MasDISPO: A Multiagent Decision Support System for Steel Production and Control

Sven Jacobi and **Esteban León-Soto** and **Cristián Madrigal-Mora** and **Klaus Fischer**
DFKI GmbH
German Research Center for Artificial Intelligence
Stuhlsatzenhausweg 3, 66123 Saarbruecken
Germany
Sven.Jacobi, Esteban.Leon, Cristian.Madrigal, Klaus.Fischer @dfki.de

Abstract

In the majority of cases, steel production constitutes the inception of the Supply Chains they are involved just as in automotive clusters or aerospace. Steel manufacturing companies are affected strongest by bull whip effects or other unpredictable influences along the production chain to the customers. Therefore, flexible planning and realisation as well as fast reorganisation after interferences are indispensable requirements for a competitive position on the market. In this paper, MasDISPO, an agent-based decision support system for production and control inside the steel works of Saarstahl AG, a globally respected steel manufacturer, is presented. It is based on a distributed online planning and online scheduling algorithm to calculate solutions supporting production and control inside the melting shop. It monitors the execution of their chosen solutions and responds to unpredicted changes during production by dynamically adapting the schedules.

This paper gives an overview of the system, the approach for solving the complex problem of steel production and control, the development process, the main experiences as well as lessons learned.

Steel Production at Saarstahl AG

Saarstahl AG is a German steel manufacturing company with a global presence on the steel production market. The production chain of Saarstahl involves a multitude of specialised and complex metallurgical manufacturing processes. First, a blast furnace produces hot metal from iron ore as raw material for the steel production. This hot metal is sent by rail to the steel works for the next production step. Here, steel of different quality grades is produced according to concrete customer orders and requirements. It is cast at continuous casting plants into billets, which are afterwards sent to the rolling mills. Probably, other production steps like a annealing or a surface treatment are needed before the end products are delivered to the customers – suppliers of the automotive, shipbuilding or aerospace sectors. The overall process chain is characterised by uncertainties, changes in customer orders and is prone to various discrepancies during production. Therefore, steel manufacturing companies must be flexible and responsive, by adapting production plans fast in order to meet customer requirements still cost-efficient.

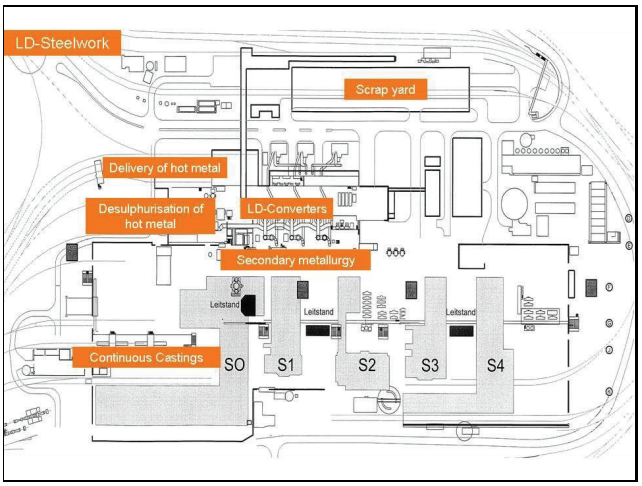

Figure 1: Steel works *Voelklingen*

The focus of MasDISPO is on the optimisation of the production inside the steel works at Voelklingen depicted above (Figure 1). The hot metal arrives by rail inside special waggons called *torpedos* at the steel works where it is filled into hot metal ladles. A desulphurisation treatment is needed before charging the hot metal into one of the three LD-converters. During the converter process the crude steel is produced. Depending on the quality grade of the specialised charge, alloying metals are already added during this process. The steel works' *Secondary Metallurgy* consists of five aggregates for various treatments like degassing or stirring. A detailed description of this *Secondary Metallurgy* follows in section *Solution*. Finally, the steel is cast into billets by one of the continuous casting plants. This is – very shortly summarized – the process of steel production inside the depicted steel works.

Inside the factory, the steel is transported in so called ladles. A ladle can be filled with a charge – this is one production unit inside the steel works – of up to $170t$ of steel. These charges are cast in sequences, grouped together according to the quality of steel in these total ordered sets. Once a sequence has started casting, it may not be interrupted. For instance, the tundish of the continuous casting plant may not fall under a defined level in between two heats, otherwise the

complete sequence fails. Hence, all the charges have to be synchronised during production to guarantee a frictionless overall process.

There are additional constraints on the charges which a ladle can serve. For instance, a ladle containing a heat with aluminium cannot take on next a heat without aluminium. The chemical composition of the heats have influence on the next two potential heats inside the same ladle, but the number of ladles available is limited, so the transitions of the qualities have to be planned very carefully in order to have a ladle available for each charge at its expected time.

Problem Description

The main problems that need to be solved are as follows: Given a daily target schedule the system has (i) to compute the points in time when the charges have to leave the converter, i.e. are tapped into ladles, (ii) make suggestions on which ladle should be used, and (iii) determine, for each aggregate, the points in time for all specified processes to begin. The aforementioned daily target schedule hereby consists of a total ordered set of sequences for each continuous casting plant to be produced on that certain day. Again, these sequences consist of total ordered sets of heats and each heat has a determined order of treatments at specified aggregates which have to be met in order to meet the demanded grade of steel. Overall, there are two main objectives to achieve: (i) The start times at the continuous casting plant has to be always met and (ii) no quality criterion is broken in the whole process.

The complete process is sensitive to uncertain circumstances. For instance, quantity of delivered hot metal and even more important its arrival time determine the start times of the sequences. As a result, it is possible that a sequence may not be produced if insufficient hot metal is available. In addition, incoming orders from customers can change, resulting in continuous changes to the daily target schedules – the input for the system.

The problem is – on the one hand – an online job scheduling problem for the aggregates inside the steel works and – on the other hand – an online planning problem concerning the ladles. The online job scheduling problem for the aggregates is clear. Summarizing the other problem, the ladles have to be available at certain points in time in certain specificated states in order to be able to produce steel at necessary quality grades. This describes a planning problem, even an online planning problem, due to the ongoing changing environment. The computational complexity is significantly increased because the two problems influence each other: a delay of a schedule on a certain aggregate implies that the ladle will probably not be available at planned time for the follow-up heat. Furthermore, the number of ladles available is fixed and they can only be used within predefined time windows. In addition, the whole process is sensible to disturbances and therefore a monitoring system should detect potential errors and re-plan as early as possible, ideally before the errors actually occur. All these uncertainties turn out to make scheduling and 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.

Application Description

In this section, the application is described in detail. In general, the system can be divided into two main parts. First, there is the monitoring. This client just displays the current state of production and actual planning above it as Gantt-Chart (**?**). This "Monitor Client" is of a read-only type, it queries periodically for new actual data from the database. The system compares it with its actual accepted schedules, detects discrepancies as early as possible and displays the results – meaning no interaction is possible. This application is needed along production inside the steel works at each working place which is influenced by the decisions of the operator and / or the system. This client is dependent of the "Simulation Client" described next. The monitorings basic schedules are the results of these simulations. Not until the operator accepts a certain solution, the "Monitor Client" is updated with a new basic daily target schedule. Hence, as long as the plant operator is simulating, monitoring works on the last accepted plan.

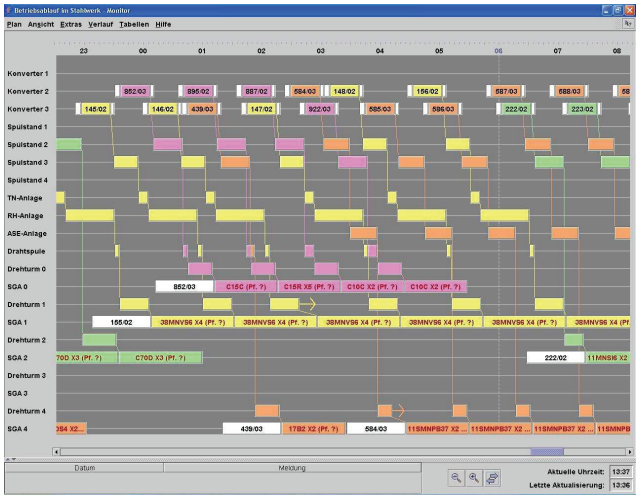

Figure 2: Monitor Client

Figure 2 *Monitor Client* shows a screenshot of this client. The GUI consists of a horizontal axis depicting time and one row for each aggregate taken into account by the system on the vertical axis. Thus, the schedules for each aggregate are displayed as well as the ones for the ladles on a second separated screen. Sequences for one specified continuous casting plant are colored uniform for clear view. Each step of a heat is displayed as a rectangle on the corresponding aggregate's row. If discrepancies from production to planning are detected, they are marked on the screen and additionally written in textual form into a separated field. If it cannot be

treated locally by the system, simulation mode is needed to return to normal production.

Second, there is the simulation part. Again, the current state of production and the actual planning above it are displayed, but now user interaction is required. This client is used by a single person, the production responsible plant operator. The operator is able to fix and manipulate any production relevant aspect he wants to and let the system calculate the consequences. For this, MasDISPO provides a set of dialogues to arrange these desired settings. Thus, the plant operator is able to compare several alternatives of how to continue with production and let the system check all potential side effects. Finally, he chooses a certain alternative. Fixing a sequence's start influences the availability of the ladles and, vice versa, assigning a certain ladle for a specific charge could influence on the sequence scheduling. If the operator now even wants to compare several alternatives, it is just too complex to be done manually and to keep the effects correctly in mind. This simulation mode is needed for arranging a new incoming daily target schedule and for fault recovery during the running production. Figure 3 *Simulation Client* depicts a screenshot of this simulation client. This user interface is quite similar to the monitor client, but it has additional features. These are the dialogues to set the desired values for all production relevant parameters.

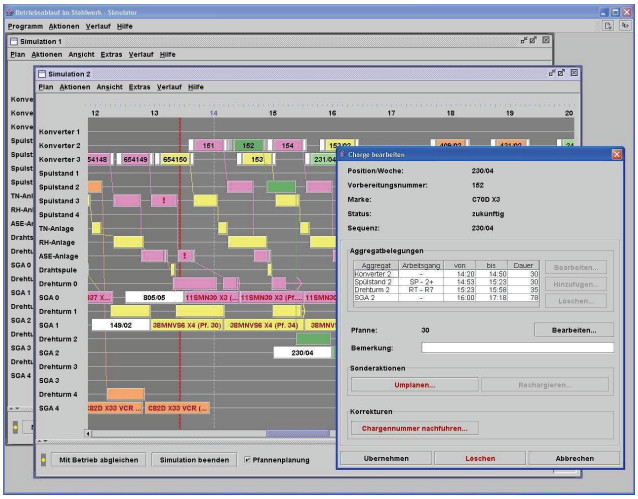

Figure 3: Simulator Client

Having quite shortly described the user interfaces, the description of workflow behind these front-ends follows right now. The planning department of Saarstahl generates the aforementioned daily target schedule every morning based on the concrete stock of orders and attended customer requirements. It consists of sets of sequences to be produced on designated continuous casting plants. Each sequence consists of an ordered set of heats. These heats consist of an ordered set of steps to be fulfilled. Hence, a daily target schedule consists of a partial ordered set of working steps to be planned by MasDISPO. It marks the initial input for the planner, but it also can be changed any time if required – after a sequence interruption for instance.

The system solves the scheduling and planning problems and presents an initial solution inside the "Simulation Client". Another difficulty was detected during development – the correct adaption of the actual state of production versus the planning part. The solution of this will be described in detail in the following section. The plant operator now has to make a detailed planning of this initial solution – meaning shifting suggested start times of sequences or choosing a certain ladle instead of the proposed one. He also could just to accept the system's suggested solution. As long as the operator is simulating of how to realize this daily target schedule, monitoring still is based on the old accepted plan. If he comes to a decision, he accepts this desired alternative and all monitoring clients distributed along the steel works' production chain control rooms are updated. This is – shortly summarized – the first task regarding the workflow of the application.

During normal operation, the plant operator also uses the monitor client for observing production planning, he has to rearrange detected discrepancies supported by the simulator, again. Effects of discrepancies could appear in production hours ahead. For instance, a delay at a continuous casting plant could affect a planned use of a ladle for a certain heat in another sequence hours ahead. MasDISPO also detects such drawbacks which might not be visible at first sight. The simulator is able to handle those operational faults and return to normal business up to a certain level. As long as the constitution of the daily target schedule is not changed, the system is allowed to make suggestions of how to rearrange, but changing the order of sequences or the quality of a charge are interferences which might not be done autonomously by the system. The operator has to give these instructions which are supported over dialogues by the application. The simulator then calculates all consequences and finally the operator again has to choose a certain alternative.

This is a brief overview on the general use of the deployed application. Remarks on design, solution, implementation and techniques used follow in the hereunder sections.

AI Technologies

In this section, the solution is described in detail. Also, an overview over the agent and internal system architecture implemented is given. The external architecture regarding the embedding of MasDISPO into the IT-environment of Saarstahl to ensure interaction with the other systems of the Supply Chain is described. The subsections point out the main technologies MasDISPO is based on.

Solution

The initial input is the daily target schedule. Additionally, the actual state of production is needed to fit the calculated initial solution into the running process. For each continuous casting plant, a total ordered set of sequences which again consist of a total ordered set of charges is received. Each charge has a specified order of tasks at determined aggregates inside the steel works. For each task, the expected amount of needed working time is given. Given that once a sequence has started casting it cannot be interrupted, the

continuous casting plant demands each charge at a fixed latest point in time. These times are propagated backwards through the aggregates of the steel works subject to required times at the specialized aggregate until the converters are reached. There, the point in time of the beginning of the converter process has to be determined. This part of the calculation is called *Backward Propagation*. Optimal points in time from converter to continuous casting plants would be reached for each charge, subject to no side-effects from the other heats using the same aggregates would be taken into account.

The converters are constrained by the fact that from tap to tap there has to be a minimum of 40 minutes time concerning a single converter. With two converters running in parallel, every 20 minutes a charge could be produced if distributed equally, which is not possible. During normal production, there are always three continuous casting plants running in parallel operation – sometimes even four. So, at least three different qualities of charges, formats to be cast, casting speeds at the continuous casting plants and further restrictions on the charges have to be taken into account. The aforementioned *Backward Propagation* leads to collisions at the converters since every single heat asks for its optimal tap-time and all these requests together do not fit in the pattern described above. The system has to decide which charges should be produced in advance at the converters. In order to avoid unnecessary cost, this time in advance has to be minimized concerning the complete daily target schedule. This is one global objective the planner has to reach. For minimizing this time in advance, the planner takes all the charges of the input into account in parallel passing a separated converter scheduling.

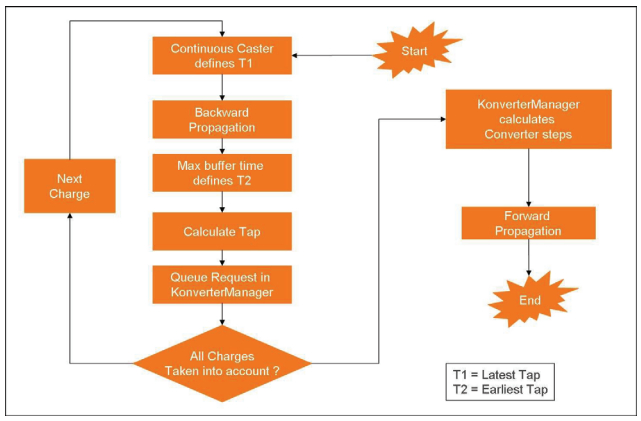

Figure 4: Process Flow Scheduling

In general, each charge has a certain buffer time which is normally taken at the ladle turret of the continuous casting plants. This time also could be used anywhere during the described process. Adding this buffer time already at the converter, the system propagates the calculated times back – but forward in terms of production – to the continuous casting plants. This process is called *Forward Propagation*. A time window with an earliest and latest start time for each aggregate is received. This is done with every charge of the daily target schedule. In order not to receive discrepancies with every new incoming data from the steel works, an exact calculated time is not used. Instead, this time window is used. It reflects a planning interval in which certain steps at specified aggregates with predefined durations have to be done. By use of these time windows, also more flexibility regarding the complete schedule of a charge is reached. Figure 4 depicts the explained process flow of this scheduling of aggregates again.

So, the sets of tasks for each aggregate are received. These aggregates – locally as an agent – calculate the complete set of possible schedules regarding their local objective function which might be changed by the user. Another agent receives the proposed schedules - one of each agent - and optimises them in respect to another user determined objective function for the global process. The result is the initial overall schedule. The described multiagent system (**?**), (**?**) consists of one agent for each corresponding physical aggregate including the ladles $L1 - Ln$ for n used ladles per daily target schedule and an additional so called planning agent with certain coordination and negotiation tasks. Additionally, each charge is modeled as an temporary agent

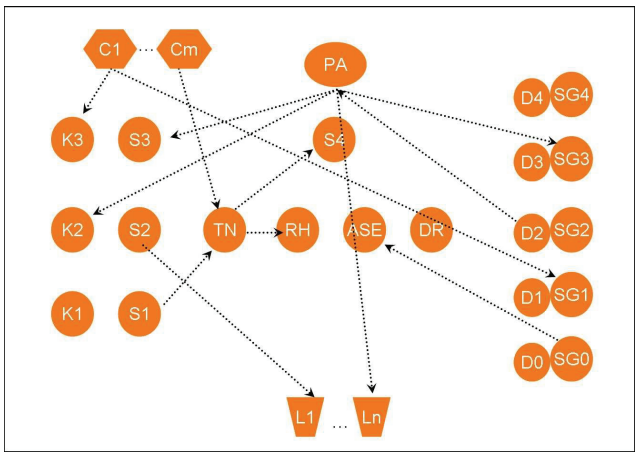

Figure 5: Agent Classification

observing its own schedule for its "life-cycle" during production – figure 5 shows the agents as defined in the system. It consists of $C1 - Cm$ agents for m charges of the daily target schedule, $n = |L|$ agents for the n used ladles, one agent initializing the processes and responsible for correct references and instances during parallel use and one agent for each corresponding aggregate of the steel works. These are the 3 LD-converters $K1 - K3$, 4 "stirring stations" $S1 - S4$, the other four aggregates of the *Secondary Metallurgy* with their specialised steps and finally the continuous casting plants $SG0 - SG4$ with their corresponding ladle turrets $D0 - D4$. The arrows between these agents constitute the communication among them.

So far the planning problem has not been taken into account. The planner has to ensure that each charge - meaning each quality - receives a compatible ladle. To do so, MasDISPO calculates for each quality a complete set of possible

follow-up qualities for the same ladle, regarding the complete target schedule. As a result, a compatibility matrix $A(m \times m)$ is received, for all m charges contained in the target schedule. Based on this matrix, a directed graph $G(v, e)$ is created, in which the vertices constitute the charges, which have to be labeled with at least one of the given ladles. A directed edge $(u_i v_i)$ exists, if charge v may be tapped into ladle i subject to charge u beeing the last charge before v in ladle i – meaning $A(u, v) = 1$. For a given set L of ladles, $|L|$ paths have to be found through this graph G, whereas all vertices must be labeled, meaning no charge is without a ladle. Before starting this search, a graph normalisation is performed, in which edges may be deleted based on the following theorems.

Theorem 1 (Incoming Edge) *Let $G(v, e)$ be the directed graph based on $A(m \times m)$. If vertex v has only one incoming edge e (uv), delete all other outgoing edges e' from u of G.*

Since v has only one incoming edge, it is clear that this transition is required, to ensure that no node is left disconnected and not labelled. Thus, all other possibilities for the ladle to choose from at u are irrelevant and hence these edges may be pruned.

Theorem 2 (Outgoing Edge) *Let $G(v, e)$ be the directed graph based on $A(m \times m)$. If vertex u has only one outgoing edge e (uv), delete all other incoming edges e' to v of G.*

Transition (uv) is mandatory if the ladle should be used later on. Sometimes it is possible, that a ladle should not be used furthermore in the corresponding daily target schedule, but instead remain its current state, in order to be used again for an oncoming quality, with special requirements, which are met by this specific ladle.

The graph normalisation is recursive, any time an edge is pruned because of one of these reasons, the normalisation has to be started again. As result, the normalised graph $(G'(v, e))$ is created.

MasDISPO starts the assignment $LadleSearch(G'(v, e))$ – as described in Algorithm 1 *Ladle Assignment* – based on the normalised graph G. The starting vertices are determined by the synchronisation to the actual state and the minimum of outgoing edges. For all identified starters a unique path through the graph has to be found. Having finished this, an initial solution for a daily target schedule has been calculated.

One great challenge identified during implementation was the synchronisation to the actual state of production. Charges already completed and charges which haven't started at all presented no problems, but running sequences and charges caused more problems than expected. For instance, if the ordering inside some schedules is as planned, semantic interpretation and merging of schedules in production and future planned schedules is problematic. On the one hand, there might be data appearing too early for the planner, but on the other hand, if the planner adapts it, additional data will be expected, corresponding to schedules that can be missing. Therefore, an interface layer has been added for plausibility check on correctness of data and additional planning work inside the agents has been introduced.

Algorithm 1 Ladle Assignment

```
procedure LADLESEARCH(G'(v, e))
    starter s = pickstarterMinOptions()
    for all ladle options do
        propagateOptionToOutgoingLinks(s)
        for all outgoing links do
            assignLadleOptionFromStarters(s)
            if currentOutgoing.Count() == 0 then
                node is a leaf
                if allLeaveshavesolutions then
                    addStatetoSolutionList()
                else
                    starter s' = pickstarterMinOptions()
                    propagateOptionToOutgoingLinks(s')
                end if
            end if
        end for
    end for
end procedure
```

The next and even more complex task of the system is the monitoring, observation and reorganisation task during the running production. So far, no multiagent system would have been definitely necessarily used although a decentralised approach with local search for an optimal schedule subject to local objective functions combined with the planning problem are easier to handle by a multiagent system instead of a centralised approach. In cases of operational faults during ongoing production the aggregates themselves have to find a new solution for themselves regarding their local objective functions. Other agent affected by these changes just combine these solutions checking whether they are compatible and optimises them in respect to their objective function. This is one main advantage for the use of a multiagent system instead of a centralised approach where this scenario would cause even more computational costs.

During monitoring the agents ask isochronously for new production data from the steel works. Received signals are compared to their calculated schedules. If the production is inside the calculated time windows, everything is working all right. If the signals are going to run out of time or any other discrepancy to the planned production shows up, this is detected as early as possible. All results are displayed in the aforementioned monitor clients which are accessible from any station of the Supply Chain. To do so, every agent has a so called *EventListener* who takes care whether the new incoming data is relevant or not. At the moment, every 60 seconds a snapshot of the production inside the steel works is taken and propagated through the agents. A description of needed interfaces and architecture of the agents follows in the next section in detail.

If an agent recognises that its schedule is running out of the determined time window, he immediately tries to rearrange this. Depending on the severe of the operational fault, the agent can handle this locally or has to contact others

which might be affected by his reactions. Following repair strategies are implemented:

- **Free float checking:** It might be that the aggregate is not used directly after that step and hence a delay has no side effect on the following steps of this charge. No further actions are necessary.
- **Internal order changes:** In case the delay has already effects concerning the aggregate it might end up into a cascading effect over all aggregates inside the steel works. The agent has to change the internal schedule. To do so, he has to contact all other agents which could probably be affected by this action. For this, the *Contract Net Protocol* (**?**) is used.
- **Alternative aggregate:** It could happen that the aggregate is not able to fulfill that specific task at all under these new circumstances, but certain other aggregates might do the same task. Using *Simulated Trading* (**?**) the agent tries to sell this task to another which produces less costs.
- **Sequence interruption:** In some cases a sequence has to be interrupted. Hence, a new daily target schedule has to be set up by the planning department of Saarstahl AG.

In the first case, the system just informs the user that the calculated time windows cannot be satisfied, but that no further actions are needed. The other three cases have more impact on the complete production inside the steel works. The reactions of MasDISPO on these faults need to be approved or even probably modified by the user. Therefore, in these cases the *Simulation Client* is started immediately. The system proposes some alternatives of how to handle these faults, but the plant operator has to submit them finally.

Every delay in the production of one charge might influence the complete production inside the steel works. It is very difficult to foresee where problems will appear as consequence. Each agent meaning each aggregate, ladle or charge tries to satisfy its own goals; additional optimisation aspects concerning local criteria are also satisfied by specific agents with particular knowledge bases. Every action is monitored and potential clashes are shown at an early stage of time with additional several repair suggestions from which the user might choose one.

A complete report over all restrictions which had to be kept and involved by the deployed system is quite impossible to give, but even this overview proves the high complexity of problems the system has to deal with. The approach of dividing this complex problem into several smaller ones in which local restrictions and goals which even might be concurrent to others can be modelled clearly and separated from each other. Each agent takes care of its own schedule, reacts as soon as possible on any discrepancy to the planning and knows which other parts are effected by any rearrangements. The very huge set of restrictions also changes quite often which also can be handled more suitable by this decentralised approach.

Agent Architecture

Multiagent systems are particularly well-suited for the scenario described in this paper because they allow a very natural mapping from the entities occurring in reality to agents.

InteRRaP (**?**) is a generic agent architecture for situated agents that integrates reactive behaviour and deliberation. The architecture was designed for agents that exist within multiagent systems and thus some emphasis is on the communication aspect. This architecture has been used as initial reference for the internal organization of agents inside the planner.

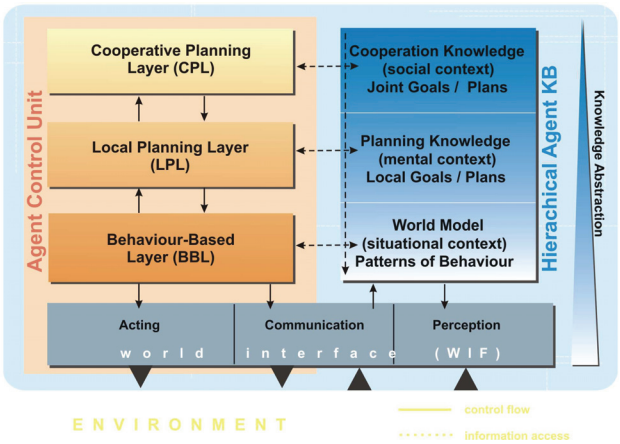

Figure 6: Agent Architecture INTERRAP

As depicted in Figure 6, InteRRaP is a layered architecture that consists of three concurrent layers:

- **Behaviour Based Layer (BBL)** This layer implements the reactive behaviour of the agent, i. e. this layer reacts to external requirements without any explicit reasoning, thus it reacts very fast.
- **Local Planning Layer (LPL)** This layer performs the planning process of an individual agent, it is also responsible to monitor the plan execution of the agents current plan.
- **Cooperative Planning Layer (CPL)** This layer is responsible for the coordination with the other agents within in the multiagent system. The coordination with the other agents is achieved with explicit negotiation protocols.

All these layers run concurrently, the intra-agent coordination between these three layers is achieved via the knowledge base. The knowledge base is conceptually divided in three layers (world model, mental model, social model), but each layer has access to the knowledge on every level. The conceptual discrimination, however, allows for a clearer design because most of the information stored in the knowledge base can be associated with a particular layer. The InteRRaP architecture offers a generic framework for agent design that must be instantiated for the particular needs of a concrete scenario. Concerning our system this architecture is very useful. The Behaviour Based Layer takes care of the signals during the monitoring or new arriving demands from the outside. The Local Planning Layer is responsible for the local optimisations and calculations concerning only this aggregate/agent and finally the Cooperative Planning Layer is used for situations in which interaction with

other agents is needed meaning cooperation inside the steel works is needed to return to an optimal plan execution. This layer is needed, if operational faults occur which cannot be handled locally and affect other aggregates as described in the former section.

Internal Architecture

The internal architecture of the implemented system is a Three-Tier Architecture (**?**). Inside the steel works each aggregate sends data of the steps it fulfills to a central computer. In fixed time-intervals the steel works' computer system sends telegrams over a host to the database. The database as backend of the architecture performs a first plausibility check on these received telegrams before writing them into the tables. Between the database and the planner are several different interfaces. One for providing the daily target schedule, one for sending actual production data and another one regarding the metallurgical restrictions of the planning problem. The exchange format is XML (**?**). The planner as the middle tier asks continuously for information from database, performs another plausibility check on the data and propagates it to the corresponding agents. The agents have the knowledge to check whether the received real-time data is still compatible to their calculated schedule or if there are time differences that represent some difficulty for the global plan. The clients as front end receive the results and visualise them. The monitoring client just visualises the state of production and planning, whereas the simulation client offers all necessary dialogues for the interaction to the user as already described. The most important objects of this architecture and their interconnections are depicted in figure 7; the arrows between the objects denote the flow of control or the flow of data, respectively.

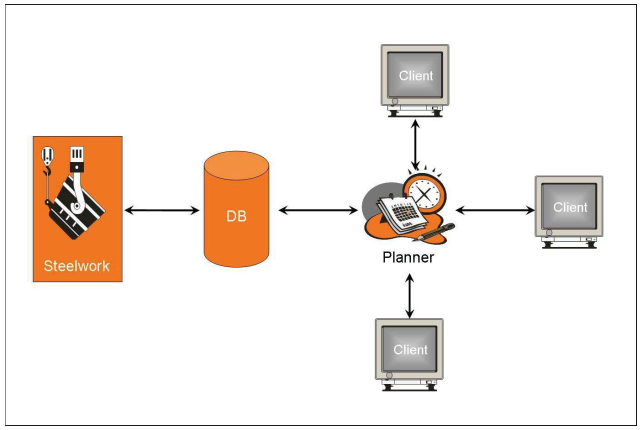

Figure 7: Internal Architecture

External Architecture

MasDISPO is not a standalone system, it is embedded in the IT-environment of Saarstahl with interfaces to interact with other parts of the complete Supply Chain. Therefore, also an external architecture is needed.

Planner-relevant data like amount, quality and delivery time of hot metal from the blast furnace are important. Also, concerning subsequent phases in the Supply Chain, like warehouses or rolling mills, it is relevant to incorporate information about delivery date of steel or fine adjustment of orders to configure a detailed planning. The planning system for the production inside the steel works cannot control these decisions, but negotiations can be supported by agents (**?**) using Web-Services. Therefore, an external service-oriented architecture (**?**), (**?**) has been developed. It provides Web-Service interfaces to other partners involved, with which relevant decisions can be made.

BPEL (**?**)is used to express the real business processes behind these negotiations. Then, WSDL (**?**), (**?**) is used to describe the endpoints behind these negotiations – i. e. the involved partners. The messages sent between these partners are realised as SOAP-messages. By using the described service oriented approach, flexibility is kept and interfaces are modelled for a further extension.

Application Use and Payoff

One benefit is the *Simulation Client* itself. This part of the application is completely new, the plant operator did not have anything comparable before. He is able to compare several settings under specific assumptions, using so called what-if scenarios generated withing seconds, for selecting a more stable plan. This is a significant advantage reached by the new planning system. The operator doesn't need to rearrange schedules that often anymore. Planning and production have become more reliable.

The delivery quantity of hot metal per day is fixed and covered by the production capabilities of the steel work, hence an increased overall throughput isn't a goal. But since the system started operating, a bottleneck aggregate of the steel work has improved its throughput: The RH-degasser – needed for certain qualities with higher grades, which normally traduces to higher value – has throughput increase of 10 %. Since there was not any monitoring facility available for the *Secondary Metallurgy*, the information visibility provided by the new system helps engineers and operators to optimize throughput at this bottleneck.

Compared to the old system with its planning horizon of only a few hours, the new one shows an another improvement. The planning horizon depends on the input which is normally 36 - 48 hours. The system calculates and shows all numerous production-relevant effects of the decisions made. Moreover, the old system monitored the converters and continuous castings, only. The new one, also takes the complete secondary metallurgy into account.

The deployed system has as a result, an increase in planning stability, an extended planning horizon and an extension to cover all the aggregates. These where features that were not present before and their effect is therefore hard to measure. Even though it is very clear that these features are significantly important, not only as an improvement, but further more, as enablers of features previously abscent.

Application Development and Maintenance

The project had a duration of three 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 – a feasibility study –, Saarstahl and DFKI were engaged in exploring, on a conceptual level, how agent technology applies to the steel production problem. A first system prototype based on a model of the factory was built to demonstrate that the agent approach could solve the aforementioned problems. This prototype used revised real data. Another two years 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 online data. The agents were built using 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 read-only, 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 system for the operator, who must accept, approve and implement the suggested plan. But this will be switched over the course of the follow-up project.

Maintenance is performed by the same group responsible for development. Since there is a follow-up project to extend the system on the complete Supply Chain, it is difficult to separate maintenance and new development from each other. New interfaces have to be arranged and of course normal bugfixing is also to be done on the running application. For instance, the metallurgical restrictions are maintained by metallurgists and the results have to be adapted by MasDISPO. Currently, one issue beeing focused is an interface to an expert system based on decision tables, to isolate this area from the systems. This frequently-changing domain knowledge should be maintained separately and used only as an input for the complete system.

Experiences

The process is complex and dynamic, due to the changing circumstances of the hot metal supply and changes in customer orders. In addition, the state of production must be constantly monitored and variations must be detected. Uncertainties have to be taken into account by the planner. Since such tasks are too complex to be handled manually, an automatic and responsive planning system is needed.

After feasibility study 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.

This case study has presented the implementation of MasDISPO, an agent-based optimisation system for steel production, developed by DFKI for the German steel manufacturer Saarstahl. The system is finished and operational, plant operators at Saarstahl have experimented with it and currently is is being use in their daily planning tasks. The operators are already relying on the system's generated plans, they settle and evaluate their daily planning tasks first with the system, before it is entered into as executable plans in the environment of the steel works. It has been running since August 2006. Few errors of lower relevance have been found. Additional spin-offs were created for the logistics of the complete steel works, which are also using the system for their planning.