Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

MasDISPO_xt: Annealing Furnace Planning inside the Supply Chain of Steel Production

Sven Jacobi [*] Esteban León-Soto [**] Cristián Madrigal-Mora [**]
Klaus Fischer [**]

[*] *Saarstahl AG, Hofstattstrasse 106, 66330 Völklingen, Germany, (e-mail:Sven.Jacobi@saarstahl.com)*
[**] *DFKI GmbH, German Research Center for Artificial Intelligence, Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany, (e-mail: {Esteban.Leon, Cristian.Madrigal, Klaus.Fischer}@dfki.de)*

Abstract: The production of steel normally constitutes the inception of many Supply Chains in different areas of industry. Therefore, steel manufacturing companies are strongly affected by bull whip effects and other unpredictable influences along their production chains. In the course of these integrated operations, making the right decision at a certain stage can be the difference between earning or losing a great benefit. Improving their operational efficiency is required to keep a competitive position on the market. Therefore, flexible planning and scheduling systems are needed to support these processes, which are based on considerable amounts of data, hardly processable manually anymore. MasDISPO_xt is an agent-based generic online planning and scheduling system for the observation on MES-level of the complete Supply Chain of Saarstahl AG, a globally respected steel manufacturer. This paper concentrates on the planning of the annealing furnaces as a representative example of the rough and detailed planning required in such an environment. The allocation of available capacities of annealing furnaces including alternatives to accepted orders is based on *Simulated Trading* to produce an evenly distributed rough planning regarding flexibility. As representative for the detailed planning, the solution for the batch-type annealing furnace is explained in detail.

1. STEEL PRODUCTION AT SAARSTAHL AG

The production chain of Saarstahl AG consists of a multitude of specialised and complex metallurgical manufacturing processes with several dependencies among them. First, a blast furnace factory produces hot metal from iron ore, as raw material for the steel production. At certain intervals during the day, a determined quantity of hot metal is sent by rail to the steel works for the next production step. Inside the melting shop, steel of different quality grades is produced, according to concrete customer orders and requirements. It is cast at continuous casting plants into billets. A single production unit inside the steel works is called heat. A heat is part of a sequence – a total ordered set of similar qualities of equal formats.

Afterwards, these billets are delivered to the rolling mills. Here, steel bars and wire rods of different shapes and formats are produced. In fixed, cyclic rolling campaigns of limited capacities certain formats are produced. These cycles are dependent on the rolling mills, billet supply of the steel works and concrete orders by customers. They vary between one to four weeks. After the rolling, potential following processes concerning steel bars are arrangement, pickling, annealing and saw cutting; wire rods probably need a annealing and a pickling. Finally, the products are delivered to the customers – mostly suppliers of the automotive, shipbuilding or aerospace sectors. Figure 1 depicts the roughly described Supply Chain.

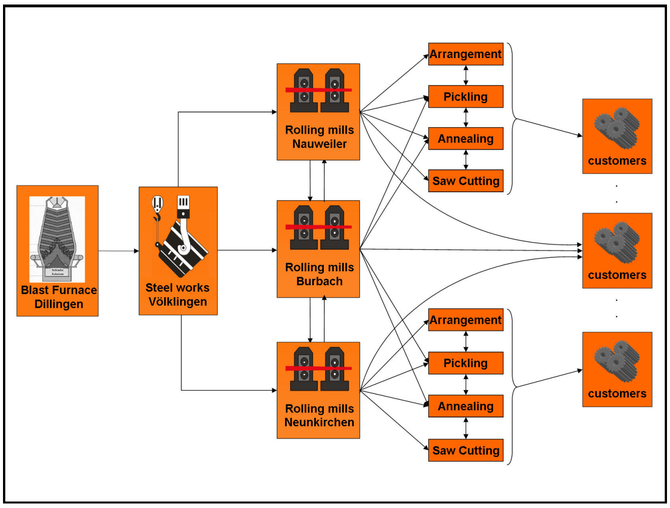
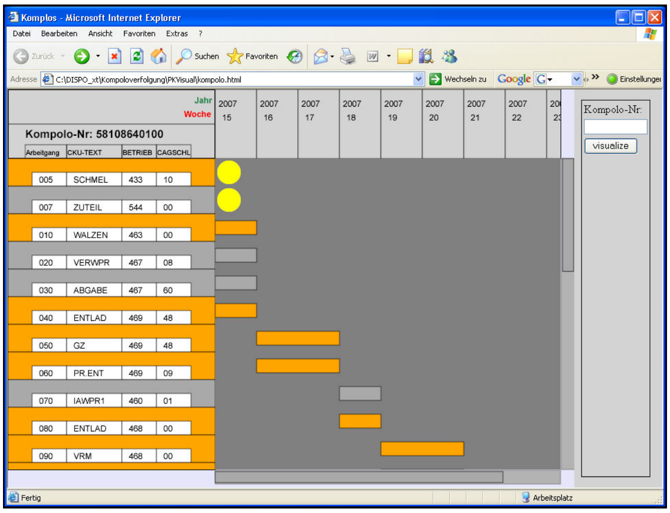


Fig. 1. Supply Chain of Saarstahl AG

Given a working plan, the system schedules the execution of each concrete order along the production chain. It monitors production on a rough—in weeks—and detailed—in days and hours—level, and executes an online detailed planning and scheduling for the different manufacturing phases. It has to detect problems in the production and handle them in order to return to normal production. The rough working plan for each manufacturing phase (shown in Figure 2) is calculated on demand, before final order

978-1-1234-7890-2/08/$20.00 © 2008 IFAC 10.3182/20080706-5-KR-1001.3251

commitment. Depending on delivery date, order size and vertical integration certain capacities at specified aggregates have to be roughly allocated.



Kompolo-Nr: 58108640100

Arbeitgang	CKU-TEXT	BETRIEB	CAGSCHL
005	SCHMEL	433	10
007	ZUTEIL	544	00
010	WALZEN	463	00
020	VERWPR	467	08
030	ABGABE	467	60
040	ENTLAD	469	48
050	GZ	469	48
060	PR.ENT	469	09
070	IAWPR1	460	01
080	ENTLAD	468	00
090	VRM	468	00

Fig. 2. Rough planning according to order position

Usual orders to Saarstahl vary between five to several hundreds of tons. Batch sizes on each manufacturing level are fixed or limited, hence, orders have to be grouped together in process units on each stage with local constraints to keep. For instance, inside the steel work, a production unit is called heat with fixed size of 160t. The orders covered by a heat have to be of same quality, same casting format and should have the same calculated processing step date. Additional restrictions concerning the production inside the steelwork, and how they are handled in MasDISPO – the Multiagent System [Weiss, 1999, Wooldridge, 2002] for the steelworks' optimisation – are described in [Jacobi et al., 2007], [Jacobi et al., 2005].

It is not only the batch size which complicates mapping from one phase to another. Inside the steelworks, orders are also grouped together by steel grades, whereas, on the next phase — rolling — physical dimension is the most relevant criterion; while for annealing treatments, the temperature is the most important one. Thus, every single order has to be mapped into different units on different production phases. Therefore, the groups created by each of these criterion cause production dependencies among the orders and, consecuently, changes in one order's schedule may impact the other's schedules at a given phase.

The average order backlog at Saarstahl is about 17500 orders, which makes it already a complex challenge to find an optimal mapping which keeps all constraints and deadlines. However, the system has to, additionally, deal with the online problem of dealing with new incoming orders and changing requirements by customers.

Normally, as a process step gets closer to a certain phase, the more concrete its allocation and the more detailed its planning has to be. Therefore, this system has to deal with smooth transitions between rough and detailed planning – a challenge which is often only non satisfying matched by traditional centralised approaches [SAP, 2004]. Dependencies between rough and detailed planning, as well as, interconnections between different manufacturing phases have to be modeled.

As presented, the overall process chain is characterized by changes in customer orders and it is affected by production setbacks or problems. Therefore, steel manufacturing companies must be flexible and dynamic, by adapting production plans fast in order to meet customer requirements while still being cost-efficient.

Since these are requirements which need to be covered in almost every industrial sector, there are a lot of commercial systems handling this. But these ERP systems (enterprise resource planning) [Gronau, 2004], [Plossle and Orlicky, 1985] like APO (Advanced Planner and Optimizer) [Bartsch and Bickenbach, 2002] or APS (Advanced Planning and Scheduling) are suitable for a rough planning only, but are frequently not very suitable for operations planning. These existing solutions are dominated by centralized decision making processes, mostly data driven and often not modeling the business processes they should. Big software companies have adopted the strategy to provide integration mechanisms for MES-level solutions [SAP, 2004] like the presented solution.

MasDISPO_xt, a decentralised agent based approach, is the proposed solution of this paper. In MasDISPO_xt, every order is modeled as an agent. The agent calculates and observes its own schedule from order entry, across rough and detailed planning, and monitors the production up to the point of delivery. It responds to changes during planning, scheduling and production by dynamically adapting the schedules. Also, each aggregate of any factory is also modeled as an agent which also calculates its schedule autonomously based on further local knowledge and restrictions.

The complete production chain is very complex and could not be addressed with the appropriate detail in the context of only one paper. Therefore, this paper concentrates on the support of the planning and monitoring of the Annealing process for steel bars and wire rods. The detailed description of the problem and discussion of the solution are presented in Section 2. In section 3, some evaluation results are discussed. Finally , in Section 4, the conclusions and acknowledgments are presented.

2. PLANNING OF ANNEALING FURNACES

In higher planning levels (sales), the global production capacities for the different production phases are booked. After that, the planning process continues by planning at lower levels. In the case of planning for the annealing furnaces, the global planning level provides the lower level with a set of orders that normally cover the complete capacity of all furnaces together. This assigned orders are planned in more detail while going down in planning levels.

Annealing Furnace planning is divided in two levels. First, the assignment of orders to furnaces is planned, allocating the available capacity of each furnace to orders until the capacities are fully allocated or all the orders have been assigned. Secondly, the precise schedule for each furnace is created by allocating orders to specific points in time where they will be processed.

2.1 Furnace assignment

The first level, called *Furnace assignment* is solved using the following model:

Definition 1. Having a finite set of *annealing furnaces* M with elements m_i, where m_i is the furnace number i,

$$M = \{m_1, \ldots, m_n\} \; n \in \mathbb{N}$$

O being the finite set of all *orders* to be planned, with elements identified with the letter o,

$$O = \{o_1, \ldots, o_q\} \; q \in \mathbb{N}$$

L_i being the ordered list of elements of M which are the suitable aggretates for order o_i in order of *preference*,

$$L_i = \{m_a, \ldots, m_b\}$$

where

$$\begin{gathered} \mid L_i \mid \leq n; \; i = 1, \ldots, q; \\ m_a \in L_i \wedge m_b \in L_i \wedge \\ m_a \text{ precedes } m_b \text{ in the list } L_i \Rightarrow \\ m_a \text{ is preferable over } m_b \text{ for order } o_i \end{gathered}$$

L being the collection of preferences for all orders:

$$L = \{L_1, \ldots, L_q\}$$

and the functions C, being the function which associates a furnace to its associated available capacity for the given period, and c, the function which associates each order to its required capacity in a furnace,

$$\begin{gathered} C : M \to \mathbb{N} \\ c : O \to \mathbb{N} \end{gathered}$$

the top level plannning problem for annealing furnaces can be *defined* as the search for a set $\mathcal{P}$ that associates each order in O to furnaces in M following the preferences provided by L and making sure that the sum of all sizes provided by $c(x)$ of the orders associated to a specific furnace do not exceed the furnace's specific maximal capacity $C(m_i)$:

$$\mathcal{P} = O \times M$$

where

$$\forall m_i \in M : \left(\sum_{x \in \{o \mid (o, m_i) \in \mathcal{P}\}} c(x) \right) < C(m_i)$$

A solution $\mathcal{P}$ is produced by looking for each order $o_i \in O$ (sorted by arrival date) a furnace $m_j \in M$ with available capacity, following the list L_i.

It is possible that by following this plan, the amounts of allocations in each furnace gets unevenly distributed, specially because of many orders o_i beeing booked on the same furnace and not in other options in L_i with lower preference. This is specially problematic for orders that mandatorily need to be booked in the specific furnace that has been booked the most. Therefore a redistribution realised of the orders is required. This redistribution is realised by use of *Simulated Trading* [Bachem et al., 1992]. *Simulated Trading* is an improvement mechanism starting from any initial solution—in this context the latest valid allocation $\mathcal{P}$. By successively "selling" and "buying" certain order positions each aggregate tries to optimise its composition of order positions. Objective is to achieve a new assignment of the already accepted order positions to the aggregates with an optimised remaining capacity for each aggregate. The trading goes over several rounds. In each cycle the aggregate agents submit one offer to sell or buy an order position. At the end of each round a trading agent tries to match the sell ond buy offers. This is a special kind of hill-climbing algorithm, which can be interrupted anytime to pick the best solution found.

For this purpose, the fact of modelling aggregates as agents comes in handy, since a solution using *Simulated Trading* is formulated in the following way:

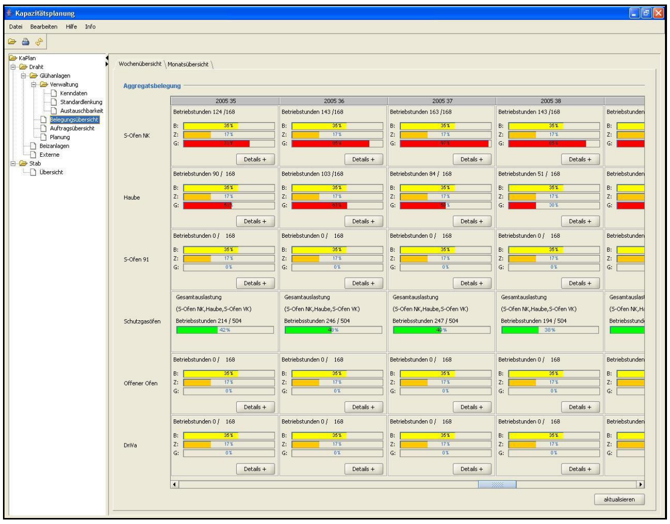
For each furnace $m_i \in M$, there is a set P_i of *planned* orders with allocated capacity at aggregate m_i.

$$P_i = \{o \mid (o, m_i) \in \mathcal{P}\}$$

Each furnace m_i advertises an order $o_j \in P_i$ with allocated capacity at aggregate m_i by auction and accepts orders o_l of other aggregates if and only if o_i also is also accepted by another suitable aggregate $m_k \in L_j$ $(k \neq i)$.

The objective of this is to achieve a new allocation of $\mathcal{P}$ of already accepted orders, which maximizes remaining capacities at all aggregates ballancing the distribution.

Figure 3 shows the result of *Furnace Assignment*: an overview of capacities per week or month for each annealing aggregate. This view shows the stock, the supply and the total quantitiy of each annealing furnace per week. In detailed views, orders with allocated capacity on a specific aggregate for a specific week are presented.



	2005 35	2005 36	2005 37	2005 38	
S-Ofen NK	Betriebstunden 124 /168 B: 35 % Z: 17 % G: 73 %	Betriebstunden 143 /168 B: 35 % Z: 17 % G: 85 %	Betriebstunden 163 /168 B: 35 % Z: 17 % G: 97 %	Betriebstunden 143 /168 B: 35 % Z: 17 % G: 85 %	Betriebstunden B: Z: G:
Haube	Betriebstunden 90 / 168 B: 35 % Z: 17 % G: 53 %	Betriebstunden 103 /168 B: 35 % Z: 17 % G: 61 %	Betriebstunden 84 / 168 B: 35 % Z: 17 % G: 50 %	Betriebstunden 51 / 168 B: 35 % Z: 17 % G: 30 %	Betriebstunden B: Z: G:
S-Ofen 91	Betriebstunden 0 / 168 B: 35 % Z: 17 % G: 0 %	Betriebstunden 0 / 168 B: 35 % Z: 17 % G: 0 %	Betriebstunden 0 / 168 B: 35 % Z: 17 % G: 0 %	Betriebstunden 0 / 168 B: 35 % Z: 17 % G: 0 %	Betriebstunden B: Z: G:
Schutzgasöfen	Gesamtauslastung (S-Ofen NK,Haube,S-Ofen VK) Betriebsstunden 214 / 504 42 %	Gesamtauslastung (S-Ofen NK,Haube,S-Ofen VK) Betriebsstunden 246 / 504 48 %	Gesamtauslastung (S-Ofen NK,Haube,S-Ofen VK) Betriebsstunden 247 / 504 49 %	Gesamtauslastung (S-Ofen NK,Haube,S-Ofen VK) Betriebsstunden 194 / 504 38 %	Gesamtauslas (S-Ofen NK,H Betriebsstund
Offener Ofen	Betriebstunden 0 / 168 B: 35 % Z: 17 % G: 0 %	Betriebstunden 0 / 168 B: 35 % Z: 17 % G: 0 %	Betriebstunden 0 / 168 B: 35 % Z: 17 % G: 0 %	Betriebstunden 0 / 168 B: 35 % Z: 17 % G: 0 %	Betriebstunden B: Z: G:
DriVa	Betriebstunden 0 / 168 B: 35 % Z: 17 % G: 0 %	Betriebstunden 0 / 168 B: 35 % Z: 17 % G: 0 %	Betriebstunden 0 / 168 B: 35 % Z: 17 % G: 0 %	Betriebstunden 0 / 168 B: 35 % Z: 17 % G: 0 %	Betriebstunden B: Z: G:

Fig. 3. Furnace Assignment Visualisation

2.2 Detailed operations planning

The deepest planning level produces concrete plans where each working step in an aggregate is specified. To help explain how this is done for furnaces, the case of the batch-type annealing furnace will be explained, as this is the most restrictive furnace available. It consists of three sockets S_1, S_2, S_3, which are simply the devices that hold the material during the process, two glowing hoods H_1, H_2 and a cooling hood K, which perform the actual process.

Each aggregate have minimal—16 hours—and maximal working periods. The orders assigned for each aggregate have to be sorted in working *programs* which are groups of orders that do not exceed this maximal period and that have the same metalurgical properties, like temperature, glowing time and cooling time requirements, and therefore will be submitted to the same process in the aggregate. Each of these subsets of P_i will be identified with the letter b:

Definition 2. Let P_i be the assigned stock of orders to be processed by aggregate m_i:

$$P_i = \{p_a, p_b, \ldots, p_c\}$$

The set $\mathcal{B}$ is *defined* as the minimal ($|\mathcal{B}|$ is held to a minimum) collection of subsets b_j of P_i, where each subset is a collection of orders p_x that share the same processing properties and together do not exceed the maximal period of uniterrupted work for the given kind of process. These subsets b_j constitute a partition of P_i:

$$\begin{gathered}\mathcal{B} = \{b_1, \ldots, b_n\}, \text{ where} \\ b_j \subseteq P_i \ \forall j = 1, \ldots, n \\ \bigcup_{j=1}^{j=n} b_j = P_i\end{gathered}$$

Each program b_j starts with a preparation of the material on one socket, it is followed by a heating and glowing phase, which will be observed as a single step that requires one of the glowing hoods, H_1 or H_2, and is finished with a cooling phase using the cooling hood K.

The objective is to process the complete stock P_i as early as possible, optimizing throughput. In other words, it is necessary to produce schedules for each of the devices, the glowing and cooling hoods, that attend the complete set $\mathcal{B}$ minimizing time gaps between programs:

Definition 3. Having a set $T \subseteq \mathbb{N}$ as the set of ordered time values, the model of the solution for the detailed planning for batch-type annealing furnaces is defined a set of associations $\mathcal{D}$ of programs to their starting time and the selected glowing hood, called *detailed plan*:

$$\mathcal{D} = \mathcal{B} \times T \times \{H_1, H_2\}$$

Definition 4. For each program, the following functions provide the next information:

- $r : \mathcal{B} \to \mathbb{N}$ provides the *preparation* duration.
- $g : \mathcal{B} \to \mathbb{N}$ provides addition of preparation and *glowing* duration
- $k : \mathcal{B} \to \mathbb{N}$ provides the addition of preparation, glowing and *cooling* duration.
- $f : \mathcal{B} \to T$ provides the earliest deadline of an order p inside the program provided as argument.

Also, the function o is defined as the evaluation whether two intervals overlap:

- $o : \{\mathbb{N} \times \mathbb{N}\} \times \{\mathbb{N} \times \mathbb{N}\} \to \{y, n\}$

This solution must fulfill the following constraints:

$\forall (b_i, t, h) \in \mathcal{D}$:

(1) $\neg\exists(b_j, t', h')$
$s.t\ j \neq i \wedge$
$o((t + r(b_i), t + g(b_i)), (t' + r(b_j), t' + g(b_j))) = y \wedge$
$h = h'$:
No other program can use the same glowing hood at the same time.

(2) $\neg\exists(b_j, t', h')$
$s.t\ j \neq i \wedge$
$o((t + g(b_i), t + k(b_i)), (t' + g(b_j), t' + k(b_j))) = y$:
No other program can use the cooling hood at the same time.

The first solution is constructed by scheduling programs one by one prioritizing programs with less slack time over others with more slacktime, which is calculated using the following function:

Definition 5. Slack time function s is defined as the calculation of how much time will remain between the end time of the production $t + k(b)$ of a program b and its deadline $f(b)$ if it is planned to start at time t:

$$s : \mathcal{B} \times T \to \mathbb{N}$$

The first program b_i chosen is the one that has the less slacktime if planned at the absolute start time t_0:

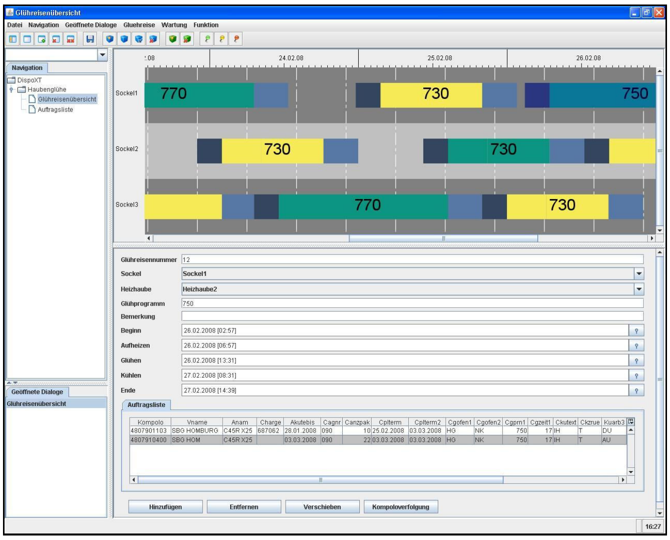
$$b_i | s(b_i, t_0) = min(s(b_x, t_0)) \ \forall b_x \in \mathcal{B}$$

This decision is repeated with the programs in $\mathcal{B}$ that have not been scheduled, picking as start time at each step, the minimum between the end times of the last allocations on each of the glowing hoods, minus the preparation time of the program being evaluated. Doing so constructs the solution without breaking the first constraint.

The evaluation of each program has to take into account that it does not break the second constraint: overlap in the cooling hood plan. If such a collision is detected for the program being evaluated, the start time used has to be adjusted shifting it forward until the collision is avoided. This shifted start time is to be used in the function $s(t, b)$ for comparing the program.

The first solution is calculated once all programs in $\mathcal{B}$ have been scheduled. This first solution is characterized by having maximum stability. It can be optimized further, if the slack times introduced to avoid breaking the second constraint are reduced. Therefore, search is continued tracking back to the points where slack time was introduced and making there decisions different than the ones initially taken. This search can be bound using heuristics, like pruning branches in the search tree that have accumulated more slack time than the best solution found by then, since it is evident that it will not provide any improvement. This search is produced exhaustively.

Figure 4 shows the result of detailed planning for batch-type furnaces as Gantt-Chart [Clark, 1952]. Its vertical axis is divided in three sockets $\mathcal{S}_1$, $\mathcal{S}_2$ and $\mathcal{S}_3$ and its horizontal axis corresponds to time.



Glühreisennummer 12
Sockel Sockel1
Heizhaube Heizhaube2
Glühprogramm 750
Bemerkung
Beginn 26.02.2008 [02:57]
Aufheizen 26.02.2008 [06:57]
Glühen 26.02.2008 [13:31]
Kühlen 27.02.2008 [08:31]
Ende 27.02.2008 [14:39]

Auftragsliste

Kompolo	Vname	Anam	Charge	Akutebis	Cagnr	Canzpak	Cplterm	Cplterm2	Cgofen1	Cgofen2	Cgprn1	Cgzeit1	Ckutext	Ckzrue	Kuarb3
4807901103	SBG HOMBURG	C45R X25	687062	28.01.2008	090	10	25.02.2008	03.03.2008	HG	NK	750	17	IH	T	DU
4807910400	SBG HOM	C45R X25		03.03.2008	090	22	03.03.2008	03.03.2008	HG	NK	750	17	IH	T	AU

Fig. 4. Batch-type Annealing Furnace Plan

2.3 Monitoring of production

As already mentioned in Section 1, this planning system has been designed as a multiagent system. For monitoring the progress in the production, each order and each aggregate is implemented as an agent that tracks its own progress and raising alarm as soon as it detects problems that can arise. This is described in detail in [Jacobi et al., 2007] and [Jacobi et al., 2005].

This same principle has been implemented for glowing programs in $\mathcal{B}$. Signals from the different devices are delivered to the system and forwarded to the corresponding programs. These evaluate the situation as it is informed and take actions in case it is necessary, like showing alarm signals in the monitoring view, like the one shown in Figure 4. Other actions can be to alarm *order agents* that can be affected. This way, consequences for other phases in the production chain of the factory can be recognized as soon as possible.

3. EVALUATION

The solution $\mathcal{P}$ as described covering the furnace assignment guarantees an even distribution of all orders $o_i \in O$. This eases the second phase, the detailed operations planning as described in section 2.2.

The detailed planning provides the plant operator with optimal solutions regarding the next 10 following programs. All programs $b_j \in \mathcal{B}$ have been scheduled in the first solution. After that, branch-and-bound is used for the first 10 scheduled programs to search for an optimal solution. Since one program has a duration of at least 16 hours and the 3 sockets are used in parallel, the next 2 days are planned optimally. $\mathcal{B}$ continuously changes and the batch-type annealing plan is created daily or on demand if necessary.

The former approach was a complete manually created plan based on $\mathcal{B}$. The first task – holding $|\mathcal{B}|$ to a minimum – was only parenthetically mentioned in the last section. This is already an instance of the *Bin Packing Problem*. The complexities of the problems faced are hard to handle manually. These are covered by the system, now. More than one hour of working time was needed to create a plan manually. Now, in a few seconds the plant operator receives an optimal solution. He is able to change any production relevant aspect and let the system calculate the consequences. This functionality is new since it was just too extensive to create several plans manually in order to compare them. Finally, the operator commits a final version of the plan to production. Hence, the deployed application marks a great surplus for the plant operator and the company since throughput is optimized.

4. CONCLUSIONS AND ACKNOWLEDGEMENTS

The described examples of this paper state a subset of the different problems which need to be solved along production inside the supply chain. But these cases already prove that the problems are too complex to be handled manually. An automatic and responsive planning system is needed. The decentralised approach with multiagent systems make the system easier to handle, really models the demanded business processes and is able to manage the huge data amount along production – requests which are not always met by the existing centralised approaches.

The authors like to thank Saarstahl AG. Without their innovation related mind the realisation of MasDISPO_xt would not have been possible.

REFERENCES

A. Bachem, W. Hochstaettler, and M. Malich. Simulated trading: A new approach for solving vehicle routing problems. Technical Report Tech.Rep. 92.125, Mathematisches Institut der Universitaet zu Koeln, 1992.

H. Bartsch and P. Bickenbach. *Supply Chain Management mit SAP APO*. SAP Press, 2nd ed., 2002.

W. Clark. *The Gantt Chart*. Pitman and Sons, London, 3rd edition, 1952.

N. Gronau. *Enterprise Resource Planning und Supply Chain Management: Architektur und Funktionen*. Oldenbourg (Muenchen), 2004.

S. Jacobi, E. Leon-Soto, C. Madrigal-Mora, and K. Fischer. Agentsteel: An agent-based online system for the planning and observation of steel production. *Proc. 4th Inter. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 05)*, 2005.

S. Jacobi, E. Leon-Soto, C. Madrigal-Mora, and K. Fischer. Masdispo: A multiagent decision support system for steel production and control. *AAAI Innovative Applications of Artificial Intelligence*, 2007.

G. W. Plossle and J. A. Orlicky. *Orlicky's Material Requirements Planning*. Prentice Hall, 2nd ed., 1985.

SAP. Integration von mes-systemen in sap for mill products, 2004.

G. Weiss, editor. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. KIT Press, 1999.

M. Wooldridge. *An Introduction to Multiagent Systems*. John Whiley & Sons, 2002.