Multi-Agent Coordination of Material Flow in a Car Plant

Hartwig Baumgärtel, Stefan Bussmann, Marcus Klosterberg
Daimler-Benz AG
Research Systems Technology
Alt-Moabit 96a, D–10559 Berlin
email: {baumg,bussmann}@DBresearch–berlin.de

Abstract

The planning of the material flow in a car plant has become a complex and difficult task which can no longer be accomplished satisfactorily without computer support. The main difficulty is to predict the effects of local decisions on the overall plant performance. In this paper, we present a framework for developing planning systems that compute plant-wide consistent production programs and thus master the effects of local decisions. In cooperation with a logistics department of Mercedes-Benz AG, the framework was applied to a subtask of the overall planning problem. We report on the design and the test of the prototype at a plant. Based on the experience with the prototype, we evaluate the framework and discuss how it meets the end-users' requirements.

1 Introduction

After several years of basic research, multi-agent systems have become a promising technology for innovative applications in fields like enterprise modelling, information management, telecommunications, network management, robotics, traffic, or production. The production domain, and in particular the manufacturing planning and control for example has received attention right from the beginning of multi-agent research (see e.g. [1,2]). The motivation to apply multi-agent technology in the field of manufacturing planning and control stems from the fact that, due to their enormous complexity, today's manufacturing systems are still fragile and inflexible (cf. [3]). On the other hand, multi-agent systems promise to build complex systems that are more robust and flexible than existing ones. But so far only few work has been reported on field applications of or field experience with multi-agent systems (see e.g. [4]).

In this paper, we report on field experience with a prototypical system designed for the planning of material flow in a car plant. First, we specify the material flow planning problem and list the user requirements. Then, we present the FAKOS framework which proposes three design steps for developing a planning system that meets the user requirements. In order to examine the appropriateness of the framework, we conducted a feasibility study, in cooperation with a logistics department of Mercedes-Benz AG, in which we applied the framework to a subtask of the given planning problem. We discuss the resulting design and report on the test of the prototype at a plant of Mercedes-Benz. We discuss the results of the feasibility study and examine how the framework met the user requirements. Finally, we conclude the paper with an outlook on the future work in this project.

published at The 2nd Conf. on Practical Applications of Intelligent Agents and
Multi-Agent Technology (PAAM'97), London, 1997, pp. 227 – 236

2 Problem Specification and User Requirements

One task, among others, of the production logistics department of a car plant is the planning of the car body flow during the production process. At the Mercedes-Benz car plant Sindelfingen (Germany), for example, passenger cars are constructed and assembled in eight major production steps which are grouped into three centers (see fig. 1). The body construction welds the lower and the upper part of the car. Afterwards, the painting puts several layers of paint on the car surface. Finally, the assembly installs the interior of the car, adds all outer parts, such as engine, drive train, doors etc., and performs a final check.

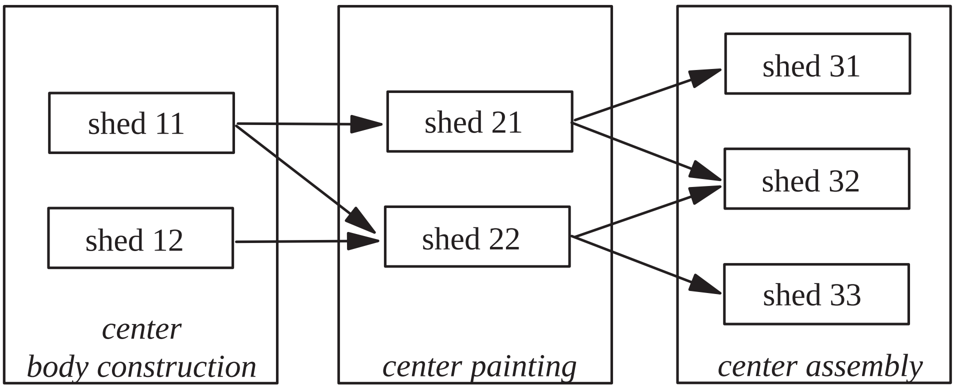

Figure 1: Factory Example

In the particular case of the car plant Sindelfingen, the material flow offers a certain degree of freedom in that (i) each production step contains several units with overlapping capabilities and (ii) a unit may supply several units of the succeeding production step (cf. fig. 1). The task of the production logistics is therefore to determine the production quota of each single unit and the routes of the car bodies between units, so that the resulting production program is consistent and the given production quota for the whole plant is achieved. A production program is consistent (from the point of view of the logistics) if and only if all cars produced by a production step are consumed by the succeeding one.

In order to fulfil this task, the logistics department performs a coarse planning on a weekly or daily basis in order to determine the quota of each single unit and the amount of car bodies passing a certain transportation line. The route of an individual car body is then determined on the spot according to the coarse plan and other parameters. In this paper though, we focus on the coarse planning.

The coarse planning of the car body flow experienced a growing complexity and an increasing dynamic behaviour of the production process over the last ten years. The growing complexity of the production process is caused by (i) increasing numbers of products produced; (ii) overlapping capabilities of units; (iii) differing working schedules of units; and (iv) different optimization strategies in each center (which are necessary due to the different production technologies used). The increasing dynamic behaviour of the production process is caused by (i) shortening product life cycles; (ii) frequent changes of the production technology; and (iii) process disturbances.

The current complexity, the current dynamics, and in particular the current tight coupling of the production steps make it practically impossible to optimize each center independently because any changes of the material flow immediately affect other centers. On the contrary, a center-overlapping coordination (that takes into account the local optimization strategies of each center) is necessary in order to achieve a globally consistent and optimal production program. In practice, this is no longer feasible without computer support.

What is needed is a decision support system that, given the current situation and an overall production quota of the factory, proposes production quota for units and a material flow plan. For such a system, the logistics department itself defined five requirements:

1. The complexity of the production process must be mastered.
2. The dynamics of the production process must be dealt with.
3. The transparency of the production process should be increased.
4. The resulting production program should be (globally) optimal.
5. The system must be configurable by the end-user.

In this paper, we present a framework for computing a material flow plan that meets these requirements and we report on our experiences with a first prototype.

3 The Framework

In order to solve the task described in the previous section, we have chosen a multi-agent approach. Multi-agent techniques are appropriate for the given task because (i) they allow to adequately represent the semi-autonomous production units and centers, their interactions and decision making; (ii) they implement dynamic decision making and interaction patterns; (iii) they offer coordination techniques for (semi-)autonomous units; and (iv) their use results in flexible and open systems.

The multi-agent approach chosen consists of three steps which determine the design of the planning system:

1. *model each production unit as an agent*
    a) *that controls its unit autonomously and*
    b) *that cooperates with other units concerning the flow of car bodies.*

   An agent that controls a production unit will be called *production agent*.

   The introduction of agents distributes the complexity of the planning problem onto two levels: The autonomy allows an agent to locally optimize its unit, whereas the cooperation assures that the material flow is globally optimized across the factory.

   But since a car body runs somehow through all production steps, the decisions of two production units must be coordinated even if they do not belong to adjacent production steps. Furthermore, two units of non-adjacent steps can only be coordinated if all units in-between are also included in the coordination process. Consequently, the coordination process must be ordered depending on the specific planning task. It is therefore necessary to perform a second step in the design process of the planning system.

2. *arrange the coordination process with the help of a (meta-level) control algorithm*

   The simplest way to implement a meta-level control of the coordination process, which is also sufficient for the given application, is to introduce a centralized coordinator. The coordinator initiates the coordination between agents of adjacent production steps and thus controls the coordination process.

   The control strategy followed by the coordinator though depends on the specific planning task to solve. If the planning task is to find an optimal plan given the quotas for the whole plant, then the coordination starts at the last production step (which is identical to the plant quota) and moves backward to the previous step (if a solution is found in the current step) until the first step is reached. If, on the other hand, a loss of capacity due to a disturbance has to be compensated, then the coordination starts at the affected step and moves forward and backward to adjacent production steps.

The final step in the design process, after the control strategy has been chosen, is to formalize the content of the coordination process. This includes the formalization of the information exchanged and the way the information is processed.

3. *formalize the content of the coordination with the help of constraints*

   The coordination is basically concerned with the material flow between units. Generally, a unit can accept any material flow that obeys the unit's restrictions. Thus, during the coordination process the production agents will first announce their restrictions on the material flow and then negotiate about assignments of material flow to units. Material flow restrictions and assignments can be very easily formalized through constraints and constraint variable assignments, as it is done in the following.

   The variables of the constraints denote the amount of each product type that is produced or consumed by a unit. Since the plant produces cars, the variables are finite-domain variables. Given these variables, relations between units can be expressed through linear equations and inequalities.

   An example of a relation between the units shown in fig. 1 is the following equation

   $$S_{11-22} + S_{12-22} = S_{22}$$

   with S_{11-22} denoting the amount of cars produced by shed 11 and sent to shed 22; S_{12-22} denoting the amount of cars produced by shed 12 and sent to shed 22; and S_{22} denoting the amount of cars consumed by shed 22.

   From the multi-agent point of view, the advantage of using constraints to express the relations between units, is that they can be easily communicated. Constraints have a clear semantics and associated inference mechanisms. Constraints thus, when communicated, fill in the message-content slot of speech acts which is always application-dependent (see e.g. [5,6]). And once communicated, the associated inference mechanism allows an agent to reason about the relations.

The design steps presented in this section build a framework, called FAKOS[1], for determining a material flow plan in a car plant. This framework has to be instantiated for specific planning tasks encountered in the production process. In the next section, we present such an instantiation and report on the prototype implemented. In section 5, we then evaluate the framework with respect to the experience gained with the prototype.

4 Feasibility Study

In order to evaluate the framework, we conducted a feasibility study. In cooperation with a logistics department of Mercedes-Benz AG, we chose a subtask of the task described in section 2, developed a prototype, and tested it at a car plant.

The task chosen is the daily production planning: Given the daily production quota and the current situation of the plant, compute a new plan that assigns each unit a quota of products to be

---

[1] FAKOS is at the same time the name of the resulting systems, but in this paper we will use FAKOS solely as the name of the framework.

produced and determines the routes of car bodies between units. The specification of the current situation includes the current capacities of units and the amount of cars currently in the factory.

The reasons for choosing this particular task were threefold: First, the task can be easily isolated from other planning activities. Second, for solving the task only few information is needed from process databases and the information can be easily provided. And third, the development of a prototype was feasible within 12 months.

In the following, we discuss the resulting design of the prototype and we present the results of the prototype test at the plant. A preliminary report on the system architecture and the control algorithm was already given in [7]. In this paper, we concentrate on the multi-agent aspects of the design, while aspects of constraint-programming can be found in [8].

The prototype was developed according to the steps of the framework presented in section 3. In the next three subsections, we discuss how each step of the framework was performed for the design of the prototype. In the last subsection, we then present the results of the prototype test.

4.1 Agent Modelling

For each production unit, we determined a process model describing the production process within this unit and its input/output behaviour concerning the material flow. The process model explains (i) in which phases cars are constructed, painted, or assembled; (ii) which parameters affect the unit's capabilities; and (iii) which alternatives exist for running the unit.

On the basis of the process model, we designed the decision making of the production agent associated with the unit (cf. fig. 2). The production agent has two tasks:

(i) to locally optimize the production process within its unit; and

(ii) to coordinate the car body flow with the adjacent units.

For the second task, the agent basically communicates constraints on its input/output behaviour, including for example capacity and circulation restrictions, and receives production quotas which lie within these constraints. The coordinator collects these constraints from the production agent, solves the constraint net, derives the production quotas from the variable assignment and sends it back to the production agents.

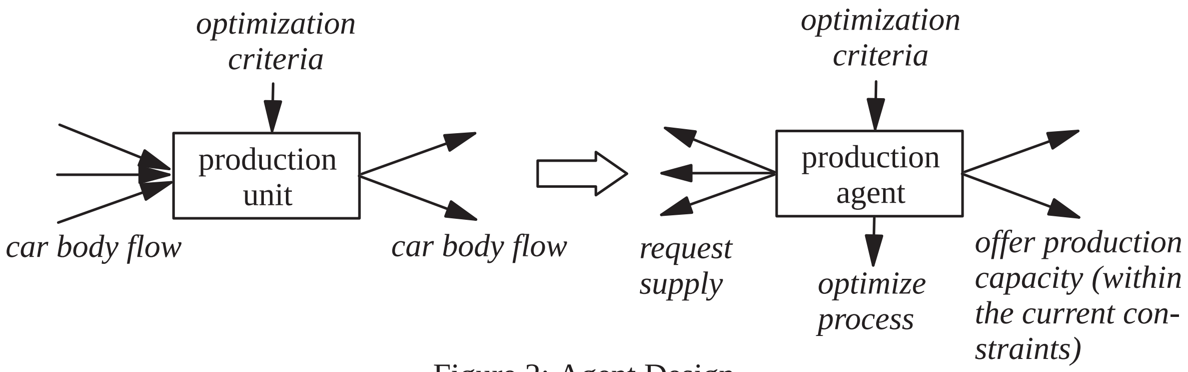

Figure 2: Agent Design

The above restriction of the coordination process may invoke the impression that all the decision making is located within the coordinator and the production agents only compute the constraints like a non-sophisticated function computes a value. But this is not the case. The decision making is equally divided between the production agents and the coordinator. The production agents decide which constraints are communicated[2] and the coordinator decides which of all the possible solutions (of the resulting constraint net) is chosen. For example, the production agents decide how much capacity they offer to the coordinator. Given the capacities of all production agents (of one step), the coordinator decides how much of the capacity offered by a certain agent is used in order to achieve the overall production quota of the current step. Furthermore, if the coordinator requests units to increase their capacity because the current constraint net has no solution, the decision whether to increase the capacity or not is within the responsibility of the production agent, i.e., it may deny the coordinator's request.

The reason for empowering the production agents to autonomously decide about the constraints communicated were threefold: First, each unit consists itself of a complex production process for which determining the constraints on the input/output behaviour of the material flow involves difficult predictions and decisions. Second, the production process of two units, in particular if they belong to different centers, may differ considerably. Third and most important, each unit is run by a different department who is responsible for the performance of the unit and which is therefore – to a great extent – autonomous in its decision concerning the unit.

To summarize, the tasks of a production agent are:

(i) to locally optimize the production process within its unit;
(ii) to decide which constraints are imposed upon the external material flow; and
(iii) to participate in the coordination process.

As argued above, the first two tasks heavily involve the agent's decision making capabilities, whereas for the last the agent merely participates in a protocol (see subsection 4.2).

The decision making of the agents, i.e., of the production agents and of the coordinator, were mapped onto the PRS architecture [9] and implemented with the agent-oriented tool dMARS [10] which provides a PRS architecture for each agent.

4.2 Control Strategy

The control strategy used in the prototype propagates the production quota of the whole factory (which is at the same time the production quota of the last step) from the last production step to the first. That is, for each step the coordinator collects the constraints from the production agents, adds the constraints expressing the production quota of that step, solves the resulting constraint net, and assigns the production quota resulting from the solution of the constraint net to the production agents. The production agents then compute a local plan that achieves the production goal for that unit, make reservations, and return details of the local plan to the coordinator if they affect the input/output behaviour of their unit. The coordinator thus receives all information on the required input of car bodies for these units and uses it as the production quota of the preceding production step.

The coordinator continues with this procedure until it either reaches the first step and thus has found a solution or there is no solution for the constraint net and it has to revise its decisions at previous steps of the coordination process (for more details on the control algorithm see [7] and also [8]).

From the point of view of the multi-agent system, the control switches between two levels of abstraction: the level of the coordinator and the level of the production agents. For one planning

---

[2] More precisely: The format of the constraints is fixed, but the production agents decide upon the constants of the constraints, such as for example the current capacity of the unit.

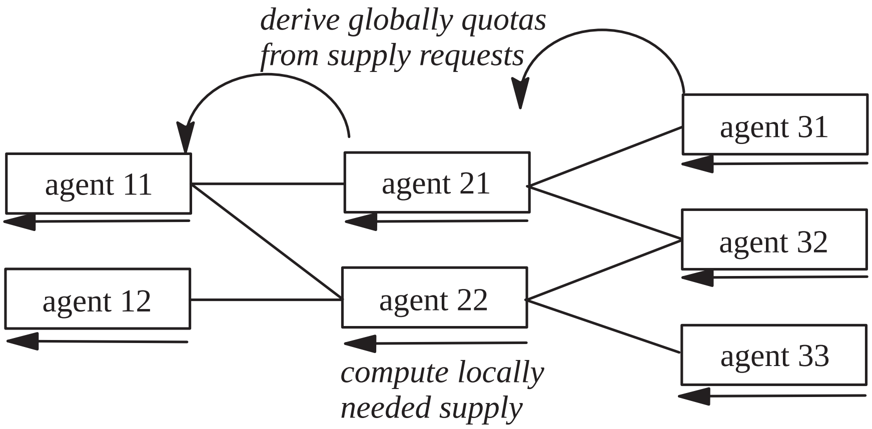

Figure 3: Propagation Scheme

step, the interactions between both levels are indicated in figure 4. Furthermore, as discussed already in the previous subsection, the decision making is divided between both levels in that the production agents decide upon the constraints and the local production plan, while the coordinator chooses a solution for the constraint net (if more than one exists).

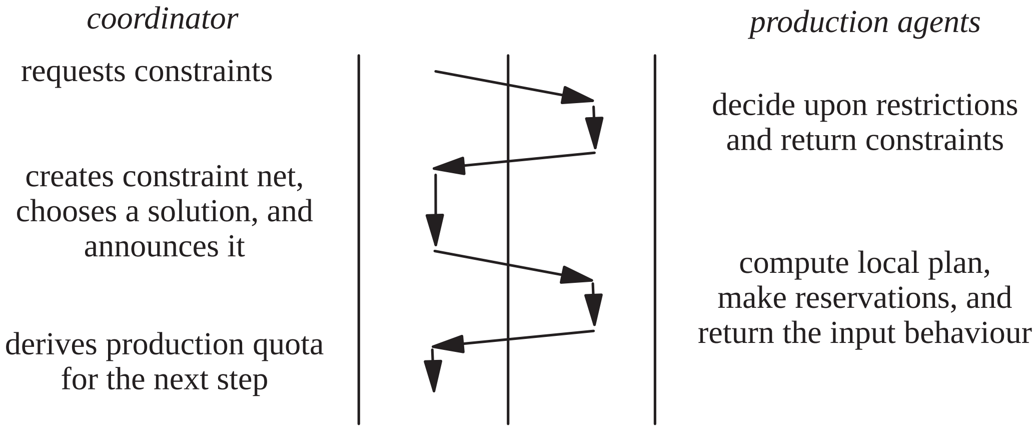

Figure 4: Diagram of Agent Interaction

The communication between agents is based on speech acts [11] and interaction patterns, such as the interaction shown in figure 4, which are expressed as cooperation protocols [6]. The cooperation protocols were also mapped onto the PRS architecture and implemented in dMARS.

4.3 Constraints

As discussed in section 3, constraints are used (i) to express restrictions on the input/output behaviour of units (i.e., capacity and circulation constraints); (ii) to express the flow of material (i.e., constraints on the flow and on the products that can be produced by a unit); and (iii) to express the production quota.

In order to solve the resulting constraint net, the prototype uses two constraint solvers, CLP(R) [12] and Oz [13], which are both integrated into the reasoning process of the coordinator. Details on the constraint processing and the optimization of the solution are treated in detail in [8].

4.4 Test of the prototype

The prototype developed was tested in cooperation with the logistics department at the plant Sindelfingen. The tests were conducted on the basis of the production data of May and June 1996. Two models, representing the factory at two different levels of abstractions, were created. The first model covers the whole factory, while the second models the center painting in more detail. Both models contain on average 25 to 30 units which results in approximately the same number of agents.

For each test run, information on the current situation of the factory was read from the production database and the production quota of the factory, which was taken from the current production program, was entered by the user. Additional tests were run with variants of the production data. In each case, the system was able to find a solution, i.e., a new plan, within four minutes on average which is a sufficient performance for the intended use of the system as a decision support in a manufacturing environment.

A comparison of the quality of the resulting plans with existing plans was only possible for a subset of the system's functionality since the current planning of the logistics department does not take into consideration all aspects of the overall planning problem. For the common subset, the prototype proved to be as good as the current planning method (performed by the expert).

5 Evaluation of the Framework

The feasibility study showed that the FAKOS framework is appropriate and practical for the problem domain described in section 2. The framework is appropriate in that it fulfils the requirements put forward by the end-users.

First, the framework masters the complexity of the production process because the process model of a production unit, which can be considerably different for each unit, is encapsulated in an associated production agent. Moreover, the decision making within the unit is performed autonomously by the corresponding agent – a system property that reflects the human decision making process. On the other hand, a production agent communicates only those restrictions to other agents which directly affect them and thus keeps the global coordination process at a minimum.

Second, the dynamic reasoning and communication capabilities of agents as well as the flexibility of constraint processing allow a FAKOS implementation to immediately adapt to disturbances or changes of the production process and thus to master its dynamics. On the one hand, agents are able to immediately react to changes in the production process, such as changes in the capacity. And on the other hand, the constraint processing can handle any constraints created by the production agents (which, of course, must satisfy a certain format).

Third, FAKOS increases the transparency of the production process in that it demonstrates global consequences of local actions or events. By varying the current situation of the factory (in an off-line test run) the user can determine optimal solutions under different conditions and thus estimate the global behaviour of the production process. For example, by simulating a loss of capacity in one unit the system can compute the necessary changes in the production program in order to compensate a disturbance. Note that in order to compensate a disturbance in one production step, it may be necessary to adapt also the production quota in other steps. The functionality of determining plant-wide consequences provided by the FAKOS framework is more than mere simulation of the production process because it also involves the planning process and how changes affect planning decisions.

Fourth, the coordination of the production units in combination with the use of constraint solving techniques assures the global optimization of the production program. For two adjacent production steps, the material flow is optimized with constraint techniques. For two non-adjacent production steps, the consistency and the global optimality of the material flow is achieved by the control strategy which transfers information between the corresponding constraint nets and thus assures the global optimality of the production program. Nevertheless, since the control strategy depends heavily on the planning task to be performed, the global optimality of the production program has to be proven for each control strategy anew.

And finally, representing internally the factory structure and the decision-making process in terms of the logistics department makes the system easy to understand and easy to configure by the end-user.

The framework is practical in that it showed sufficient performance during the feasibility study. For real models of the plant Sindelfingen, the system's response time remained below five minutes on average which is sufficient for a decision support system in the context of manufacturing. On the other hand, whether the performance will still be satisfactorily if the FAKOS framework is applied to other planning tasks of the problem domain is yet unknown and left to future research.

6 Conclusion

In this paper, we have presented a framework for the planning of the car body flow in a passenger car plant. First, we described the problem domain, discussed the complexity and the dynamic nature of the planning task, and listed the requirements of the end-users. Then, we presented the FAKOS framework which proposes a design process consisting of three steps: (i) modeling the production units as agents; (ii) arranging the coordination process with a control algorithm; and (iii) expressing the material flow relations in form of constraints. Our main hypothesis was that the planning system resulting from the design process will meet the user requirements. In order to examine this hypothesis, we conducted a feasibility study in which we applied the framework to a subtask of the overall planning tasks described in section 2. The prototype developed was tested in cooperation with the logistics department of the Mercedes-Benz car plant Sindelfingen on real production data.

The evaluation of the feasibility study showed that the FAKOS framework is appropriate and practical for the planning task of the car body flow in a car plant. The framework is appropriate in that it fulfils the requirements of the end-users (as shown in section 5). It is practical in so far as the prototype developed is efficient enough to solve the chosen planning subtask. Nevertheless, it is left to future research whether the performance will remain satisfactory when the framework is applied to the whole planning problem.

The success of the feasibility study has motivated the end-users to apply the FAKOS framework also to the disturbance handling. A project is currently being set up in cooperation with Mercedes-Benz in which a prototype for both planning tasks, i.e., planning of the production program and disturbance handling, will be developed and transferred into a product that can be installed at the plant. From the point of view of research, this will involve extending the process model, introducing more levels of abstraction, and developing a control algorithm for the disturbance handling. From the point of view of software engineering, this will involve transferring multi-agent technology into a product and coupling it with production databases.

published at The 2nd Conf. on Practical Applications of Intelligent Agents and
Multi-Agent Technology (PAAM'97), London, 1997, pp. 227 – 236

References

[1] H. Van Dyke Parunak: “Manufacturing Experience with the Contract Net”. In: M.N. Huhns (ed.), Distributed Artificial Intelligence, Pitman, London, 1987, pp. 285 – 310.

[2] P. Raulefs: “Cooperating Agent Architecture to Manage Manufacturing Processes”. In: W. Brauer, D. Hernandez, Verteilte Künstliche Intelligenz und Kooperatives Arbeiten, Springer Verlag, 1991, pp. 6 – 17.

[3] K. Schäfer, V. Patriti, P. Charpentier, P. Martin, D. Spath: “The Multi-Agent Approach in Scheduling and Control of Manufacturing Systems”. In: Proceedings of the 1st Int. Conf. on Practical Applications of Intelligent Agents and Multi-Agent Technology (PAAM’96), London, 1996, pp. 497 – 521.

[4] G. Fleury, J.-Y. Goujon, M. Gourgand, P. Lacomme: “Multi-Agents Approach for Manufacturing Systems Optimization”. In: Proceedings of the 1st Int. Conf. on Practical Applications of Intelligent Agents and Multi-Agent Technology (PAAM’96), London, 1996, pp. 225 – 244.

[5] T. Finin, R. Fritzson, D. McKay, R. McEntire: “KQML as an Agent Communication Language”. In: Proceedings of the 3rd Int. Conf. on Information and Knowledge Management (CIKM’94), ACM Press, November 1994.

[6] B. Burmeister, A. Haddadi, K. Sundermeyer: ”Generic Configurable Cooperation Protocols for Multi–Agent Systems”. In: C. Castelfranchi and J.-P. Muller (eds.), From Cognition to Action, Proc. MAAMAW’93, Springer-Verlag, 1995.

[7] H. Baumgärtel, S. Bussmann, M. Klosterberg: ”Combining Multi-Agent Systems and Constraint Techniques in Production Logistics”. In: Proceedings of the Int. Conf. on AI, Simulation and Planning in High Autonomy Systems, La Jolla, Ca, USA, March 1996, pp. 361 – 367.

[8] H. Baumgärtel: ”Distributed Constraint Processing for Production Logistics”. In: Proceedings of the Conf. on Practical Applications of Constraint Technology (PACT’97), London, 1997.

[9] M.P. Georgeff, F.F. Ingrand: ”Decision-making in embedded reasoning systems”. In: Proceedings of the 6th Int. Joint Conf. on Artificial Intelligence (IJCAI-89), 1989, pp. 972 – 978.

[10] Australian Artificial Intelligence Institute: “dMARS” at `http://www.aaii.oz.au/proj/dMARS-prod-brief.html`, 1996.

[11] S. Bussmann, J. Müller: “A Communication Architecture for Cooperating Agents”. In: Computers and Artificial Intelligence, Vol. 12, No. 1, 1993, pp. 37 – 53.

[12] S. Michaylov, P. Stuckey, J. Jaffar, R.H.C. Yap: “The CLP(R) Language and System”. In: ACM Transactions on Programming Languages and Systems, Vol. 14, No. 3, 1992, pp. 339 – 395.

[13] M. Henz, G. Smolka, J. Würtz: “Object-oriented Concurrent Constraint Programming in Oz”. In: V. Saraswat, P. Van Hentenryck (eds.), Principles and Practice of Constraint Programming, MIT Press, Cambridge, 1995, pp. 27 – 48.

published at The 2nd Conf. on Practical Applications of Intelligent Agents and
Multi-Agent Technology (PAAM’97), London, 1997, pp. 227 – 236