburg, FB Informatik
Escherweg 2, D-26121 Oldenburg
sauer@informatik.uni-oldenburg.de

Abstract

When a supply chain is established supply chain management (SCM) needs supporting tools for the tasks of operative planning, scheduling, and coordination. These tasks have to be performed not only on the level of the enterprises involved but also within their established business entities (e.g. plants, areas, resource groups, resources) in which the high level schedules have to be put into operation. Most approaches of SCM favor a hierarchical coordination of the supply chain together with powerful algorithmic solutions for the mainly predictive scheduling tasks. These approaches are lacking the incorporation of feedback from lower levels and possibilities of reactive scheduling. Thus flexibility and reactivity are main issues to be improved.

In this paper we will present an approach using teams of cooperating agents in a hierarchical as well as heterarchical way to overcome the problems mentioned. To simplify the generation of such a hierarchy of agents, we develop a framework for scheduling agents that contains the basic features of an agent e.g., event handling, data storage, communication, and possibilities for appending scheduling algorithms and specific cooperation mechanisms. The framework will be described and an example shall illustrate how it is used to build teams of cooperating agents on several levels of the scheduling hierarchy.

1. Introduction

Decreasing transaction costs, advanced control of processes and thinking in profit centers increasingly lead to companies outsourcing those parts of the creation of value without core competencies. The growing force to shorten delivery periods and product innovation cycles while at the same time increasing the rates of return induced by globalized markets requires an intensified cooperation of all companies along the inter-company supply chain ([1]). The cooperation shall be supported by systems of cooperating agents

which should be easily implemented by using actual software engineering principles like frameworks.

Supply Chain Management

A *supply chain* is a network of facilities and distribution options that performs the functions of procurement of materials, transformation of these materials into intermediate and finished products, and the distribution of these finished products to customers [2; 3]. Supply chains exist in both service and manufacturing organizations, although the complexity of the chain may vary from industry to industry and firm to firm.

In a simple supply chain for a single product the raw material is procured and transported from vendors to the manufacturing sites, transformed into finished goods, then transported to distribution centers, and ultimately, sold to customers. Realistic supply chains have multiple end products with shared components, facilities and capacities. The flow of materials can be shown as a network from the suppliers to the manufacturer and the retailer using various modes of transportation.

When a supply chain shall be implemented successfully we are faced with a bunch of problems. Among them are:

- Because the companies involved (marketing, distribution, planning, manufacturing, and the purchasing organizations) typically act independently there is not a single, integrated plan for the whole organization, there are as many plans as businesses. Additionally, the organizations have their own objectives and these are often conflicting.
- Not only the efficient materials flow is important within the supply chain, more important are the flow of information and the coordination of the interacting business entities of several independent companies.
- There is a massive exchange of information which presupposes that the companies involved can trust

each other, which means that the information provided has to be correct, actual and complete.

Supply chain management is the combination of strategies and tools to integrate all the units involved in order to achieve a maximum of profit out of the supply chain, i.e., reduce costs, reduce wasted time, meet delivery dates etc.

Characteristic for supply chain management is the strategic, long-term cooperation of companies as well as the small number of suppliers for a particular product. Cooperation according to the supply chain management approach relies on massive exchange of information, which presupposes trust between the partners within the supply chain and the long-term abolishment of information barriers between the individual companies.

The decisions to be made within supply chain management are characterized in different categories which are interrelated. Most often we find strategic, operational and tactical categories which are mainly time dependent separations of the decision tasks.

Strategic decisions are made typically over a longer time horizon. These decisions are also called configuration decisions. Here the supply chain network is designed with the choice of partners, the placement of production facilities, stocking points, and distribution centers. Another main strategic decision which is closely related to the location decision is what products to produce, in which plants, and how to provide the customer markets.

Tactical decisions are made over a mid-range time horizon. Here longer term planning and scheduling decisions have to be taken, e.g., the master production schedule for the whole supply chain which also creates the master schedules for transportation and storing, and inventory decisions or make or buy decisions for all kinds of tasks.

The operational decisions are short term, and focus on activities over a day-to-day basis. Here the detailed scheduling and rescheduling activities within the plants are performed, e.g., production scheduling in different production sites, transportation scheduling within the transport facilities, storage scheduling within warehouses.

Within this paper we will focus on the tactical and operational scheduling decisions to be made within the business units which belong to the supply chain.

Agent Systems

Software *agents* represent a software development paradigm which is appropriate for distributed problem solving. They are employed in numerous systems of

distributed artificial intelligence (DAI). In common linguistic usage, an agent is everyone who acts on behalf of another [4].

In DAI an agent is defined as “a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives” [5]. Therefore, a fundamental property of an agent is *autonomy*: an agent operates without direct interference by humans or other systems, and has control over its behavior and its internal state. The concept of an *intelligent agent* extends this definition by the capability of acting flexibly, whereby the notion of flexibility comprises three characteristics:

- *reactivity*: agents perceive their environment and react timely and appropriately to changes within this environment;
- *pro-activeness*: agents do not only react to observed changes within their environment, but are capable of taking the initiative in a goal-directed fashion;
- *social ability*: agents interact with other agents (and possibly humans) by exchanging information formulated in a mutually agreed communication language. Moreover, the notion of social abilities comprises complex patterns of behavior based on communication protocols, e.g. for the purpose of negotiation.

The central problem of multi-agent systems is how to achieve coordinated action among agents in a way yielding problem solving capabilities that exceed those of any individual agent. Thus the coordination and communication structures are very important. Communication structures are divided into shared memory approaches and direct communication approaches. Most common are [6; 7]

- the contract net protocol: it is a high-level protocol for achieving efficient cooperation through task sharing in networks of communicating problem solvers,
- blackboard like systems: a common global data structure is used to exchange tasks and solutions,
- market mechanisms like auctions which are used to distribute goods to the agents, and
- other negotiation protocols especially designed for specific problem scenarios.

The concept of intelligent agents presented above is suitable for the domain of production planning and scheduling. First, an agent has some kind of knowledge of the problem to be solved (the scheduling problem) and its environment (e.g., other agents or the shop-floor), and is capable of negotiation. Second, it is able to quickly react to changes within its environment, e.g. a machine breakdown. And third, agents are

pro-active, allowing them, e.g., to improve their schedules while no other service request are issued [8]. Therefore, this definition is adopted for the agents to be developed here: Every agent shall be able to schedule its activities (autonomy), to change its schedule in case of disturbances (reactivity), and to optimize its schedule (pro-activeness). Messages concerning changes, disturbances etc. are exchanged using a communication language commonly agreed upon (social abilities). Thus several approaches for solving scheduling problems using multi-agent systems have been proposed [9]. Most of them are based on local scheduling scenarios where a set of agents is solving a scheduling problem for a single manufacturing site.

One of the main pitfalls of agent systems is the communication overhead if too many agents are involved. We will therefore limit the number of cooperating agents due to specific scheduling tasks and organize them into teams. Each agent of a team is provided with specific scheduling and domain knowledge in order to fulfill the tasks.

Frameworks

In order to create reusable software systems techniques have been developed to reuse software design principles as well as code segments. One of the main ideas is to provide software components that can be found and used easily to build new software systems. These components can be used in a black or white box manner. If we use black box components we only need to know what the interfaces of the component look like, if we use white box components we have the possibility of changing or extending these components. But if we do so, programming knowledge is necessary. Object oriented programming supports the building and reuse of software components. Because objects resp. classes are often too fine-grain and too specialized for reuse larger-grain abstractions, called frameworks, have been presented. A *framework* is a sub-system design made up of a collection of abstract and concrete classes and the interfaces between them [10]. Some details of the system are already implemented but some classes are left abstract and have to be implemented concretely for the specific application. Frameworks are often instantiations of several design patterns and successful realizations are found in GUI development and in enterprise application frameworks like enterprise java beans.

Outline of the project

In the ongoing project we develop an approach where cooperating agents solve the scheduling problems on the several layers. Therefore we have to

- model the organizational background of the supply chain together with the scheduling tasks of the business units involved,
- map the organizational structure to a multi-agent system,
- identify the structure, tasks and communication needs of the single agents,
- define the teams of cooperating agents,
- provide a framework for building agents.

The paper will give a short overview of the tasks and focus on the generic framework structure of the agents to be used within the scheduling teams.

2. Mapping the organizational model to agents

In a first step we have to model the organizational structure which reflects the business units involved together with the scheduling tasks they have to perform.

The organization model (see also [11; 12]) proposed is based on two main aspects, a hierarchy of (intra-company) scheduling units, and a network-like, inter-company dimension. Hierarchical structures are a common representation for intra-company directional systems for they are suitable for defining powers of decision, authorities to instruct, duties of supervision and tasks of inspection. Therefore, posts are defined according to resource-oriented aspects and arranged in a hierarchy. Potential posts are, e.g., an entire company, production sites, job shops, warehouses, transport vehicles, resource groups, or machines. A post is represented by a scheduling agent.

Complex organization structures, however, are not exclusively organized hierarchically. This applies especially to legally and economically independent enterprises, for relationships between them do neither define powers of decision nor authorities to instruct; they only represent the aspect of coordination between their directional systems. In addition to the hierarchical, static dimension of the intra-company directional system the network-like, dynamic dimension of the coordinating, logistical relationships on all levels of hierarchy is considered by a special type of relationship. Therefore organizations are accordingly represented by an overlay of hierarchical and network-like structures, thus achieving both vertical and horizontal integration. The resulting organization model is illustrated by figure 1. The nodes are representing (from a resource based point of view) business entities involved in the scheduling and supply chain management process. On the upper levels they represent companies, plants, working or resource groups. On the lower levels resource groups or even single machines

can be represented by an agent. The two kinds of relations (shown as dotted lines and arcs) define the hierarchical authority structures used within the organizations which are also used for defining responsibilities about schedules and placing orders, and second, the flow of materials which defines functional relations with suppliers and consumers of materials.

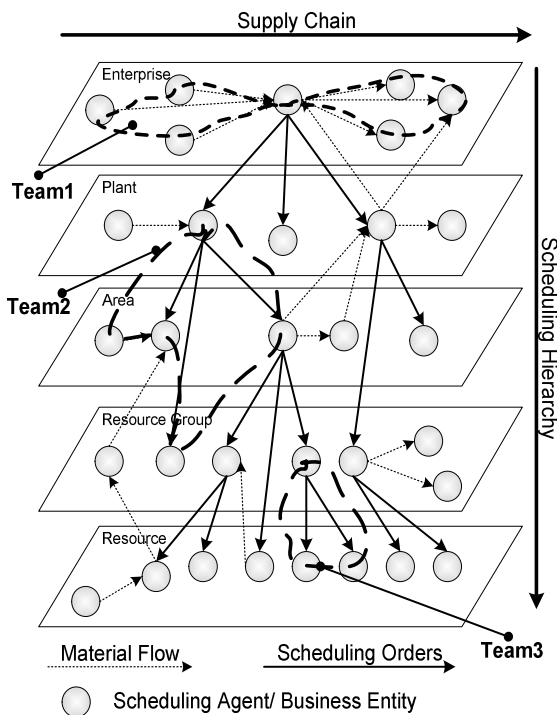


Figure 1: Supply Chain and Scheduling Hierarchy

Forming teams of agents

Teams of agents are the main feature of the model of the scheduling process that is used within our project. In order to prevent the communication overhead known from earlier multi-agent systems, teams of agents are formed that are responsible for solving a specific scheduling task. On the other side every agent is responsible itself for a specific schedule (the schedule of the resources it represents). Therefore each agent has to be provided with the scheduling knowledge to create or maintain the schedule without contacting the members of the team. Both kinds of relations are used for forming the teams of scheduling agents. Examples of teams are (see the loops in figure 1):

- the top level team of agents consisting of the top level nodes of the supply chain including the focal company),
- an intermediate team consisting of an agent representing a production site together with its subse-

quent agents representing the business units or production sites of the company,

- a local team consisting of the agent responsible for a transportation unit together with the agents representing the single vehicles.

The teams are acting as control loops and with agents in different control loops a hierarchical structure of control loops covering the whole supply chain model can be created. With these modeling possibilities the structure of the supply chain as well as the intra-organizational authority structures is reflected.

3. Scheduling tasks in SCM

If we look at the tasks of a supply chain we find different areas where scheduling problems have to be solved. Starting with the final products to be delivered there has to be a master schedule which says how much of these products should be available at what times. From that we can schedule when products have to be assembled, when parts have to be produced, stored and transported, and what raw materials have to be purchased. This is more or less a global scheduling task where a schedule for all the participating partners of the supply chain has to be generated. This global schedule can be separated into several schedules for the companies involved, e.g., a global production schedule, a global transportation schedule, etc. This is what APS (advanced planning and scheduling) systems are doing thereby using predictive scheduling approaches [3; 13]. But this is only a part of the solution of the supply chain scheduling problem because the global schedules have to be put into action within the companies which means that the companies too, have to create schedules for the tasks within their business units. And these business units have to create schedules for resource groups or for single resources. Thus we have a lot of interrelated scheduling problems on different organizational levels with schedules for each of the business units or resources. And what if the schedule on a lower level is disrupted and reactive scheduling has to be performed for the business unit involved. If the changed schedule will not affect the global schedule, this will be fine. But if the global schedule is affected too, which means that preceding or following activities cannot be performed as previously scheduled, then those have to be rescheduled too. This can lead to a rescheduling task even for the top level global schedules. This also means that schedule revision is a continuous process. So we are faced with predictive as well as reactive scheduling tasks on all levels. Additionally, we have to consider:

- interdependencies between production processes that are performed in different plants,

- generalized and imprecise data are used in global scheduling problems,
- the uncertainty about the actual "situation" in individual plants,
- that the scheduling problems are many-sided, there are production scheduling as well as transportation scheduling or storage scheduling problems,
- different goals on the different levels or within different business units,
- that schedules on higher levels should be more robust than those on lower levels.

From the scheduling point of view we have to look at the scheduling problems to be solved by the business units on the several levels of the supply chain hierarchy to reach the overall goal of maximum profit.

On the top level of our supply chain model the master schedule for the final products is created. This so called "global scheduling" task can be performed by a single agent (which would be similar to the APS-approach) or by a team of top level agents each representing a company of the supply chain. We will use a combination of both ideas, that is: we have a selected agent representing a focal company that is able to predictively create the master schedule as well as a team of agents that reinforce or change the global schedule if reactive scheduling is necessary. Global scheduling also means that requirements are generated for intermediate products to be manufactured, stored or transported. The global schedule is therefore divided into global schedules for the units involved.

The next step is then again a (intermediate) global scheduling, where schedules for the business units on the lower levels are generated, e.g. within the transportation or production facilities. On this level we have to distinct between production-, storage-, and transportation scheduling tasks because the tasks usually differ slightly and therefore the scheduling strategies have to be adapted too, although the overall structure of the problems can be modeled identically (see table 1).

Production scheduling is defined in the usual way, i.e., find a schedule that guarantees, that all intermediate or final products are manufactured within a given time window.

Transportation scheduling here means: find a schedule which guarantees that all intermediate products and materials are in time at the location where they are needed. Additionally, other goals of transportation scheduling should be regarded as well [14].

Storage scheduling means: find a schedule that guarantees that intermediate products or raw materials can be stored within a given time window.

On the upper levels the generation of a robust global schedule is very important. That means, that a

schedule should be generated that gives enough flexibility for a subsequent scheduler to react to disturbances without affecting the other sites. Additionally, it is important to detect capacity problems as early as possible and in case of reactive scheduling, to preserve as much as possible of the existing global schedules in order to minimize the subsequent effort on the subsequent level.

On the lowest level local scheduling is performed, which means, that the global schedules are transformed into concrete local schedules, e.g., production schedules which represent the assignment of operations to machines or transportation schedules defining the assignment of transportation activities to transport facilities.

On all levels predictive, reactive as well as interactive problems are addressed, not only to generate schedules but also to adapt them to the actual situation in the production and logistics process. The scheduling tasks can be characterized as follows:

- *predictive scheduling*: An initial or new distribution of orders to resources is generated.
- *reactive scheduling*: In case of disturbances, the reactive scheduling first tries to remedy them locally. If problems cannot be solved locally or the modified schedule influences other schedules (inter-plant dependencies), reactive scheduling on the next higher level or within the team of agents to which the unit belongs is initiated.
- *communication and coordination*: All levels have to be provided with data as actual and consistent as possible. Therefore information has to be sent between the levels, e.g. the global schedule consisting of information on internal orders, affiliated intermediate products, machine groups to use, time windows that should (possibly) be met, and required quantities of intermediate products, unexpected events that effect the local resp. the global level (e.g. the cancellation of an order or breakdowns of machine groups).

Global, local, transportation and storage scheduling problems can be modeled similarly using the seven-tuple (R, P, O, HC, SC, E, G) [15], where R denotes the set of required resources, P the set of producible products, O the set of actual orders, HC and SC stand for the sets of hard and soft constraints, respectively, E is the set of possible events, and G the set of goal functions.

Table 1 shows this model applied to global, local and transportation scheduling with examples for the items.

	Global Scheduling	Global intermediate Scheduling	Transportation Scheduling	Storage Scheduling	Local Scheduling
R	groups of resources, single plants	groups of machines or resources	transportation vehicles with capacity and other restrictions	storage facilities	machines
P	final products consisting of several intermediate products	intermediate products consisting of several intermediate products	transport of intermediate products using specific transportation vehicles	intermediate products, raw materials	intermediate products consisting of several production steps (operations)
O	external orders for final products	internal orders for intermediate products	internal orders for transportation of intermediates	internal orders for storage of intermediate products	internal orders for intermediates
HC	schedule all external orders, regard production requirements (one variant, precedence constraints, capacity)	schedule all orders, regard production requirements (one variant, precedence constraints, capacity)	schedule all orders, regard technical requirements (type of vehicle, transport capacity)	schedule all orders, regard storage requirements (place, type of facility, maximum duration)	schedule all orders, regard production requirements (one variant, precedence constraints)
SC	meet due date, minimize transportation times/ costs, reduce inventory costs.	meet due date, minimize transportation times/ costs, reduce inventory costs.	meet due dates, "optimal" vehicle utilization, minimize costs.	minimize inventory costs.	"optimal" machine utilization, meet due dates, minimize work-in-process costs.
E	global events, e.g. breakdown of machine groups	global events, e.g. breakdown of machine groups	transport events, e.g. traffic delay	global events, e.g. capacity problems	local events, e.g. machine breakdowns
G	global goals, e.g. meet delivery dates	global goals, e.g. just in time providing	transportation goals, e.g. minimize costs	local goals, e.g. minimal costs	local goals, e.g. optimal machine utilization

Table 1. Modeling global, local and transportation scheduling

The problem description is in part identical to the multi-site scheduling problem described in [16] where a distributed production has to be scheduled. We will therefore use the ideas of the solution presented there as well as solutions from single plant scheduling systems where a set of orders for products has to be scheduled on a set of machines.

For the predictive and reactive scheduling tasks several problem solving approaches are useful. Some of them have been checked for the MUST (Multi-Site Scheduling System) approach [16]. For single plant (or local) scheduling problems a huge variety of approaches can be found, e.g., in [17-20].

APS (advanced planning and scheduling) systems presented so far are basing on a centralized solution with a focal company that schedules the whole supply chain with predictive scheduling approaches (genetic algorithms and constraint solvers) [3; 13]. Additionally, like in the scheduling procedures used for multi-site scheduling today there is no immediate feedback from the local plants to the logistics department and communication between the local schedulers takes place without any computer-based support.

Teams of scheduling agents

Next we will answer the question how we can combine the resource-based partition of the busi-

nesses, the scheduling tasks and the model using teams of agents. According to the structural model presented before each agents represents - depending on the level on which it resides - a business entity, a group of resources or a single resource and is responsible for the schedule of the unit. The coordination of the agents of one common level is heterarchical, the coordination between the levels is hierarchically. This means the scheduling agent of an upper level together with dependent agents of the next lower level will form a team of agents which solves the scheduling problems of that level. In figure 1 three examples of possible teams are given:

- team 1: top level team scheduling the whole supply chain maintaining a global schedule.
- team 2: intermediate team scheduling business units or resource groups maintaining an (intermediate) global schedule.
- team 3: local team scheduling concrete resources e.g. machines, and maintaining a local schedule.

The team approach combines the flexibility of the cooperative work to resolve disruptions with efficient scheduling and single responsiveness for one schedule.

The agents therefore must have the possibility to

cooperate with agents on the same level as well as with agents on the level above resp. below. Depending on the level on which the agent resides it can deal with the specific scheduling problems of this level, i.e. every agent may have a different scheduling behavior, e.g., if the entity is a transportation unit, then the agent has to perform transportation scheduling tasks [14].

Agents to be employed in the production, transport and stock keeping domain differ regarding the knowledge as well the heuristics and strategies used for performing their planning, scheduling and coordination tasks. The development of software agents and a respective agent platform within our project will therefore pursue a component-based approach [21]. Specific types of agents can then be created by exchanging and configuring software components (cf. [22; 23]).

Thus we have to look for

- scheduling strategies that provide good schedules for the scheduling tasks on different levels of the system model,
- a communication mechanism which allows an efficient cooperation of the scheduling systems (agents).

And, additionally, our intention is to

- develop common models of the different scheduling problems in order to reuse scheduling strategies and system models,
- develop generic system models in order to create reusable components for the different scheduling systems (agents) involved in the system,
- evaluate scheduling strategies for their appropriateness in several scheduling scenarios.

4. The agent framework

With the approach presented so far the enterprises forming the supply chain as well as the business units and resources of the enterprises can be represented by a system of agents which have to be dynamically adapted to the position where they are needed. The basic idea for the implementation of such a dynamic system is the use of a framework with defines building blocks that can be adapted as necessary.

First we have to define an abstract agent resp. the agent framework which allows the building of the multi-agent system we need. A generic agent should meet the following requirements:

- The agent can be used on different scheduling/organizational levels.
- Each agent can be pursued with special scheduling strategies depending on the level it resides. This

combines predictive and reactive scheduling strategies.

- Each agent has an interface to the local data management system which can be a database as well as a file system.
- Each agent must have a communication interface to establish contact and exchange messages with the other agents of the multi-agent system.
- The language used for communication is KQML (knowledge query and manipulation language), one of the standard communication languages used in building agent based systems.
- Most of the negotiation is done via the contract net protocol. So this mechanism should be provided for every agent. Other forms of negotiation are possible as well.

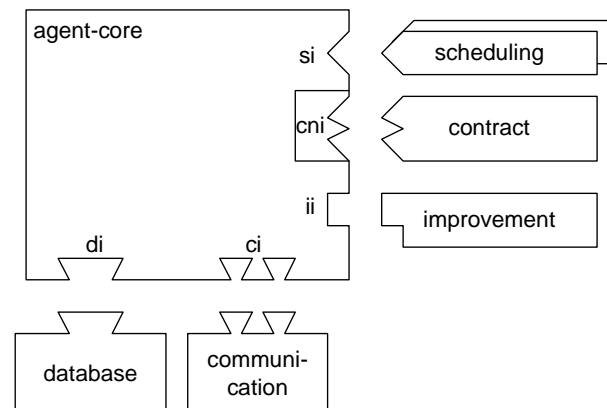


Figure 2: Framework for scheduling agent

On the basis of these requirements a generic agent framework has been developed. It is shown in figure 2 and consists of the following components:

- Agent core: it is responsible for the basic features like event handling, starting scheduling and improvement algorithms, performing database requests and integrating the components which are plugged to it.
- Database interface (di): it allows the connection to a relational database system which manages the local data.
- Communication interface (ci): allows basic communication (message passing) between agents and human schedulers involved.
- Scheduling component interface (si): allows the integration of different scheduling components implementing scheduling strategies to be used, e.g., predictive scheduling algorithms or reactive strategies useful in the case of specific events.
- Contract net interface (cni): basic features of the contract net protocol are provided so that the agent

can be involved in a contract net negotiation procedure.

- Schedule improvement interface (ii): here improvement strategies for the schedule can be incorporated. These are used by the agent if there are no events or messages to be treated.

The other components are defined in an abstract manner and have to be adapted as needed:

- Database: a relational database system, e.g. Oracle or MySQL.
- Communication: Messages are packed and unpacked, events are generated for agent core.
- Scheduling: all kinds of scheduling strategies.
- Contract: extensions of the contract net protocol needed for special negotiations.
- Improvement: all kinds of improvement strategies.

The basic rules of computation (the control loop of the agent core) are:

- if new events then interrupt improvement and event-handling
- if idle then schedule improvement
- event-handling means:
 - if database request then call database interface
 - if scheduling task then call appropriate scheduling algorithm and perform negotiation if necessary.

The system is implemented using object oriented programming with JAVA [24].

Examples

With the following two examples we will show how the framework is used. The first example shows the model of a global and local scheduling scenario e.g. of a production site with machine groups and machines. The second one shows the top level of a supply chain where a set of cooperating companies (suppliers, manufacturers, transport units etc.) has to be modeled.

Example 1. global and local level scheduling

Suppose we have a manufacturer with three different sites where a product P consisting of the intermediate products IP1 and IP2 is produced. The production is performed in four steps:

- manufacture IP1 (there are 2 alternative resources on different locations)
- manufacture IP2 (there are 2 alternative resources managed by one business unit in a separate location)
- manufacture IP3 using IP2 (one production site)
- assemble IP1 and IP3 to get the final product P (one assembly site).

This also defines that IP1 and IP3 have to be produced before IP2, and IP2 has to be produced before IP3.

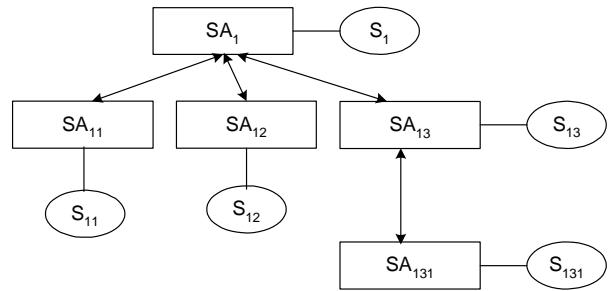


Figure 3: Agents for example 1

According to our resource based modeling approach the sites and resources are represented by a team of agents. Figure 3 shows the agents with the schedules they are responsible for and their connections for cooperation:

- Agent S_1 :
 - represents the assembly resource and the focal site delivering the final product,
 - responsible for the global schedule (overall schedule of production),
 - scheduling strategy: latest possible start,
 - cooperation: manager in contract net and direct negotiation with agent S_{13} .
- Agent S_{11} :
 - represents the first of alternative production sites for intermediate product IP1,
 - responsible for the local schedule of production site,
 - scheduling strategy: earliest possible end,
 - cooperation: contractor in contract net.
- Agent S_{12} :
 - represents the second of alternative production sites for intermediate product IP1,
 - responsible for the local schedule of production site,
 - scheduling strategy: latest possible start,
 - cooperation: contractor in contract net.
- Agent S_{13} :
 - represents a production facility where intermediate product IP2 is produced,
 - responsible for the local schedule of production site,
 - scheduling strategy: latest possible start,
 - cooperation: direct negotiation with agent S_1 .
- Agent S_{131} :
 - represents two machines for manufacturing product IP2,
 - responsible for the local schedule of the two machines,
 - scheduling strategy: earliest possible end,

- cooperation: direct negotiation with agent S_{13} .

Together these agents form a team responsible for the scheduling of the product P. The scheduling process then is as follows:

- Agent S_1 creates the master schedule and initiates a negotiation using the contract net.
- The agents S_{11} and S_{12} receive the request from S_1 and check if they can fulfill the task. If yes, they give a bid to S_1 .
- Agent S_1 decides which bid to accept and schedules the task it has selected.

- There is a direct negotiation with agent S_{13} because it is the only responsible agent.
- Agent S_{13} directly communicates with agent S_{131} , so they form a little team of their own.

The reactive scheduling then looks like: if an activity cannot be performed as previously scheduled a rescheduling procedure is started by S_1 . Again a request is sent to the agents S_{11} and S_{12} and the best bid is chosen. The agent S_{13} is contacted directly with a proposal for rescheduling.

The complete example is described in [24].

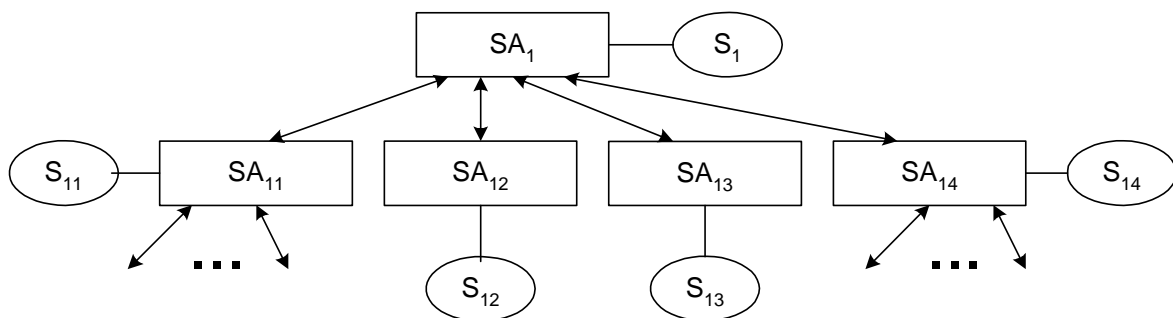


Figure 4: Agents for example 2

Example 2. Scheduling the supply chain

The supply chain example is modeled by a set of agents, with one agent representing the focal company and agents representing the enterprises involved (supplier, transport, storage, production, retailer). These together form a team representing the top level of the supply chain and performing the master scheduling tasks. Thus the agents have to solve different scheduling tasks, e.g. production scheduling, transportation scheduling. The agents representing the companies are themselves top level agents of teams responsible for the scheduling within the companies. So we get several control loops created by teams of agents which perform the scheduling tasks on the different levels. Figure 4 shows an example how these top level agents could form the (top level) supply chain scheduling team.

- Agent S_1 :
 - represents the focal site of the supply chain,
 - responsible for the global schedule (overall schedule of production),
 - scheduling strategy: own heuristic,
 - cooperation: manager in contract net.
- Agent S_{11} :
 - represents the suppliers,
 - responsible for the global schedule of supply,
 - scheduling strategy: just in time,

- cooperation: contractor in contract net with focal company, manager in contract net with all suppliers.
- Agent S_{12} :
 - represents the transport units,
 - responsible for the global schedule of transport from suppliers to manufacturers,
 - scheduling strategy: own heuristic,
 - cooperation: contractor in contract net.
- Agent S_{13} :
 - represents the production facilities,
 - responsible for the global schedule of production,
 - scheduling strategy: earliest due date,
 - cooperation: contractor in contract net.
- Agent S_{14} :
 - represents the second group of transport facilities,
 - responsible for the global schedule of transport from production sites to retailers,
 - scheduling strategy: just in time,
 - cooperation: contractor in contract net with focal company, manager in contract net with all suppliers.

Agents S_{11} and S_{14} give an impression how the next control loop could be added to the top level team.

A concrete implementation using the Euro-coin production and distribution as example scenario is under development.

4. Conclusion and further work

An approach has been presented using a generic agent framework to build teams of agents which are able to support the scheduling tasks within the supply chain. The model combines an hierarchical intra-company view of the scheduling process (as it is used in multi-site scheduling) with heterarchical inter-company relationships and heterarchical negotiations on different levels of the organizational hierarchy. The generic agent framework allows the consistent design of agents residing on several levels of the organization and performing different scheduling tasks together grouped in so called teams of agents. The first prototypical implementation shows the feasibility of the approach using examples from the global and local scheduling level. Further examples, e.g. from the top level of the supply chain down to the local facilities have to be provided to show the efficiency of the approach. Thereby the supply chain is divided in hierarchically ordered control loops built by teams of agents. Once the scheduling tasks have been modeled and performed, the other tasks of the supply chain management should be supported as well, i.e. the additional features of supply chain management have to be investigated too.

References

- [1] Scheer, A.-W., and Borowsky, R. 'Supply Chain Management: Die Antwort auf neue Logistikanforderungen'. *LM'99 - Intelligente I+K Technologien*, Bremen. Springer Verlag. 1999, pp. 3-14.
- [2] Ganeshan, R., and Harrison, T. P. *An Introduction to Supply Chain Management*. 1995.
- [3] Stadtler, H., and Kilger, C. *Supply Chain Management and Advanced Planning*. Springer. 2000.
- [4] Franklin, S., and Graesser, A. 'Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents'. *Third International Workshop on Agent Theories, Architectures, and Languages*. Springer-Verlag. 1996.
- [5] Wooldridge, M. 'Intelligent Agents'. In: G. Weiss, ed. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press. 1999, pp. 27-77.
- [6] Huhns, M. N., and Stephens, L. M. 'Multiagent Systems and Societies of Agents'. In: G. Weiss, ed. *Multiagent Systems: a modern approach to distributed artificial intelligence*. MIT Press. 1999, pp. 79-119.
- [7] Wooldridge, M. *An Introduction to MultiAgent Systems*. John Wiley & Sons. 2002.
- [8] Henseler, H. 'From Reactive to Active Scheduling by using Multi-Agents'. In: R. M. Kerr and E. Szelke, eds., *IFIP TC5/WG.5.7 Second International Workshop on Knowledge Based Reactive Scheduling (KBRS) PREPRINTS*. Budapest. 1994.
- [9] Parunak, H. v. D. 'Industrial and Practical Applications of DAI'. In: G. Weiss, ed. *Multiagent Systems: a modern approach to distributed artificial intelligence*. MIT Press. 1999, pp. 377-421.
- [10] Sommerville, I. *Software Engineering*. Addison-Wesley. 2001.
- [11] Appelrath, H.-J., Sauer, J., Freese, T., and Teschke, T. 'Strukturelle Abbildung von Produktionsnetzwerken auf Multiagentensysteme'. *Künstliche Intelligenz*, 3/00. 2000, pp. 64-70.
- [12] Sauer, J., Freese, T., and Teschke, T. 'Towards Agent-Based Multi-Site Scheduling'. In: J. Sauer and J. Köhler, eds., *Proceedings of the ECAI 2000 Workshop on New Results in Planning, Scheduling, and Design*. Berlin. 2000, pp. 123-130.
- [13] Corsten, H., and Gössinger, R. 'Advanced Planning Systems - Anspruch und Wirklichkeit -'. *PPS Management*, 6(2). 2001, pp. 32-39.
- [14] Sauer, J., and Appelrath, H.-J. 'Integrating Transportation in a Multi-Site Scheduling Environment'. In: H. Sprague, ed. *HICSS-33, Hawaii International Conference on System Sciences*. IEEE, Maui. 2000, pp. on CD.
- [15] Sauer, J. 'Knowledge-Based Systems in Scheduling'. In: T. L. Leondes, ed. *Knowledge-Based Systems Techniques and Applications*. Academic Press, San Diego. 2000, pp. 1293-1325.
- [16] Sauer, J. 'A Multi-Site Scheduling System'. *AAAI's Special Interest Group in Manufacturing Workshop on Artificial Intelligence and Manufacturing: State of the Art and State of Practice*. 1998, pp. 161-168.
- [17] Dorn, J., and Froeschl, K. A. *Scheduling of Production Processes*. Ellis Horwood. 1993.
- [18] Sauer, J., and Bruns, R. 'Knowledge-Based Scheduling Systems in Industry and Medicine'. *IEEE-Expert*(February). 1997, pp. 24-31.
- [19] Smith, S. F. 'Knowledge-based production management: approaches, results and prospects'. *Production Planning & Control*, 3(4). 1992.
- [20] Zweben, M., and Fox, M. S. *Intelligent Scheduling*. Morgan Kaufman. 1994.
- [21] Szyperki, C. *Component Software - Beyond Object-Oriented Programming*. Addison-Wesley Verlag. 1998.
- [22] Swaminathan, J. M., Smith, S. F., and Sadeh, N. M. 'Modeling Supply Chain Dynamics: A Multiagent Approach'. *Decision Sciences Journal*, 29(3). 1998, pp. 607-632.
- [23] Sauer, J. 'Knowledge-Based Design of Scheduling Systems'. *Intelligent Automation and Soft Computing*, 7(1). 2001, pp. 55-62.
- [24] Bröske, S. *Ein Framework für Planungsagenten*. Diploma Thesis, Universität Oldenburg, FB Informatik. 2001.