Available online at www.sciencedirect.com
SCIENCE DIRECT®
PERGAMON
Engineering Applications of Artificial Intelligence 16 (2003) 335–348
Engineering Applications of ARTIFICIAL INTELLIGENCE
www.elsevier.com/locate/engappai

Agent-based optimisation of logistics and production planning

Anthony Karageorgos[a,*], Nikolay Mehandjiev[a], Georg Weichhart[b], Alexander Hämmerle[b]

[a] *Department of Computation, UMIST PO Box 88, Manchester M60 1QD, UK*
[b] *PROFACTOR Produktionsforschungs GmbH, Wehrgrabengasse 1-5, A-4400 Steyr, Austria*

Abstract

Manufacturing and logistics service provision enterprises are currently moving towards open virtual enterprise collaboration networks to meet the needs of the Global Economy. In such networks, manufacturing and logistics planning and scheduling is challenging due to the difficulties in integrating information from different partners and in exploring a large and dynamically changing number of planning and scheduling alternatives. Agent-based technology is considered suitable to support planning and scheduling in such enterprises because agents can dynamically adapt their behaviour to changing requirements and they can reduce the number of planning and scheduling alternatives via negotiation.

This paper presents an agent-based approach for supporting logistics and production planning, taking into account not only production schedules but also availability and cost of logistic service providers. This is achieved through efficient negotiation mechanisms based on an extended contracting protocol. The agent infrastructure is being developed within the context of Agentcities, a successful EU-funded initiative to build a world-wide distributed and open platform which provides agent-based services.

The proposed approach is illustrated in a case study concerning optimisation of production planning of a virtual manufacturing enterprise in relation to sub-contracted logistic services used to transport materials between the enterprise units.

Keywords: Agents; Agent-based systems; Enterprise integration; Manufacturing systems; Interaction protocols; Holonic systems

1. Motivation

Manufacturing and logistic service provision enterprises are trying to organise and optimise the efficiency of their cooperation, using software that supports negotiations at different levels of automation. E-marketplaces are examples of such software. In contrast to these centrally organised software solutions, systems using the agent paradigm are built to reflect the distributed and autonomous nature of virtual enterprises and thus provide a natural way to design and implement such environments. When, for example, a non-trivial reasoning and negotiation has to be performed in order to optimise the execution of the system, agent capabilities can be used to illustrate advantages as compared to a solution based on centralised optimisation techniques and conventional inter-enterprise business process infrastructures.

Calculating the optimum of a scheduling problem is a non-trivial task. For example, scheduling of product-orders on a number of machines is a non-deterministic polynomial (NP) hard problem (Bongaerts, 1998; Jain and Meeran, 1998). It is not possible for such problems to calculate the best solution in a straightforward manner. Therefore all possible solutions have to be calculated, to be able to then choose the best solution. But with this set of problems, the number of solutions literally explodes when increasing the values of input variables (numbers of machines and product-orders in this case). The following "mathematically" simple example shall illustrate the statements above (Jain and Meeran, 1998).

Given a finite set O of n (product-)orders $\{O_i\}_{i=1}^{n}$ to be processed on a finite set $\grave{I}$ of m machines $\{M_k\}_{k=1}^{m}$ and assuming that each order $O_i \in O$ has one operation on each machine m, a machine can work on one order at a time (machine capacity = 1). We further assume that one order can be processed on one machine at a time, and a

*Corresponding author. Tel.: +44-161-200-3306; fax: +44-161-200-3324.

E-mail addresses: karageorgos@acm.org (A. Karageorgos), mehandjiev@acm.org (N. Mehandjiev), georg.weichhart@profactor.at (G. Weichhart), alexander.haemmerle@profactor.at (A. Hämmerle).

0952-1976/$ - see front matter
doi:10.1016/S0952-1976(03)00076-9

started operation cannot be interrupted. Given this set of assumptions there are $(n!)^m$ possible solutions for the allocation of orders to machines (Jain and Meeran, 1998). Fig. 1 illustrates this combinatorial explosion by displaying the logarithm of the number of possible (but not necessarily feasible) solutions ($\log_{10}((n!)^m)$) for different numbers of orders and machines. This indicates that even for small problems the number of solutions is greater than 10^{100} (the enormous size can be realised better if compared to 8.64×10^{10} ms = 24 h).

Given the combinatorial explosion of "brute force" optimal scheduling, a number of approaches have been developed to prune down the search tree. We can divide those into centralised scheduling approaches, such as SYSPRO (2020 Software 2003), and decentralised agent-based solutions as proposed in this paper. Centralised approaches have many advantages but suffer from a number of serious weaknesses which make them unsuitable for virtual enterprises. One of these weaknesses is the "closed system" assumptions inherent in the software, which does not allow for dynamic change of enterprise configuration and provide poor interoperability mechanisms. The second weakness is the centralised mode of planning and scheduling, which does not favour localised re-planning and use of negotiation to prune down the search tree.

In this paper, an alternative agent-based approach to planning and scheduling of manufacturing and logistics services in virtual enterprises is proposed. Agents can use negotiation to reduce the search tree and agree on viable optimised schedules. Within the holonic modelling paradigm, local optimal solutions can be found at lower levels of the holonic organisation, and combined at the higher level into aggregated solutions using negotiation.

The holonic paradigm is an emerging approach for modelling of manufacturing enterprises (Vancza and Markus, 1998; Leitão and Restivo, 2002). The idea of holonic business is based on the collaboration of autonomous and cooperative business units (*holons*). A holon is composed of sub-holons acting as a whole. A (dynamic) business network can be seen as a temporarily existing holon where the sub-holons are the necessary service providers (business partners) within the network. The holonic relationships can be extended to many levels inside an enterprise, creating organisational structures known as *holarchies* (Koestler, 1976, p. 12). In contrast to hierarchies, the decision power in holarchies lies on the lower organisational levels. The holonic view captures the dynamism of virtual enterprises since holons can be created dynamically according to the requirements as is the case in open business environments (Ulieru et al., 2001). Due to their flexibility and adaptivity, software agents are particularly suitable to support holonic business organisations (Ulieru et al., 2001; Leitão and Restivo, 2002). The terms holonic agent and holonic agent system are used to refer to agents and agent systems organised according to the holonic metaphor.

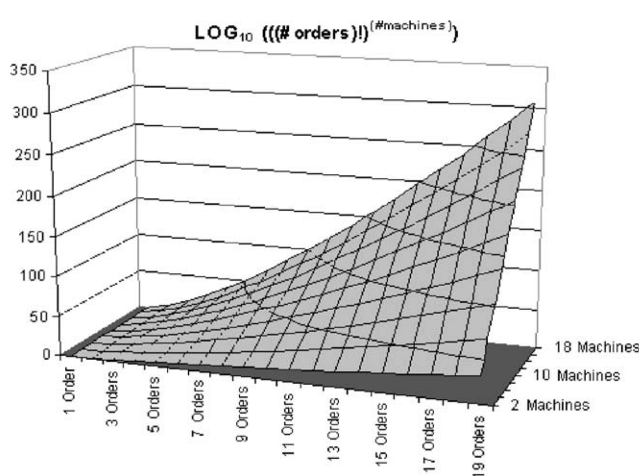


Fig. 1. Solution space of a simple scheduling problem.

Providing an agent-based infrastructure where holonic agents will be able to interoperate is currently an open issue (Gou et al., 1998). Major obstacles towards this goal are the lack of standardised technologies that would enable (possibly heterogeneous) agents corresponding to different enterprises advertise their services, locate each other and interact in a dynamic manner using commonly shared domain knowledge.

A successful agent-based systems interoperation initiative is *Agentcities* (Willmott et al., 2001). The Agentcities initiative works towards the vision of an ambient proactive environment where heterogeneous, autonomous and increasingly intelligent systems, which represent businesses, services and individuals, are able to interact with each other and enable flexible and dynamic composition of services. In the first year of its operation, the Agentcities initiative has resulted in an experimental open infrastructure providing "white" and "yellow pages" information concerning available agent platform locations and agent service types.

The project described in this paper is based on the holonic agent framework as a modelling and design paradigm. The holonic design is implemented using open agent technologies compatible with the Agentcities platform. The main goal is to build such a system which can deliver co-optimisation of production and logistics planning within a virtual enterprise. The approach proposed in this paper is based on dynamic interaction of holonic business agents, including an appropriate ontology and an extension of the standard *Contract Net Protocol (CNP)*.

The objectives of the project are to develop an agent-based planning and scheduling approach, a demonstrator to show the feasibility of this approach, and to evaluate the proposed approach using a number of scenarios drawn from a case study. The case study used is discussed in the third section of this paper. It concerns a simple virtual enterprise including a number of manufacturing units, which request external logistics services. The virtual enterprise receives orders from a wholesale distributor. The planning of the production batches depends on the orders, the abilities and workload of manufacturing units and on the availability and pricing of logistic services for moving the products between the two units and to the wholesale distributor. Logistic service providers are in turn selected based on the transport requirements arising from the production plan.

The rest of this paper is organised as follows: Our approach is described in Section 2, including the extended contracting protocol used for production and logistics planning optimisation. An intelligent production and logistics planning scenario is described in Section 3. In Section 4 the prototype is discussed based on a walk-through scenario and a model of the ontology used. The lessons learned from building the prototype are summarised in Section 5 and Section 6 discusses the advantages of the proposed approach in comparison with relevant work. Finally, Section 7 outlines the scope for further work and Section 8 concludes the paper.

2. Our approach

Agents are considered an important paradigm for developing industrial distributed systems (Fox et al., 2000). A number of researchers have attempted to apply agent technology to manufacturing enterprise integration, supply chain management, manufacturing scheduling and control, for example (Brückert et al., 1998; Bussmann, 1998; Camarinha-Matos and Afsarmanesh, 2001). In such approaches negotiation is used to reduce the number of planning and scheduling alternatives via negotiation-based contracts. Agent technology has also been applied to material handling, logistics service provision and holonic manufacturing systems, for example (Fischer et al., 1996; Goletz and Ferreira, 2000; Gou et al., 1998; Ulieru et al., 2001; Leitão and Restivo, 2002). In such approaches, the holonic paradigm was applied to address the openness and dynamism of virtual enterprises. Our approach combines both agent technology and the holonic paradigm to address planning and scheduling in virtual manufacturing enterprises. In this section we motivate our approach to integrated logistics and production planning across companies, and outline the resulting system architecture.

2.1. A holonic view of business networks

In our approach we consider a holonic view of business networks. A (dynamic) business network can be seen as a temporarily existing holon, where the subholons are the necessary service providers (business partners) within the network. A holonic agent (holon) is an agent which is composed of sub-agents acting in a corporate way (Bürckert et al., 1998). The holonic picture can be extended to lower levels inside an enterprise, giving rise to the notion of an ‘‘enterprise holarchy’’, for example a recursively structured agent society naturally reflecting the nested service structure, (see Fig. 2). Services are decomposed into sub-services at each holonic level. The holonic architecture reflects the mapping between (sub)-services and (sub)-holons.

The network level in Fig. 2 focuses on bringing independent organisations together and on forming dynamic virtual enterprises. At the *organisation level*, the units that form that holon, are interconnected in a rather static way, for example their links do not change frequently, forming what we consider as *static virtual enterprise*.[1] These units could be departments of the same or different business organisations. At the level of functional units (*functional level*) the focus is on completing a concrete task or function. Finally, the lowest level is representing the machines or humans that do the actual work (*operative level*) within a department of a business organisation.

The approach described in this paper focuses on the organisation layer integrating the functional and operative levels. This approach can be extended to allow the dynamic formation of temporary virtual enterprises. Due to constraints at the current stage of our project, however, we had to consider virtual enterprise formation as a totally separate activity preceding the planning and scheduling stages. Mixing these activities into a three-way optimisation is a subject of our ongoing research work.

2.2. Implementing a holonic architecture

Resource agents represent the active system components in the Virtual Enterprise scenario. The top-level resource agent is the Virtual Enterprise itself. It is responsible for managing incoming customer requests, decomposing manufacturing and logistics tasks based on product specific process plans, allocating tasks to lower level resources (manufacturing facilities and logistics providers) and placing offers to customers.

Manufacturing facilities and (optionally) logistics providers are resources inside a Virtual Enterprise. They are responsible for further decomposing tasks and for local planning and scheduling of manufacturing and

[1] In the context of this paper a static virtual enterprise is formed to fulfil strategic goals of the participating partners and its structure does not change frequently. The tactical and operational links between partners, however, are continuously updated using market mechanisms according to dynamic business requirements.

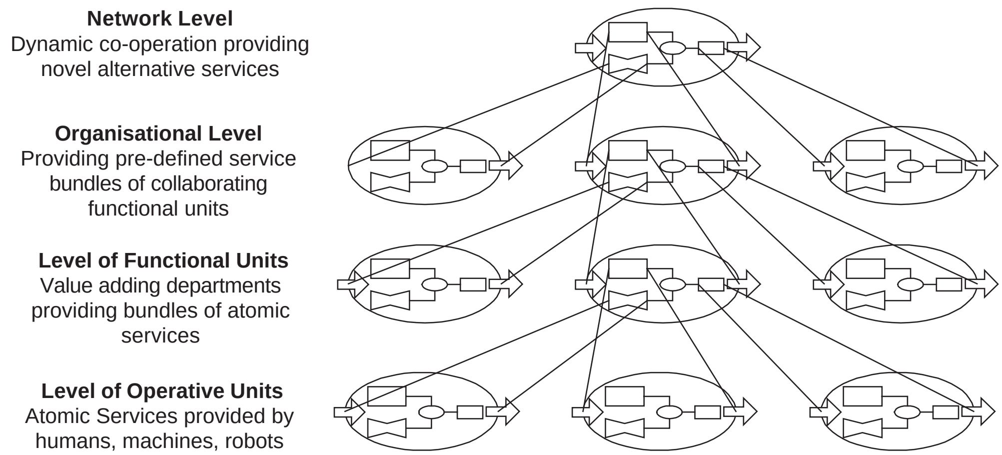


Fig. 2. Enterprise holarchy.

logistics tasks by allocating tasks to machines (or trucks).

Machines are resources inside manufacturing facilities actually processing the work-pieces. They manage their internal production plans and interact with the manufacturing facility to generate global production plans.

The resource agents build up a recursively structured agent society, implementing the holonic framework presented above. The described structure is based on the case study described in Section 3 and illustrated in Fig. 5. In other scenarios, additional resource level types may be involved, for example (production) departments and work cells.

2.3. Negotiation-based planning and scheduling

This project is a part of our general effort to provide a software infrastructure that would enable integration of activities of virtual manufacturing enterprises such as design, planning, scheduling, simulation and execution, with those of its suppliers, customers and partners into a distributed intelligent open environment. The main innovation of our approach is an agent-based infrastructure including negotiation mechanisms for optimising manufacturing and logistics planning and scheduling based on extended contracting protocols.

2.3.1. Nested Contract Net protocol

The Contract Net is a negotiation protocol proposed by Smith, (1980), which facilitates distributing subtasks among various agents. The agent wanting to solve the problem broadcasts a call for bids, waits for a reply for some length of time, and then awards a contract to the best offer(s) according to its selection criteria. This protocol has been widely used for multi-agent negotiation. Some researchers have proposed modified versions of Contract Net protocol for special applications, such as the Extended Contract Net Protocol (ECNP) proposed by Fischer et al. (1995). The recursive nature of the holonic architecture requires us to define a recursive or "Nested" Contract Net Protocol, illustrated in Figs. 3 and 4.

A Query from the customer, which is outside our system, arrives at the Wholesaler. The Wholesaler uses the Directory Facilitator provided by the FIPA compatible platform to find all Virtual Enterprises able to provide the appropriate services. Subsequently, he sends them a "Call For Proposals" (CFP).

Each Virtual Enterprise splits the request into basic products. This separation of the task into subtasks reduces computational complexity. The Manufacturing Facilities are allowed to strive for an optimal solution of the subtask. The Virtual Enterprise later on just combines the proposed solutions of subtasks. This limits the number of possibilities that have to be evaluated, but the draw back is that a globally optimum solution can not be guaranteed. The Virtual Enterprise also considers time and precedence.

Manufacturing Facilities, which are part of the Virtual Enterprise, receive one or more CFPs for producing basic products. Here the CFP is forwarded to the Machines, where the local machine-usage plan is checked and a "Propose" message is returned. The information returned with the message includes the earliest start time, the estimated finish time, and the costs.

The Manufacturing Facility now chooses the proposals to combine them in a way to have an optimum solution of the subtask requested by the Virtual Enterprise, and proposes it to the Virtual Enterprise. The proposals that are not amongst the chosen ones are rejected right after a solution is chosen. This is a slight change of the original Contract Net Protocol, which is rather a technical optimisation in terms of freeing system resources, than a semantic modification.

2.3.2. Using the Nested CNP for planning optimisation

The VE in turn calls the logistic service providers for proposals on the transport between the facilities and to

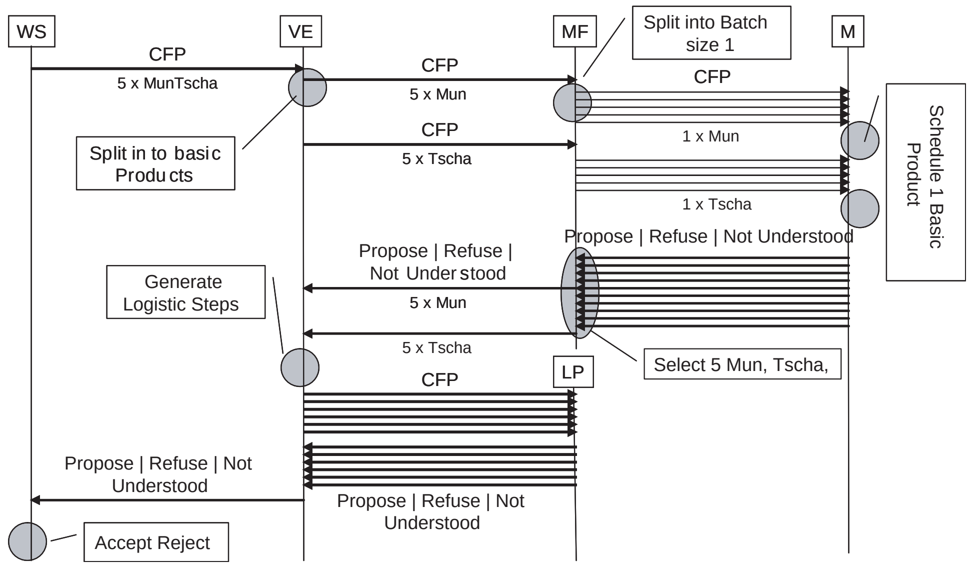


Fig. 3. Par 1 of the Nested Contract Net Protocol.

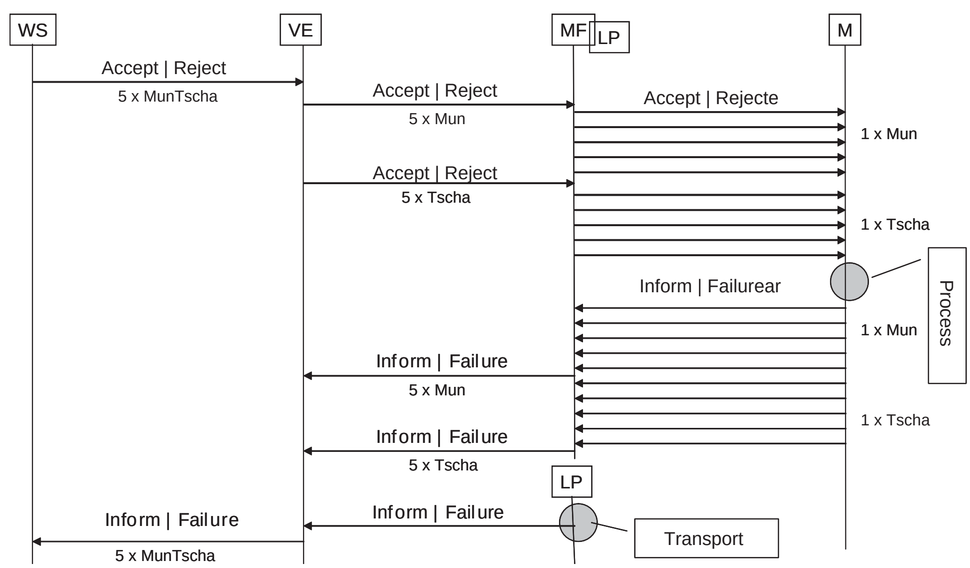


Fig. 4. Part 2 of the Nested Contract Net Protocol.

the wholesaler. If one facility is able to produce the whole desired product, then of course only transport from the facility to the wholesaler is necessary. When the Virtual Enterprises receive the proposals from the logistic service providers, it combines the subtasks to work out the optimal one in terms of combination of planning and logistics according to the following criteria: cost, timing, terms of agreement (which would eventually affect the cost, etc.). This is proposed to the wholesaler, which accepts or rejects the proposal, perhaps after doing a similar comparison between several proposals from different Virtual Enterprises.

The decision propagates down to the machines. If it was an "Accept" message, the machines, after having finished, inform the Manufacturing Facilities which after having received all messages, in turn inform the Virtual Enterprise, which again wait for all sub-holons to finish and then informs the wholesaler upon completion. In case of an error, a "Failure" message is propagated.

3. The case study scenario

In our case study we regard several static Virtual Enterprises (VE) consisting of two Manufacturing Facilities (MF)—see Fig. 5. The geographical distribution of the VE is reflected in the Agentcities demonstrator, where one MF will run on an Agentcities node in Manchester, the other MF will run in Steyr. Each MF

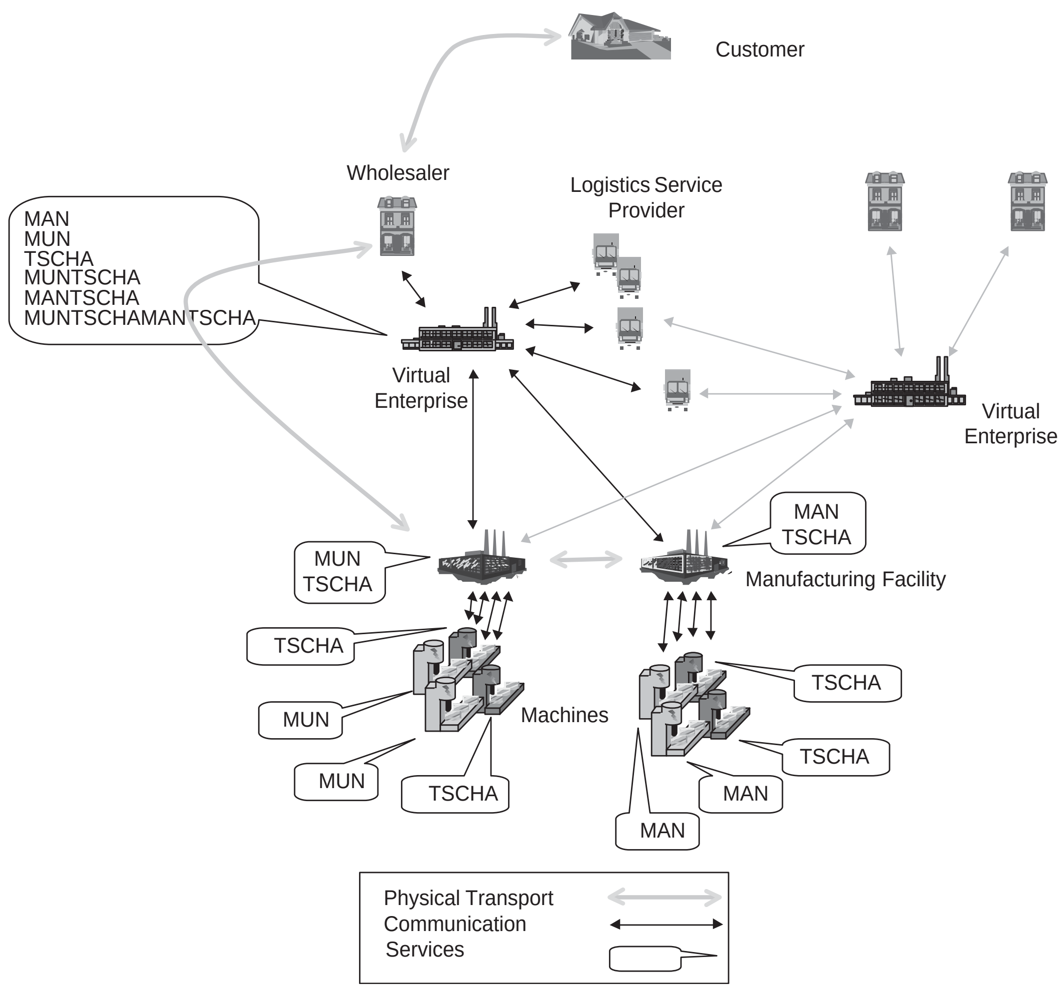


Fig. 5. Case study system architecture.

is part of several VEs. The services provided by the MF are the following:

1. MF1 is able to manufacture the basic products MUN and TSCHA,
2. MF2 is able to manufacture the basic products MAN and TSCHA.

Basic products may be combined to form more complex products. A wholesaler requesting products from the VE therefore has the choice between

MUN
MAN
TSCHA
MUNTSCHA
MANTSCHA
MUNTSCHA-MANTSCHA.

There are constraints when producing MUNTSCHA and MANTSCHA which require MUN (MAN) to be produced first, and TSCHA be then applied. In case of MUNTSCHA-MANTSCHA both subparts can be produced in parallel, and get assembled at either facility.

Let us look at the case where an enquiry for MUNTSCHA has come in. The Virtual Enterprise will contact the two manufacturing facilities MF1 and MF2 and ask them to prepare quotations regarding the scheduling of the production batch and the price they demand. The VE will initially receive two quotations from MF1: for MUN and TSCHA; whilst MF2 will only quote for TSCHA. It will be the job of the Virtual Enterprise to combine these offers with the data regarding availability and pricing of logistics services, and to create its own offer to be sent to the wholesaler. For example, if MF2 is much closer to the wholesaler and TSCHA adds considerable transportation costs on the product, it may be cheaper to only produce MUN in MF1, then ship the part-finished goods to MF2 where TSCHA can be added before dispatch to the wholesaler.

4. Prototype implementation

To demonstrate the feasibility of implementing an agent-based support system for combined optimisation of production and logistics planning, a prototype system was developed using the Java Agent Development Framework (JADE) (JADE, 2003). A generic library has been initiated that enhances JADE with services and components for building holonic supply chain management systems. The holonic architecture allows us to start with the enterprise level, dealing with a fixed set of parties forming a static Virtual Enterprise. We will develop it so that later on it could be extended both outwards to the dynamic Network level, where holons are dynamically formed, and inwards to the level of functional units.

The selection of JADE was done primarily because is offers a considerable amount of ready-to-use infrastructure components. It also provides a mechanism where the application programmer is able to concentrate on creating customized interaction protocols and ontologies.

4.1. A walk-through scenario using the prototype

When starting up the system, a number of agents are generated as shown in Fig. 6. There is one wholesale agent (WS1), one Virtual Enterprise agent (VE1), two Manufacturing Facility agents (MF1 and MF2), two Logistics Provider agents (LP1 and LP2), and five Machine agents (M1 to M5). There are also two graphical user interface agents (GUIAgent11..56 and GUIAgent11..62).

The scenario starts by the user requesting 5 pieces of MUNTSCHA from the wholesaler agent WS1 as shown on Fig. 7.

Fig. 8 demonstrates the planning within Manufacturing Facility 1 in response to this request.

The user is then asked if the offer proposed on Fig. 9 should be accepted.

Accepting the proposal would result in executing the expected production plan, demonstrated in Fig. 10.

Fig. 7. Requesting 5 pieces of MunTscha.

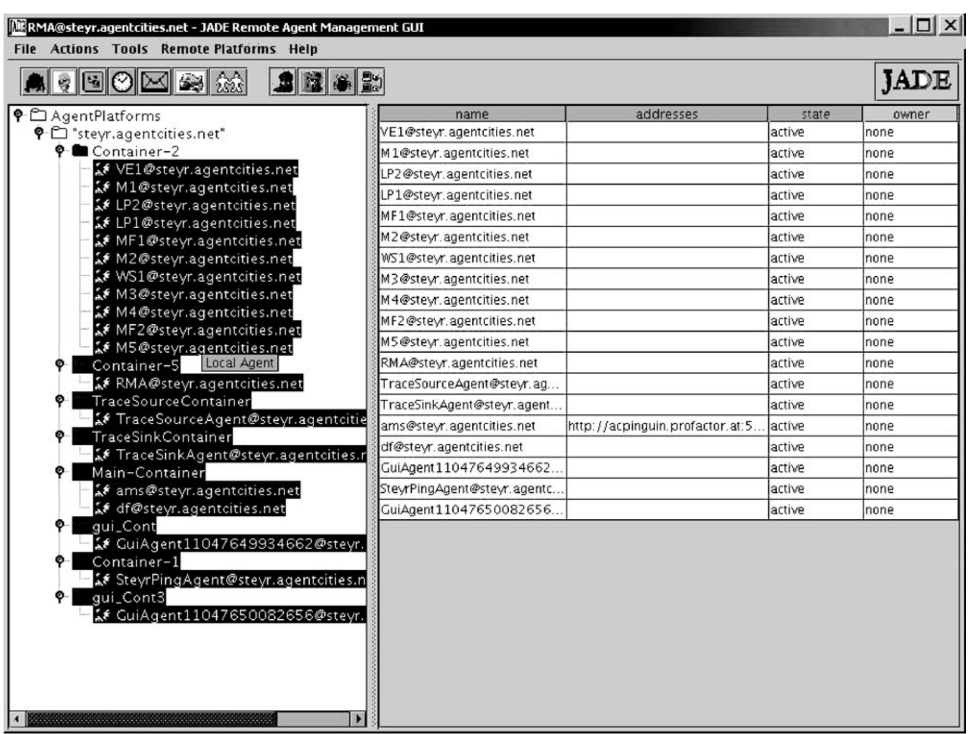


Fig. 6. Start-up agents in our scenario.

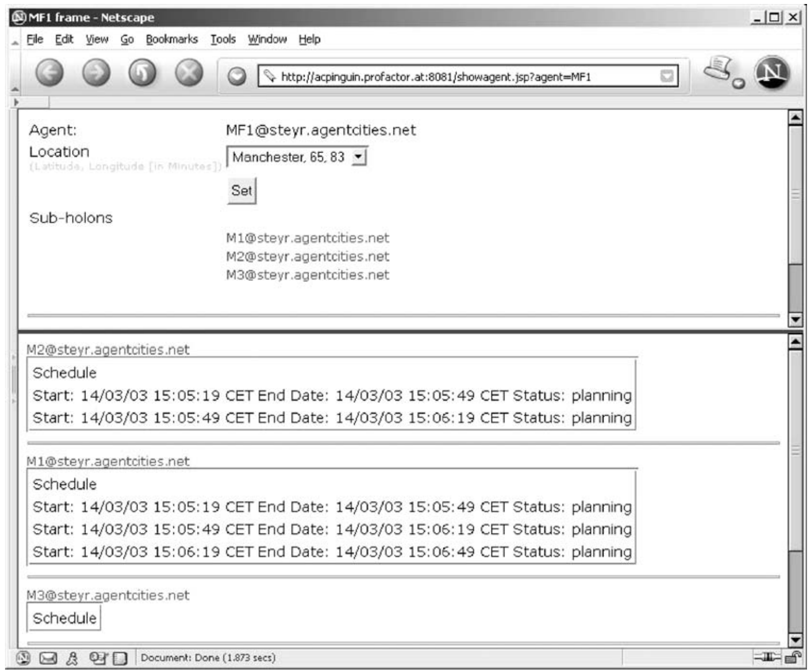
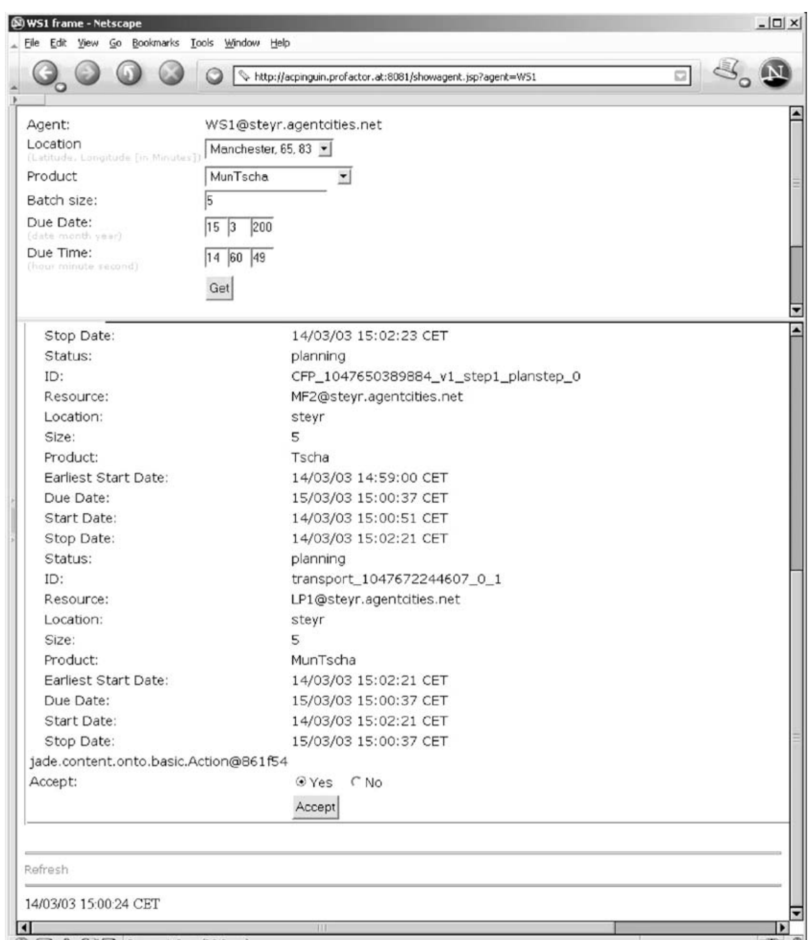


Fig. 8. Planning within one of the manufacturing facilities.



Fig, 9. Accepting a proposal.

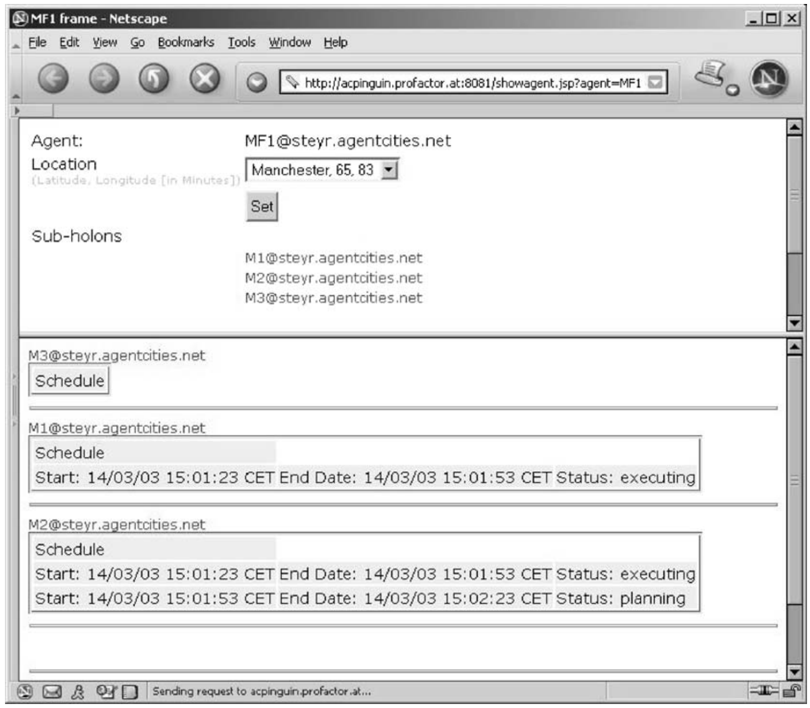


Fig. 10. Executing the manufacturing plan in MF 1.

This could be followed either by an "Inform" message that everything as been produced or by a "Failure" message notifying the receiver that the VE is not able to produce (refuse). The former message is shown in Fig. 11.

4.2. Ontology

For the agents to be able to share knowledge we developed an ontology modelling the essential concepts of interest in the INTLOG domain, which is shown on Fig. 12. The ontology classes are generated using the Protégé Ontology Tool. The super classes *Concept*, *AgentAction* and *Predicate* are provided by the bean-generator plug-in for Protégé, which allows creating an ontology implementation from the Protégé model ready to be used by JADE agents.

5. Lessons learned

Implementing the prototype and applying it to several scenarios derived from our case studies identified a number of issues which require further examination.

5.1. The Nested Contract Net protocol

The final "inform" and "failure" messages in the protocol can be interpreted in two different ways: (a) as messages sent after production of the proposed product has finished and (b) as messages sent at the end of the planning stage. In case of interpretation (a), these messages can be used to coordinate production by sending them to the agent responsible for the next step in the production to signal "go ahead" or "stop, exceptional circumstances". In the second case it would allow the two involved agents to reschedule locally the tasks that failed, without involving the other agents (this would be a modification of the original CNP). Using the CNP as illustrated above, the wholesaler would be the only agent that could react in case of a failure and would be forced to react, or at least trigger reaction by the involved agents.

The second interpretation (b) would treat the final inform/failure messages to signify the result of the scheduling (planning) process. A different interaction protocol would then be needed to coordinate the production process itself.

5.2. Communication load

In the prototype implemented for the case study agent negotiation is simplified to selecting the best offer from several bids. Even this reduces the search tree by considering only one of the alternatives.

However, running some simple experiments demonstrated that even simple applications generate a substantial number of communication acts, which imposes an unacceptable overhead and slows down the application. Further research is therefore necessary to implement protocols which minimise the number of redundant communication acts. For example, the two

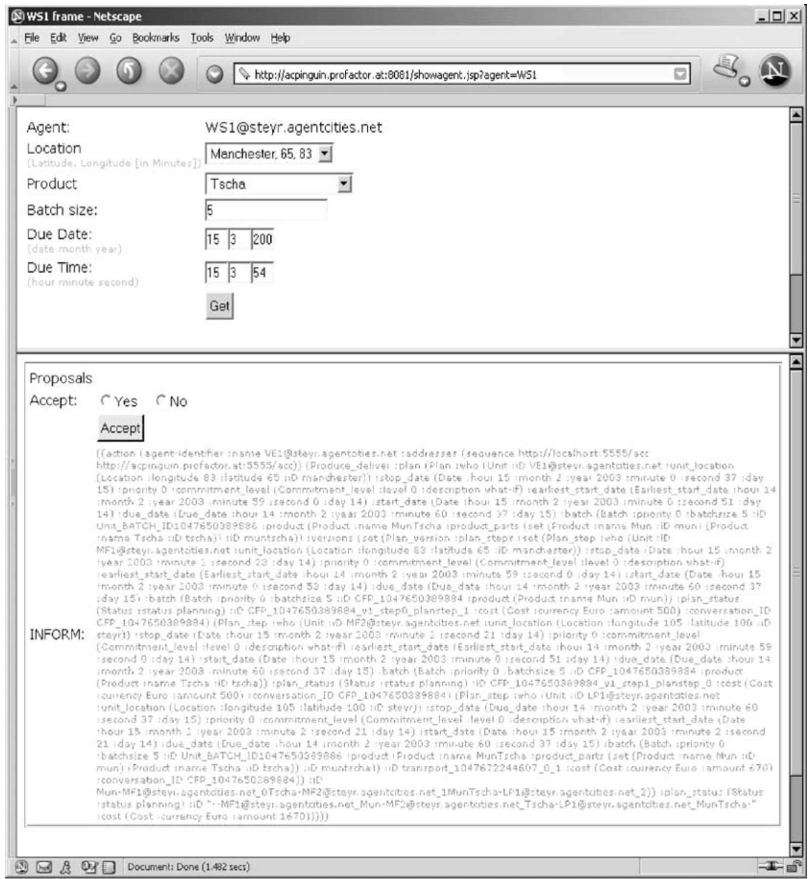


Fig. 11. Informing the user about a successful outcome.

interpretations of the interaction protocol described in the previous section generate different number of communication acts.

5.3. Holonic organisation of agents

The holonic organisation of the agents reduces the combinatorial explosion of possible steps in scheduling. As is the case in heuristic approaches, this improvement comes to the expense of not guaranteeing a globally optimum solution.

The benefit of the holonic organisation can be illustrated by applying it to the simple example described in Section 1. In the extended example, we consider a virtual enterprise that distributes orders to two manufacturing facilities. Each manufacturing facility has the same number of machines. Based on the holonic organisation, scheduling optimisation is done at three levels: *machine level*, *manufacturing facility level* and *virtual enterprise level*. As is the case with heuristic methods, this planning and scheduling method does not guarantee an optimum solution.

Based on the above assumptions, there are $2*(n!)^{m/2}$ possible solutions for the allocation of orders to machines (Fig. 13).

The benefit of using the holonic organisation for planning and scheduling can be seen clearly if the numbers of possible solutions for the same numbers of orders and machines for holonic and non-holonic organisation are compared. For example, for 19 orders and 18 machines it can be seen from Fig. 1 that the number of possible solutions for the non-holonic organisation exceeds 10^{300}. In the holonic organisation case, the number of possible solutions reduces to 10^{160} for the same numbers of orders and machines (Fig. 13).

To further illustrate the reduction in scheduling space achieved by the holonic approach, the numbers of the possible solutions of the non-holonic and the holonic cases taken from Figs. 1 and 13 are divided and the ratio is depicted in Fig. 14. As can be seen from Fig. 14, the holonic organisation reduces the number of possible solutions of up to 1000 times when two manufacturing facilities are considered. This effect increases with increasing number of manufacturing facilities.

Further research is necessary on how the Virtual Enterprise could influence the sub-holons to strive for achieving a global optimum, without loosing the performance advantages of hierarchical scheduling.

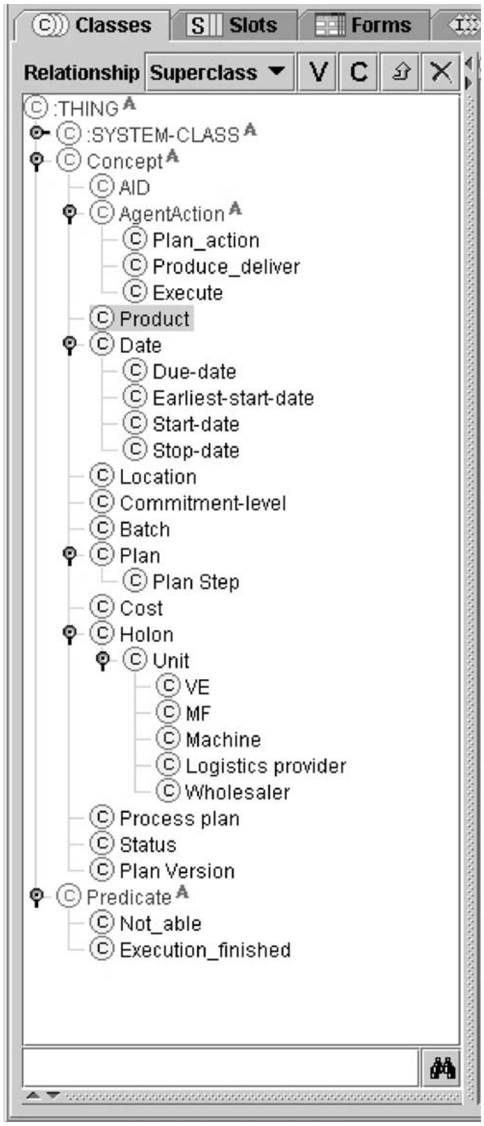


Fig. 12. Ontology tree.

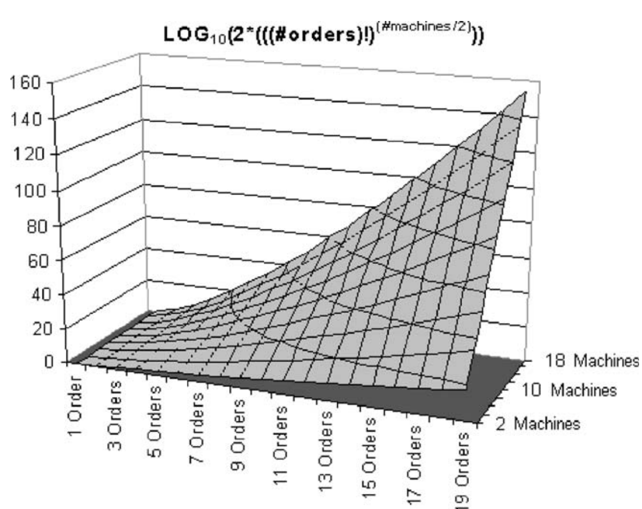


Fig. 13. Solution space of a simple scheduling problem using a holonic approach.

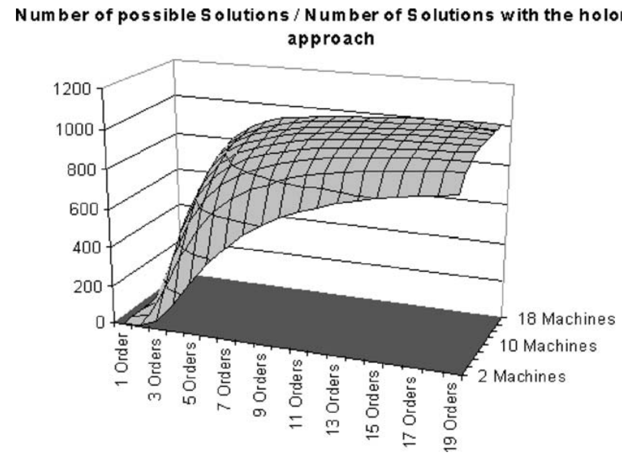


Fig. 14. Reduction in solution space in the holonic approach.

6. Related work

Examples of relevant approaches include logistics planning systems, centralised job-shop scheduling systems and holonic manufacturing systems. In all cases there are certain limitations which are addressed by our approach.

6.1. Logistics planning systems

DFKI have developed a prototypic software system TeleTruck (Bürckert et al., 1998) for planning, optimising, and monitoring of road haulage. The underlying approach is based on multi-agent technology, where physical objects of the transport domain (trucks together with their drivers, trailers, and load spaces) are modelled by active software processes (intelligent agents). Those agents are able to reason and plan on the basis of their individual resources and means provided by the corresponding physical objects. They are embedded in a common environment (a multi-agent system)—potentially distributed in a network of several computers which could be located at different transport departments—reflecting the communication and other interaction structure of the agents. The major difference between TeleTruck and our approach is that TeleTruck only focuses on logistics planning and does not consider manufacturing co-scheduling.

6.2. Centralised versus decentralised scheduling systems

In job-shop scheduling systems the scheduling problem consists of finding an (optimal) allocation of resources and time slots for a given set of tasks. However, scheduling is a combinatorial problem and thus suffers from combinatorial explosion. The space of possible solutions virtually explodes as the numbers of resources, tasks and possible time slots increase. Constraints (e.g. precedence constraints between tasks) may be used to somewhat narrow the solution space, but the problem is still NP-complete. The practical consequence is that the required calculation time for finding the optimal solution grows at least exponentially with the problem size (Parunak, 1991). (See also the discussion in Section 1). This implies that near optimal scheduling algorithms are the best possible ones. A near optimal scheduling algorithm spends time on improving the schedule quality, but does not continue until the optimum is found (Bongaerts, 1998).

In agent-based scheduling, basic system elements (usually resources and tasks) are represented by agents, thus reflecting the distributed nature of the physical system in the scheduling system (Parunak, 1999). Schedules are generated as a result of agent interactions. Like traditional, centralised approaches agent-based scheduling strives for near optimal solutions, because optimal solutions are not computable (see above). However, agent-based scheduling shows some substantial advantages to centralised approaches in terms of reactivity, localisation and adaptability.

Reactivity: if schedules get invalid (due to internal or external disturbances) re-scheduling imposes severe requirements on the reactivity of the scheduling system. To avoid standstill of the physical system, the scheduling system has to determine an alternative solution in a short period of time. In distributed scheduling the overall scheduling problem is decomposed into sub-problems which are assigned to independent processing units for solving. The sub-solutions are then compiled to a sub-optimal solution for the global schedule. Decomposition of the overall scheduling problem is facilitated by viewing the virtual manufacturing enterprise as a holonic system having a recursive structure (Bongaerts, 1998).

Agent-based scheduling exploits the power of parallel computation. The overall scheduling problem is decomposed into sub-problems which are assigned to agents for problem solving (each agent can be seen as an independent processing unit). However, agents can offer more than mere distributed problem solvers. For example, agents can perform sophisticated problem decomposition determined via agent negotiation. Furthermore, agents can achieve collaborative problem solving, for example they can support scheduling at different holonic layers and then merge the results.

Localisation: refers to another aspect of re-scheduling. Agent-based systems easily allow limiting the impact re-scheduling has on different parts of the system. While calculating alternative schedules, agents aim to find a localized solution, i.e. not to bother many other agents. This behaviour results in a preservation of large parts of the old schedule and avoids having to re-construct the whole schedule, saving computation time.

Adaptability: from an abstract point of view an agent-based system consists of loosely coupled modules (the agents) with autonomous features, organised in a decentralised way. In contrast to centralised, monolithic systems an agent-based system is easy to maintain and to adapt to new requirements. Single agents may be changed while the overall system is still operative (Parunak, 1999).

6.3. Holonic manufacturing systems

Agent technology has been used in Holonic Manufacturing Systems (HMS) (Gou et al., 1998; Leitão and Restivo, 2002; Bongaerts, 1998) which view manufacturing systems from a holonic perspective. In this type of systems, intelligent agents are organised in a recursive manner. A holon consists of a number of sub-holons, each sub-holon also consists of a number of sub-holons and so forth. This type of architecture allows integration of appropriate elements of hierarchical and heterarchical systems into an intelligent and open structure. An example of such a system is ADACOR (Leitão and Restivo, 2002). The ADACOR system is a concrete implementation of an HMS using agent technology.

However, a limitation of ADACOR, as is the case in HMS in general, is that it focuses on manufacturing scheduling and it does not consider planning and transportation scheduling between production units. We address this limitation by co-optimising production and logistics planning and scheduling using holonic organisation and agent technology.

7. Future work

We plan to extend our work on the system in the following directions.

7.1. Heterogeneous information sources

To demonstrate integrated supply chain management using heterogeneous technologies, we have also developed a bridge from FIPA to Web Services to link our agent-based holonic system with a simulated e-marketplace of logistics services. This is still to be integrated into the system, when it will enable the system to seek logistics services and to obtain simulated results regarding their availability and pricing.

7.2. Flexible SLA formation and fulfilment in virtual enterprises

In cross-organizational settings, service customers increasingly obtain, monitor and enforce the quality of service by signing Service Level Agreements (SLAs), formal contracts with the service providers guaranteeing quantifiable service at defined levels.

Agents are considered suitable for supporting formation and fulfilment of SLAs by negotiating the SLA formation and by interacting to manage SLA fulfilment. This re-quires extended agent interaction protocols associated with SLA representation formalisms and evaluation mechanisms.

An extension of the work reported here would apply the basic principles of agent-based virtual enterprise to create a mechanism for flexible SLA formation and fulfilment.

7.3. Optimisation of network performance in virtual enterprises

Agent technology is capable of providing reactive schedule execution systems. Such systems are also applied as decentralised distributed scheduling applications.

But research is missing in optimising the performance of the calculated schedules. Within the context of a virtual enterprise, the objective is to schedule orders in a cost efficient manner. To achieve this objective, one could compare centralised methodologies for resource scheduling in a virtual enterprise to the decentralised approach powered by software agents. The aim would be to identify a cost-efficient method of scheduling, which may mix and match the two main approaches for different aspects of the problem. Some initial observations in this direction are contained in Section 6.2.

8. Summary and conclusions

This paper has presented an approach which allows holonic agent systems to be used to support non-trivial integration of manufacturing and logistics service planning. The system is open with the necessary level of agent-interoperation ensured by using the Agentcities agent-platform interoperation infrastructure. The Nested Contract Net protocol was the main factor enabling our approach. A prototype was built which enabled us to investigate design and performance issues, and select a number of topics for further investigation. The main problem proved to be the number of messages generated by the Nested Contract Net, which may lead to poor scalability. On the other hand, using agents and contract net allows us to work with dynamically changing configurations of the virtual enterprise, and with dynamically sourced offers for logistics services.

Based on our previous research work, we are now working towards integrating the manufacturing enterprise's activities with those of its suppliers, customers and partners through an open, distributed integrated intelligent manufacturing environment. Therefore, this work could be extended in the following ways:

1. Dynamic formation of the virtual enterprise, demonstrating the open nature of agent systems.
2. Re-planning when plan is broken, using constraints and information provided in formally defined contracts and service level agreements.

References

Bongaerts, L., 1998. Integration of scheduling and control in holonic manufacturing systems. Ph.D. Thesis, KU Leuven.

Bürckert, H.J., Fischer, K., Vierke, G., 1998. Transportation scheduling with holonic MAS: the TELETRUCK approach. In: Nwana, H.S., Ndumu, D.T. (Eds.), Third International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'98), pp. 577–590.

Bussmann, S., 1998. An agent-oriented architecture for holonic manufacturing control. In: Xirouchakis, P., Kiritsis, D., Valckenaers, P. (Eds.), First International Workshop on Intelligent Manufacturing Systems. Lausanne, Switzerland, pp. 1–12.

Camarinha-Matos, L.M., Afsarmanesh, H., 2001. Enterprise modeling and support infrastructures: applying multi-agent system approaches. In: Luck, M., et al. (Eds.), Multi-Agent Systems and Applications, Ninth ECCAI Advanced Course ACAI 2001 and Agent Link's Third European Agent Systems Summer School, EASSS 2001, Prague, Czech Republic, July 2–13, 2001. Selected Tutorial Papers. Springer, Berlin, pp. 335–364.

Fischer, K., Müller, J.P., Pischel, M., Schier, D., 1995. A model for cooperative transportation scheduling. In: Lesser, V. (Ed.), Proceedings of the First International Conference on Multi-Agent Systems. AAAI Press, San Francisco, CA, pp. 109–116.

Fischer, K., Müller, J.P., Heimig, I., Scheer, A.W., 1996. Intelligent agents in virtual enterprises. In: Crabtree, B., Jennings, N. (Eds.), Third International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96). London, England, pp. 205–223.

Fox, M.S., Barbuceanau, M., Teigen, R., 2000. Agent-oriented supply-chain management. The International Journal of Flexible Manufacturing Systems 12, 165–188.

Goletz, T., Ferreira, J.J.P., 2000. Enacting dynamic distribution networks—the DAMASCOS project. In: Camarinha-Matos, L.M., Afsarmanesh, H., Erbe, H. (Eds.), Advances in Networked Enterprises—Virtual Organizations, Balanced Automation and Systems Integration. Kluwer Academic Publishers, Dordrecht, pp. 73–80.

Gou, L., Luh, P.B., Kyoya, Y., 1998. Holonic manufacturing scheduling: architecture, cooperation mechanism, and implementation. Computers in Industry (Special Issue on Manufacturing Systems) 37, 213–231.

JADE, 2003. Java Agent DEvelopment Framework, http://sharon.cselt.it/projects/jade/, last accessed 10th March 2003.

Jain, A.S., Meeran, S., 1998. A state-of-the-art review of job-shop scheduling techniques. Technical Report, Department of Applied Physics, Electronic and Mechanical Engineering, University of Dundee, Dundee, Scotland, 1998.

Koestler, A., 1976. The Ghost in the Machine. The Danube Edition. Hutchison & Co., London, ISBN 0-09-127130-4.

Leitão, P., Restivo, F., 2002. A holonic control approach for distributed manufacturing applications. In: Marik, V., Camarinha-Matos, L.M., Afsarmanesh, H. (Eds.), Proceedings of the Fifth IEEE-IFIP International Conference on Information Technology for Balanced Automation Systems in Manufacturing and Services, September 25–27, Cancun, Mexico.

Parunak, H.V.D., 1991. Characterizing the manufacturing scheduling problem. Journal of Manufacturing Systems 10 (3), 241–259.

Parunak, H.V.D., 1999. Industrial and practical applications of DAI. In: Weiss, G. (Ed.), Multiagent Systems—A Modern Approach to Distributed Artificial Intelligence. MIT Press, Cambridge MA, pp. 79–120.

Smith, R.G., 1980. The Contract Net Protocol. High level communication and control in a distributed problem solver. IEEE Transactions on Computers 29, 1104–1113.

Ulieru, M., Walker, S., Brennan, B., 2001. Holonic enterprise as a collaborative information ecosystem. In: Brennan, R.W. (Ed.), Workshop on Holons: Autonomous and Cooperative Agents for the Industry, 5th International Conference on Autonomous Agents 2001, Montreal, Canada.

Vancza, J., Markus, A., 1998. Holonic manufacturing with economic rationality. In: Xirouchakis, P., Kiritsis, D., Valckenaers, P. (Eds.), First International Workshop on Intelligent Manufacturing Systems. Lausanne, Switzerland, pp. 383–394.

Willmott, S.N., Dale, J., Burg, B., Charlton, C., O'brien, P., 2001. Agentcities: a worldwide open agent network. Agentlink News, 13–15.