AgentSteel: Agent-Based Steel Production Optimisation

DFKI and Saarstahl AG
A Case Study

1. Introduction

This article presents a case study of an agent based system for steel production optimisation, AgentSteel, which was developed by DFKI for Saarstahl AG, a global steel manufacturer. The system uses a distributed planning and scheduling algorithm to calculate daily target schedules for the operations necessary to produce steel of different quality levels, according to given customer orders. It also monitors the execution of the schedules and responds to unpredicted changes in customer orders and to operational faults, by dynamically adapting the schedules. This document gives an overview of the system, the optimisation approach, the development process, and the main experiences and lessons learned.

2. The production optimisation problem at Saarstahl

Saarstahl AG is a German steel manufacturer, with a global presence in the steel production market. The production chain of Saarstahl involves a multitude of specialised and complex metallurgical and manufacturing processes [1]. First, a furnace factory converts the iron ore into pig iron, the raw material that is needed to produce steel. The pig iron is then sent to the steelworks, where it is processed and combined with other metals to produce steel of different quality levels, as requested by different customer orders. The steel is poured into blocks which are sent to the masticators to be given different shapes (e.g. wires, sheets, etc.). Firstly, these are delivered to different customers, such as part suppliers for the automotive, shipbuilding, or aerospace sectors. The overall production chain is characterised by uncertainties, changes in customer orders and is prone to various operational faults. Steel producers must therefore be flexible and responsive by adapting production plans fast, in order to deliver to the customer the required quality on the agreed date and at acceptable cost.

The focus of this case study is on the optimisation of steelworks production. Here, three converters transform the pig iron into steel of low quality, then five aggregates combine this steel with other ingredients like sulphur and chrome to achieve a higher quality level, and finally the steel is poured into blocks by five pouring aggregates, to be then manufactured into different shapes by the masticators. Inside the steelworks, the steel is transported into pans. A pan can take a charge of up to 170 tonnes of steel. These charges are poured in sequential order, grouped according to the quality of steel in that charge. For example, a pan containing a charge with chrome cannot take on next a charge without chrome. The problem is to find a daily schedule, consisting of a totally ordered set of sequences of charges for each pouring aggregate, and also to find the partially ordered set of all the pouring aggregates together (including those running in parallel). The process is sensitive to uncertain circumstances. For example, the quantity of the pig iron and its arrival time from the supplier determine the quantity and quality of the steel, but are in turn highly influenced by other orders the supplier has to fulfil for other steelworks companies, and therefore can change unexpectedly. As a result, a sequence may not be produced if insufficient pig iron is available. In addition, incoming orders from customers can change, resulting in continuous changes to daily target schedules.

The problem is, on the one hand, a job scheduling problem for the aggregates inside the steelworks, and on the other hand, a planning problem for the pans. The pans must be available in certain states at certain points in time in order to be able to produce steel at the necessary levels of quality. The complexity of the optimisation is significantly increased because the two problems influence each other: a delay of a schedule of a certain aggregate means the pan will not be available at the planned time for the next charge. Furthermore, the number of pans available is fixed and they can only be used within predefined time windows. In addition, the whole process is accident-sensitive and therefore a monitoring system should detect potential errors and re-plan as early as possible, ideally before the errors occur.

All these uncertainties make the rescheduling process too complex to be done manually. Moreover, even where manual planning is possible and is currently done by plant operators, the vast knowledge and experience that go into the task, and which were acquired over many years, are likely to be lost when the operators leave the company. Driven by this need, Saarstahl approached DFKI to deliver an automated planning and scheduling system that could be integrated with the existing production control system used in the process of converting iron into steel.

3. DFKI

DFKI is the German Research Centre for Artificial Intelligence, with sites in Kaiserslautern and Saarbrücken. Founded in 1988, DFKI now is one of the largest non-profit contract research institutes in the field of innovative software technology based on artificial intelligence. The centre focuses on the complete cycle of innovation: from research and technology development to demonstrators and prototypes, as well as commercialised products. The main areas of expertise include deduction and multi-agent systems, knowledge management, image understanding and pattern recognition, intelligent user interfaces, intelligent visualisation and simulation, language technology.

4. AgentSteel: The Agent-Based Solution

AgentSteel is the agent-based system developed by DFKI to address the planning problems of Saarstahl, and was developed on top of an interoperability framework to which DFKI contributed as part of the Athena project. Athena is a European Commission funded project, aiming to investigate and build a model for the interoperability of enterprise applications, and encompasses all aspects of interoperability, from technology components to applications and services. The industry sectors addressed are aerospace, telecommunications, automotive and furniture, so Saarstahl offered DFKI the opportunity to extend this system to a new and very different industry domain.

Because Athena uses a service oriented architecture approach to interoperability, any agent solution must be based on services, with services being wrapped into agents by adding autonomous behaviour. The services would have goals and would be able to generate and execute plans in order to achieve those goals, would exhibit commitment to achieve those goals, and would be able to cooperate with each other when their goals are interdependent in order to build compatible plans.

4.1 Optimisation approach

Agents are assigned aggregates to calculate all the possible schedules within a given time window, for a given set of tasks for that aggregate. An objective function is then used to find the optimal schedule (local optimisation). All the schedules (one from each agent) are then sent to a planning agent to determine an overall production schedule, using an objective function for optimisation according to the overall steelworks process (global optimisation). Both local and global optimisation functions are defined by the user. The planning agent coordinates and negotiates with the aggregate agents to find the global schedule.

The next step is the monitoring, observation and reorganisation task during the running production. Monitoring agents receive data continuously from the steel plant and determine whether this data conforms to the calculated schedules (i.e. whether it is in the pre-determined time windows). When discrepancies arise, they are detected as early as possible by the monitoring agents, and are displayed in graphical form, with additional textual explanations. Then, when an agent identifies a schedule outside the predicted time window, it attempts to find a new schedule using one of several predefined repair strategies. If a schedule affects the schedule of another agent (e.g. a time delay at a certain aggregate may result in another delay, visible only hours later, at another aggregate), the two agents will cooperate to find two compatible and optimal schedules.

The decentralised search for locally optimal solutions, combined with the planning problem, are easier to handle with a multi-agent system than with a centralised approach. In addition, in case of operational faults during production, an agent assigned to an aggregate has to find a new solution for itself according to its local objective function. The agents affected by these changes must then check that this solution is compatible with other schedules, and combine and optimise them according to their objective function.

4.2 Agent architecture

Every agent is built using a three-layer architecture (Figure 1):

1. The Behaviour-Based Layer implements the reactive behaviour of the agent, whereby a certain action is triggered in response to an external event (e.g. arrival of a new customer order). This is designed for quick response and does not involve any computation, because it sends the request to the Local Planning Layer.

2. The Local Planning Layer is responsible for the local planning for that particular agent and monitoring the execution of the plans.

3. The Cooperative Planning Layer is designed for coordination with the other agents in the system, using explicit coordination protocols (e.g. the Contract Net Protocol). This layer also deals with situations when operational faults cannot be handled locally,

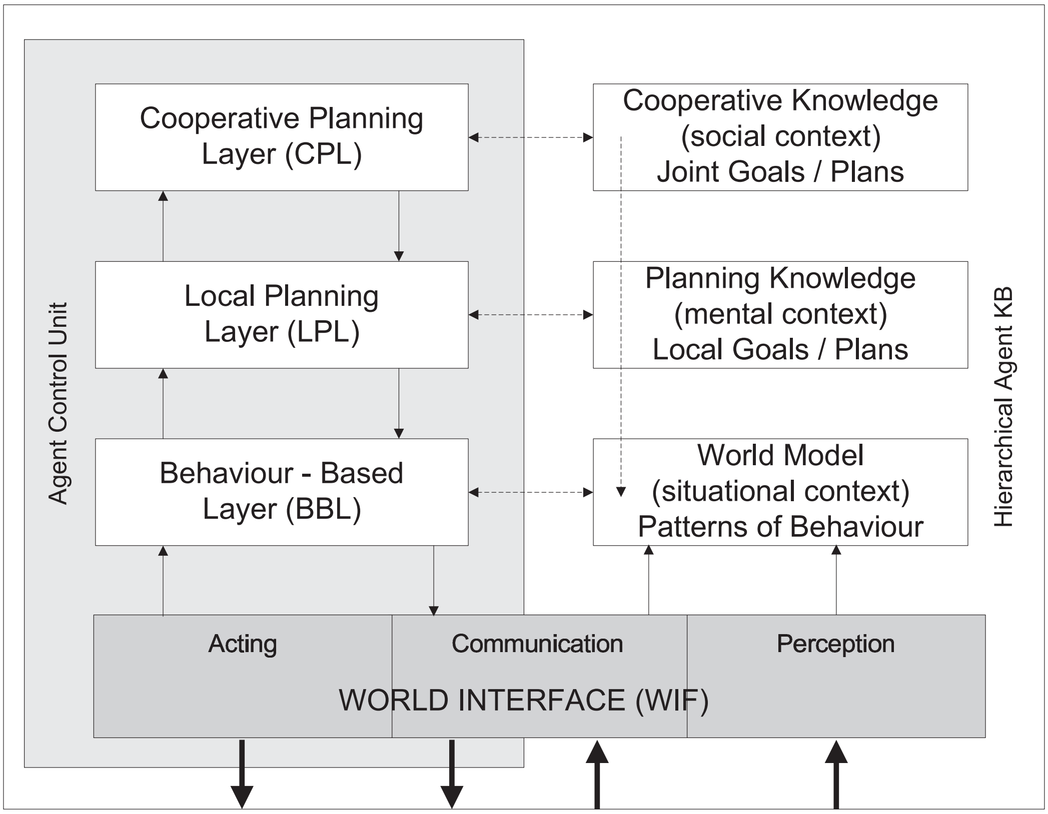

Figure 1: Multi-layer agent architecture. Source: [1].

and the corrected plan requires negotiation with other agents.

Similarly, the knowledge base of the agent is conceptually separated in three layers, one for each computational layer:

1. the world model contains the situational context based on which the reactive behaviour of the agent is determined;

2. the mental model is the knowledge base of the planning layer containing local goals and plans used in the optimisation at agent level; and

3. the social model contains knowledge about the joint goals and plans used in cooperation with other agents for global optimisation.

4.3 Integration with back-end systems

Within the steelworks, aggregates send information about the sequences of tasks they perform to a central database, from where the planner extracts information in XML form about the overall state of the steelworks and propagates it to the agents. These agents verify that the data corresponds to their calculated schedules. The agent-based system also relies on external databases for information necessary in the planning task, like the amount, quality and delivery time of pig iron from the furnace. Other data, such as delivery date of the steel or other adjustments in customer orders, are also important to determine the daily schedules. The integration with the databases at partner sites to retrieve this data is done via web services, with the processes expressed in BPEL and messages exchanged in SOAP.

5. System development process

The project had a duration of two and a half years, and involved two employees from Saarstahl (to connect the agent system to the back-end database and for the GUI implementation, respectively) and three from DFKI. During the first year, Saarstahl and DFKI were engaged in exploring, on a conceptual level, how agent technology applies to the steel production problem. DFKI built a first system prototype based on a simplified model of the factory, to demonstrate in principle that the problem could be solved, and validated this prototype by running simulations with artificial input data. Another year and a half were then necessary to extend and add more detail to the model, to link it with the control systems at Saarstahl and to validate it with real data. The agents were built in Java and no off-the-shelf toolkit was used in the development, due to commercial licensing restrictions. Communication between agents was designed using FIPA protocols for interoperability reasons, but without committing to a FIPA implementation platform. Integration with the back-end systems was of a read-only type, with no automated link between the agent system and the system that commands the execution of plans. The agent-based system was thus intended to act only as decision support for the human operator who must accept, approve and implement the suggested plan.

The strategy now adopted by DFKI is to extend AgentSteel to the full supply chain of the steel industry. A new project has already been initiated at DFKI to build a model of the steel supply chain using agents, including the interaction between iron suppliers and steel producers. The Jack Intelligent Agents framework and its BDI philosophy are used to model interactions outside the steel production company.

6. Lessons and experiences

- The process is complex and dynamic, due to the changing circumstances of the pig iron supply and changes in customer orders. In addition, the state of production must be constantly monitored and errors detected, and planning must take into account uncertainties. Since such tasks are too complex to be handled by humans, an automatic and responsive planning system is needed. However, because correctness and accuracy of plans are of critical importance, human planners are reluctant to use the automatically generated plans and schedules without first understanding how they were generated, which is not trivial because of the very complexity that must be addressed in the first place. Hence daily use of the system is needed for people to understand and trust system results.

- After validation through simulations in a lab setting, the system had to be deployed and validated in the real environment. To ensure that the system correctly models the factory, and that the schedules suggested are correct, the input data must also be correct. However, it was often the case that information retrieved from the back-end database did not correspond to the real state of the factory. In such situations, as a result of processing the wrong information, the plans suggested by the agents would also be incorrect and therefore not applicable. For this reason, an interface layer was added between the agents and the back-end systems, aiming to correct any potential errors in the input data.

7. Summary

This case study has presented the implementation of AgentSteel, an agent-based optimisation system for steel production, developed by DFKI for the German steel manufacturer Saarstahl. The system has been finalised and operationalisation will follow by having plant operators at Saarstahl experiment with it and eventually use in their daily planning tasks. However, further development is necessary to accomplish the following objectives:

- first, additional implementation is needed to translate the schedules generated by the agent-based system into executable plans, a task which is currently done manually;
- second, the system must be extended to support and optimise other processes across the supply chain of which the Saarstahl steelworks is part.

Thus planning and optimisation, as well as the integration with the necessary back-end systems, will have to address the furnaces, the masticators and the customers of the final steel products.

References

[1] Jacobi, S., Madrigal-Mora, C., Leon-Soto, E. and Fischer, K. (2005). AgentSteel: An agent-based online system for the planning and observation of steel production. In M. Pechoucek, D. Steiner, and S. Thompson, (Editors), *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems: Industry Track.* ACM Press, 2005.

Roxana Belecheanu and Steve Munroe
School of Electronics and Computer Science
University of Southampton
Southampton SO17 1BJ
United Kingdom
{rab2, sjm}@ecs.soton.ac.uk

Klaus Fischer and Sven Jacobi
DFKI GmbH
Stuhlsatzenhausweg 3
44123 Saarbrücken
Germany
{Sven.Jacobi, Klaus.Fischer}@dfki.de

www.dfki.de
www.athena-ip.org
www.staarstahl.com