Development of a knowledge model for managing schedule disturbance in steel-making

R. Roy , B. A. Adesola & S. Thornton

To cite this article: R. Roy , B. A. Adesola & S. Thornton (2004) Development of a knowledge model for managing schedule disturbance in steel-making, International Journal of Production Research, 42:18, 3975-3994, DOI: 10.1080/00207540410001716453

To link to this article: https://doi.org/10.1080/00207540410001716453

Published online: 21 Feb 2007.

INT. J. PROD. RES., 15 SEPTEMBER 2004,
VOL. 42, NO. 18, 3975–3994

Taylor & Francis
Taylor & Francis Group

Development of a knowledge model for managing schedule disturbance in steel-making

R. ROY†*, B. A. ADESOLA† and S. THORNTON‡

The development of a knowledge model, which describes the reasoning process in managing schedule disturbance in steel-making, is presented. A literature review shows the lack of research in developing a knowledge model for decision-making in steel-making. The knowledge model distinguishes three knowledge categories: task, inference and domain. Knowledge is captured for the ten most common types of disturbances in steel-making. A common inference model exists for disturbance management. A knowledge elicitation methodology called eXpert Process Knowledge Analysis Technique (XPat) combined with a CommonKADS approach was used to capture process knowledge for managing schedule disturbance in steel-making. Finally, the knowledge model is validated through paper-based simulations of three common disturbance scenarios. The validation process consisted of three components: accuracy, completeness and consistency.

1. Introduction

In practice, the scheduling systems used to assign activities to resources often assume the generated schedule will remain workable for the foreseeable future. The process of manually constructing the predictive schedule for steel-making of a 12 h shift by a human scheduler takes at least 2 h. This manual scheduling time means that it is difficult to react to unforeseen production events, e.g. a rushed order, especially during night shifts and weekends when experts might not be available (Cowling and Rezig 2000). In addition, when it is necessary in advance to schedule several parallel activities that share resources, the quality of manually generated schedules deteriorates with time due to unplanned events. This can cause disturbances and disruption to plans requiring modification actions or even rescheduling (Brown 1988). Frequent rescheduling often results in instability and a lack of continuity in detailed schedule execution. Due to the dynamic nature of the steel-making process, however, it is often difficult to maintain the original short-term schedule (Numao and Morishita 1991). The schedule disturbance management is a manual process and requires many years of experience. The research presented in this paper intends to formalize the knowledge required to manually modify a schedule in order to minimize the impact of any disturbance. The knowledge can then be

Revision received March 2004.

†Department of Enterprise Integration, School of Industrial and Manufacturing Science, Cranfield University, Bedford MK43 0AL, UK.
‡Corus UK Ltd, Teesside Technology Centre, PO Box 11, Grangetown, Middlesborough, Cleveland TS6 6UB, UK.
*To whom correspondence should be addressed. e-mail: r.roy@cranfield.ac.uk

used in a decision support system to improve the management of schedule disturbance and avoid unnecessary rescheduling.

The paper presents a knowledge model for decision support to manage schedule disturbance in steel-making (hypothesis). Knowledge modelling is an approach to develop a knowledge-based system (Schreiber *et al.* 1999; MOKA 2001). This is a transformation approach to knowledge capture by modelling 'what an expert does'. A knowledge model for the MSD is a semiformal representation of the tasks involved, the inference mechanism to manage the disturbance and any domain-specific knowledge. Section 2 presents an overview of managing schedule disturbance in steel-making. Section 3 describes the approach for knowledge model development. Section 4 presents validation of the knowledge model through case studies. Section 5 presents a discussion on the research methodology and results. Section 6 concludes with the limitation of the approach and the future research.

2. Managing schedule disturbance (MSD) in steel-making

Scheduling in general is a dynamic activity where several repair actions may be required depending on internal or external influences. MSD is a complex knowledge-intensive activity performed by human experts. It encompasses several ranges of dynamic tasks such as generating alternative actions and making decisions. MSD is necessary to ensure a reaction in one domain does not affect the rest of the schedule. Disturbance in steel-making can be caused by a variety of unexpected events ranging from external influence to internal constraints. The word 'disturbance' has been used in this context to mean an interruption due to unexpected disruption in the steel-making process. This suggests disturbance is an incident in which the state of normal behaviour is upset. For example, an external influence may be rushed orders and an internal constraint may be machine or tool breakdown, or rework due to wrong product specifications. MSD in steel-making is a problem-solving process whereby specific problem-solving knowledge (PSK) is specified in order to generate a set of instructions as possible actions. The generation of instruction set depends on the *time available* and the state of the overall steel-making schedule.

The task of MSD is popularly termed 'reactive scheduling'. The human schedulers, as experts, solve problems by inferring knowledge from experience and communicating instructions about the schedule either by word of mouth or via a Gantt chart. The use of a Gantt chart in general scheduling is widespread. It is a formal tool for communicating change in the schedule. In practice, there is more to reactive scheduling than updating the Gantt chart. Informal communication is common between schedulers and shop floor operatives. Note that no one has addressed the issue of knowledge capture to support manual MSD. It is important to understand 'what the experts do to solve reactive scheduling problems' and how to represent the heuristics employed during this process. The major aim of this research is to prove that a generic model for MSD in steel-making can be developed. The overall argument is that MSD in steel-making is a complex knowledge-intensive activity that should be aided by a decision support mechanism to address any process constraints. According to Dorn and Shams (1991), compatibility constraints of higher grades of steel impose requirements on the sequence in which orders are produced. To achieve certain characteristics, when chemicals are added it may also react with the steel-making aggregate.

3. Developing a knowledge model for MSD in steel-making

The ability of human schedulers to react to unexpected events or disturbances is identified as their capacity to reason about the predictive schedule and possible actions to minimize disruption on the shop floor. Reasoning about possible actions requires understanding of processes and knowledge from past experience. This section focuses on the development of a knowledge model for MSD in steel-making.

3.1. *The approach*

In steel-making scheduling, each category of disturbance is handled differently. It is essential that each category of disturbance and possible action to modify or reschedule is well understood. To model the process of MSD in steel-making scheduling, ten most common categories of disturbances were identified: steel out-of-specification in a BOS plant, steel out-of-specification in an SSM plant, steel temperature too hot, steel temperature too cold, hot metal supply, tap needs outlet, heat needs outlet, clash on SSM equipment, clash on concast equipment, and ladle gate failure. The categories of MSD are identified using a series of semistructured interviews with experts from three different plants within Corus. The authors also studied shift logs for any disturbance and analysed previous company documentation. PSK from these categories was captured using the XPat methodology (Adesola *et al.* 2001). XPat knowledge elicitation methodology is easy to use by the experts and is suitable for process knowledge capture. This is followed by the development of knowledge items contained in the PSK. Adesola (2002) reviewed seven knowledge modelling frameworks to evaluate their suitability to support different stages of knowledge capture and reuse. CommonKADS methodology in Schreiber *et al.* (1999) emerged as the most effective in terms of explicit realization of evaluation criteria. Therefore, CommonKADS methodology is followed to analyse the XPat interview results and develop the knowledge model. Validation of the model is performed using paper-based simulations of three case studies for accuracy, completeness and consistency.

The process of developing PSK for MSD consists of seven steps. The first two identify and capture domain-specific knowledge and their sources. Steps 3 and 4 describe direct knowledge elicitation techniques used to collect, interpret and transform problem-solving processes. Direct knowledge elicitation techniques such as interviewing and protocol analysis have been used in steps 1–4.

Step 1. Review existing documentation.
Step 2. Generate scenarios.
Step 3. Interview experts using a questionnaire based on XPat.
Step 4. Transcribe and interpret.
Step 5. Determine task type.
Step 6. Validate and repeat steps 3–5.
Step 7. Document PSK.

The protocol knowledge acquired is analysed to produce output in the form of rules and procedures. These represent the domain expertise that people bring to bear in the decision process. Once the knowledge has been identified and elicited, the next step is to classify the nature of the task. To elicit knowledge, simple structured questions were developed based on the XPat approach. The questions were intended to allow knowledge engineers to draw out from expert schedulers how they reason during problem-solving and what information sources are used and/or reused, the people involved and the nature of interactions. Four experts were interviewed; in addition,

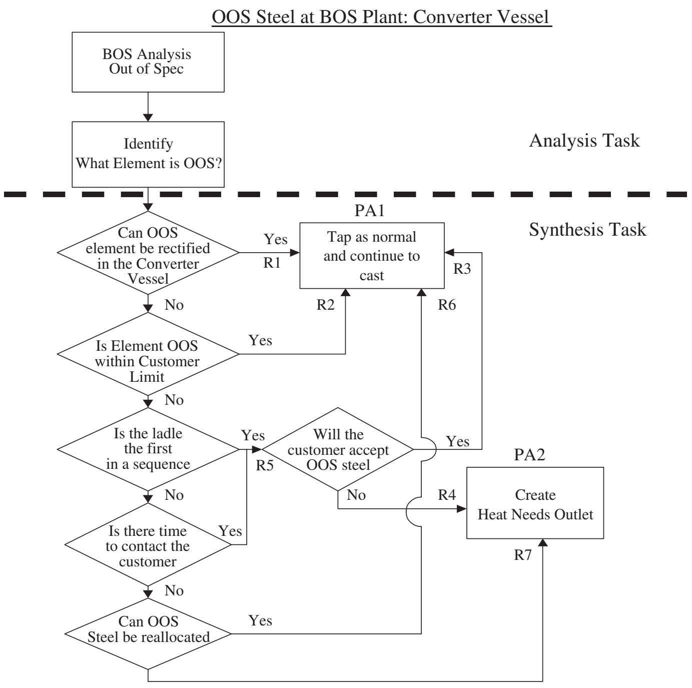
OOS Steel at BOS Plant: Converter Vessel
BOS Analysis Out of Spec
Identify What Element is OOS?
Analysis Task
Synthesis Task
Can OOS element be rectified in the Converter Vessel
Yes
R1
PA1
Tap as normal and continue to cast
R3
No
R2
R6
Is Element OOS within Customer Limit
Yes
No
Is the ladle the first in a sequence
Yes
R5
Will the customer accept OOS steel
Yes
No
R4
PA2
Create Heat Needs Outlet
No
Is there time to contact the customer
Yes
R7
No
Can OOS Steel be reallocated
Yes

Figure 1. Problem-solving (procedural) flowchart for out-of-specification (OOS) steel in the BOS vessel.

the authors observed experts during MSD. Figure 1 presents a problem-solving (procedural) flow chart for steel out-of-specification in the basic oxygen steel-making (BOS) vessel.

Different categories of disturbance require different PSK. A PSK in this case consists of 11 knowledge items:

(1) Problem description: brief description of the problem in terms of the nature of the problem and its location.
(2) Entities involved: list of the steel-making equipment and systems involved in the disturbance.
(3) People involved: list of people consulted by the shift scheduler about possible actions.
(4) Relevant knowledge: refers to both tacit and explicit knowledge relevant to address the disturbance.
(5) Problem recognition: indication of who informs the shift scheduler; sources may include people and systems.
(6) Consequences: measure of the effect on business and scheduling overall.
(7) Possible actions: refers to repair actions linked to the reasoning process.
(8) Considerations: describe the possible implication on cost and performance.
(9) Implementation: procedural flowchart illustrating the flow of reasoning.
(10) Glossary of terms: list of common terminology used during problem solving.
(11) Data used: required data and information to generate actions.

Item nine represents procedural or problem-solving flowcharts, which illustrate the flow of reasoning. Together, all these represent a structured format for eliciting knowledge about different categories of disturbance.

3.2. *From PSKs to a knowledge model*

CommonKADS, the leading methodology, influenced the development of a knowledge model in this research. It distinguishes three knowledge categories: task knowledge, inference knowledge and domain knowledge. The task knowledge defines control over the inferences, the inference knowledge describes basic inference steps performed using domain knowledge and the domain knowledge specifies knowledge and information types in an application. The knowledge model development starts by selecting a template knowledge model (TKM) from the CommonKADS library of templates. Template selection is itself a knowledge-intensive activity, because it requires understanding of the domain and the goals that the task intends to achieve.

From the initial study, it is observed that the task of MSD is a combination of analytical and synthetic tasks. The nearest inference structure in this case is the configuration design task template in Schreiber *et al.* (1999). The configuration design method uses a variation of the propose–critique–modify class of method described by Chandrasekaran and Johnson (1993). The 'propose' part of the method is similar to the predictive schedule; the 'critique–modify' does not exactly fit MSD but some of the features are similar to construct and repair a schedule. Since existing task templates are not adequate for MSD, this template presents a useful starting point to adapt and construct a TKM for the application task at hand.

3.2.1. *Mapping XPat to CommonKADS: knowledge specification*

The flowchart in figure 1 features two major classes of problem-solving tasks: analytical and synthetic (Breuker *et al.* 1987, Tansley and Hayball 1993, Schreiber *et al.* 1999). It captures typical analytical and synthetic features of the PSK. For each disturbance category, the flowchart method was used to collect procedures used for problem solving. The flowchart indicates how steel-making schedulers reason during the problem-solving process. The interpretations of these flowcharts provide the necessary understanding for the role of knowledge and the inferences made.

Figure 2 shows the two routes prescribed in CommonKADS methodology to map a reasoning process onto the knowledge model specification. 'Middle-out' requires parallel activities involving decomposition of tasks through the application of methods whilst refining the domain knowledge at the same time. The inference structure represents inference functions with the 'ellipse-shape' and knowledge roles with the 'rectangle-shape'. The task knowledge and domain knowledge are mapped to the inference structure via the inference functions and knowledge roles, respectively. The approach taken in this research is the middle-out route. The decision to start construction of the inference structure by the middle-out route was influenced by the flowchart method of collecting PSK.

3.2.2. *Inference knowledge specification*

Note that existing CommonKADS TKM are not adequate for the problem of MSD in steel-making, hence it is necessary to adapt the existing template and construct an inference structure for MSD. The decision to adapt and construct the TKM via the middle-out route was made above based on data available about the problem-solving process. This section discusses the evolution of the inference structure for MSD.

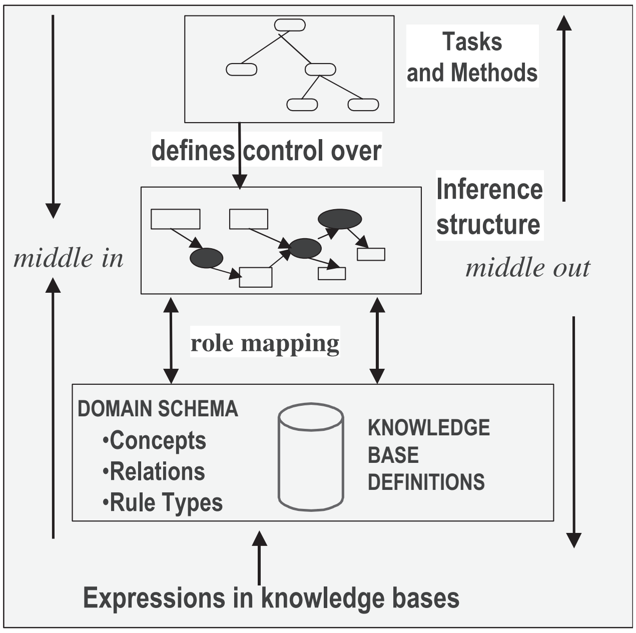

Figure 2. Knowledge model specification (Schreiber *et al.* 1999).

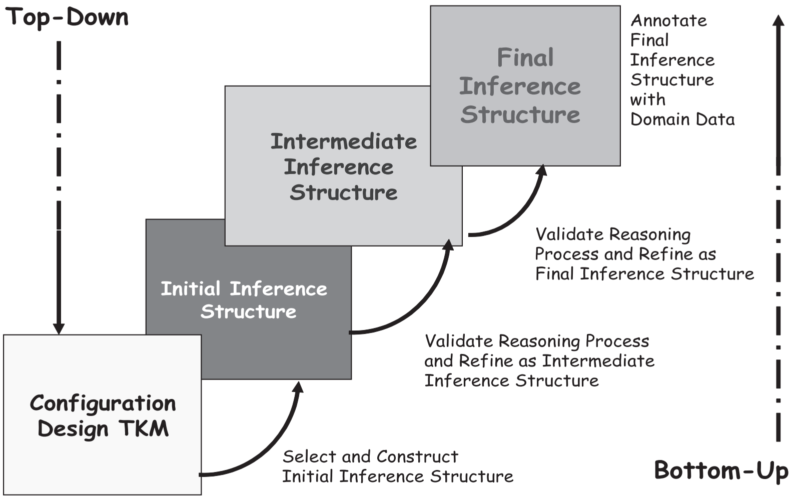
Top-Down
Configuration Design TKM
Initial Inference Structure
Intermediate Inference Structure
Final Inference Structure
Annotate Final Inference Structure with Domain Data
Validate Reasoning Process and Refine as Final Inference Structure
Validate Reasoning Process and Refine as Intermediate Inference Structure
Select and Construct Initial Inference Structure
Bottom-Up

Figure 3. Evolution of inference structure for MSD in steel-making.

3.2.2.1. *Evolution of inference structure for MSD in steel-making*. Figure 3 illustrates the evolution of the inference structure. A bottom-up approach to model-based knowledge acquisition has been applied to construct an inference structure. The inference structure is based on the available 'configuration design' task template that fits only part of the reasoning pattern of the knowledge-intensive task identified during knowledge elicitation (Adesola *et al.* 2002). The authors studied all ten categories of disturbances (corresponding to ten categories of PSKs) identified during the research. Note that a common pattern exists between the PSKs, which suggests steel-making scheduling experts reason about the problem solving in a similar way. To achieve a generic inference structure, the steps in problem solving were identified to abstract patterns of behaviour. The steps were then put together to form an inference structure through an iterative approach that involves communication with the experts and refinement at each stage to update the reasoning process. There were two iterations in the evolution of the inference structure.

Changes made to the initial and intermediate inference structure include identifying the input/output for the newly discovered inference functions in order to extend the reasoning process. The first task was to identify and evaluate model mismatches in the intermediate inference structure and delete them. With further iteration, the intermediate inference structure was annotated with domain data and further refinement was carried out to produce the final inference structure.

3.2.2.2. *Final inference structure.* The final inference structure in figure 4 demonstrates a combination of task types. Monitoring is an analytical task to establish the behaviour of a system (Breuker *et al.* 1987). The task involves selecting a system parameter that can reveal new findings. A norm value is then specified and compared with the new findings. If there is any difference, it is usually classified as a minor or a major discrepancy representing the analysis part of the task. The task of modifying a schedule is synthetic by nature. The goal of a synthetic task is to find a structural description of a system in terms of some given set of elements, formalism or partial structures. To identify a specification, a synthetic task may initially contain an analytical task (Breuker *et al.* 1987).

Construction of the inference structure is realized by identifying first the inference functions for the analysis task and then synthetic tasks. For the analytical task, the three inferences identified were 'monitor', 'verify' and 'specify'. For the synthetic

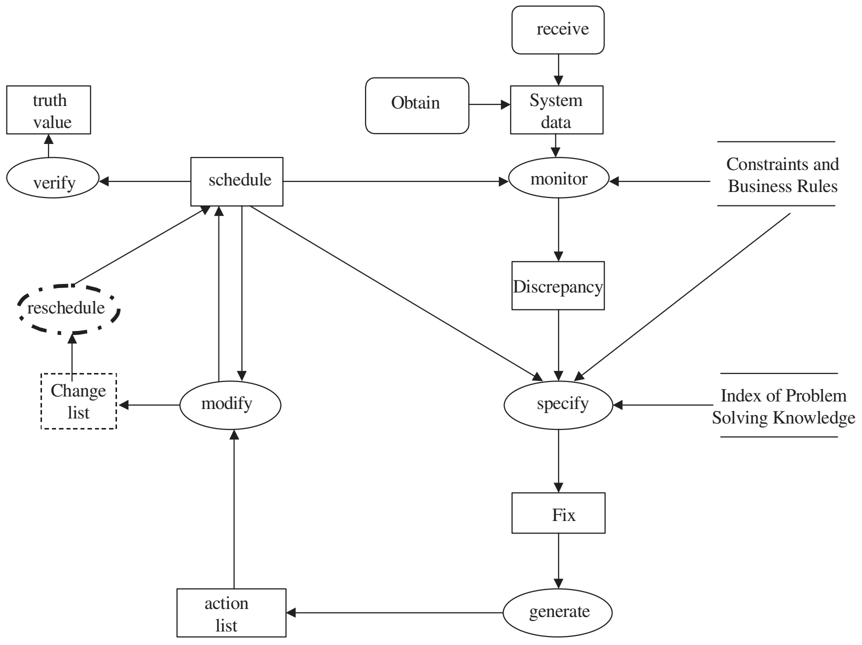

Figure 4. Inference structure for MSD in steel-making.

task, the three inferences identified were 'generate', 'modify' and 'reschedule'. The flowchart method abstracts the analytical task of monitoring and classifying system behaviour. It provides a top-level view of the problem-solving method.

The human scheduler receives system data (unsolicited message), analyses it and can obtain additional data if wanted. The 'system data' is a message informing the user that there has been a violation. The violation is of a type represented as 'constraints' or 'business rule'. In this example, the 'monitor' inference checks the chemical analysis against the specifications of each element for 'constraint violation'. The 'verify' inference confirms the status of the schedule via a 'truth' value. If a constraint is violated, it returns 'True' else 'False'. Depending on the categories of disturbance, possible actions may range from 'do nothing' to local modification action, or in the last resort reschedule of the whole program. If the truth-value returned is 'True', a 'discrepancy' class is identified. This will then point to 'specify' inference to find an appropriate 'fix'. The process of generating actions to 'modify' the heat sequence depends on the state of the schedule and implements the action list, and in the worst case produces a message to ask for rescheduling.

3.2.3. *Task knowledge specification*

In the previous section, a middle-out approach was followed to construct an inference structure for MSD in steel-making. This section describes the process of specifying task knowledge, the control structure and general characteristic of the task manage–schedule–disturbance. Figure 5 shows the task structure for MSD. The task manage–schedule–disturbance was identified using XPat. The task structure was developed by following five steps:

Step 1. Define the top-level task. This is the goal the scheduler intends to achieve.
Step 2. Identify task method to realize the top-level task.
Step 3. Decompose the task method into subtasks.
Step 4. Decompose subtasks into subtask methods.
Step 5. Decompose subtask methods into primitive tasks, i.e. leaf functions.

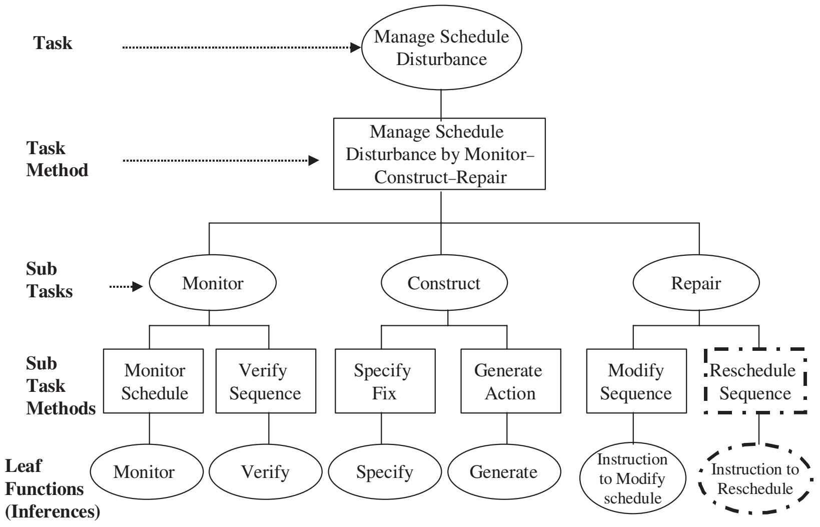

Figure 5. Task structure for MSD in steel-making.

The link between task and inference structure is shown as the leaf function. The structure is a decomposition of top-level composite task. The task knowledge category describes the goals and the strategies that will be employed to realize the goals. In this case, the task is to manage–schedule–disturbance to maintain sequence of heats to minimize the risk of needing to reschedule orders. The task knowledge is described in a hierarchical fashion. The top-level task manage–schedule–disturbance is decomposed into smaller tasks, which in turn are split into even smaller tasks. (A composite task is a problem-solving action. It specifies an abstraction level and requires decomposition before it can be executed.) The task does not include rescheduling task. It only decides whether rescheduling is necessary, therefore rescheduling is shown as a dashed dotted line in figure 5. The lowest-level tasks are the leaf functions and are linked to inferences and transfer functions in the inference structure. They are primitive tasks as they have enough knowledge available that they can be solved without further decomposition.

The task method defines a reasoning goal, in other words, how the task manage–schedule–disturbance can be realized through decomposition into subfunctions. In this case, the top-level task is decomposed by a task method monitor–construct–repair. The subtask 'monitor' describes the activities that track schedule execution and verify the state of the original schedule. This involves looking up the Gantt chart and other information sources for systems data. The subtask 'construct' specifies PSK and generates possible actions for a given problem. This involves negotiating with shop floor operators, suppliers (blast furnace for hot metal) and the customers (mill schedulers) to find a solution to maintain stability of the steel-making process. For example, in the event of out-of-specification steel in a BOS vessel, the scheduler uses his/her expertise to generate possible actions by analysing requirements and trade-offs to balance the objectives of the plant. The subtask 'repair' presents possible modification actions to remove constraint violation or generate a change list for rescheduling action where it is necessary. The latter action is the last resort.

The subtasks are decomposed into subtask methods. For example, the subtask 'monitor' is decomposed into 'monitor-schedule' and 'verify-sequence'. In real life, these two functions are performed iteratively by the scheduler during the problem-solving process. To respond to system data that indicate a discrepancy, the monitor function tracks system data (feedback from various sources including unsolicited messages received) from the shop floor operators. For verification, the scheduler checks specification for constraint violation. For example, if the temperature is too hot in a ladle such that it is not possible to send the steel to a caster, the verify subtask will return false, a Boolean value. This result of verification will be passed onto a specify subtask which will call the appropriate PSK from the index of PSK.

The index of PSK is where procedures are stored for the different categories of disturbance such as out-of-specification, temperature too hot, ladle gate failure, etc. The leaf functions describe the lowest level of reasoning in the inference structure (Schreiber *et al.* 1999). For the purpose of problem solving, the 'construct' method uses 'specify' and 'generate' functions as procedures for handling specific problems. To present actions to address a specific problem, the subtask 'modify' is employed and the 'reschedule' function is used when it is not feasible to 'modify' a schedule.

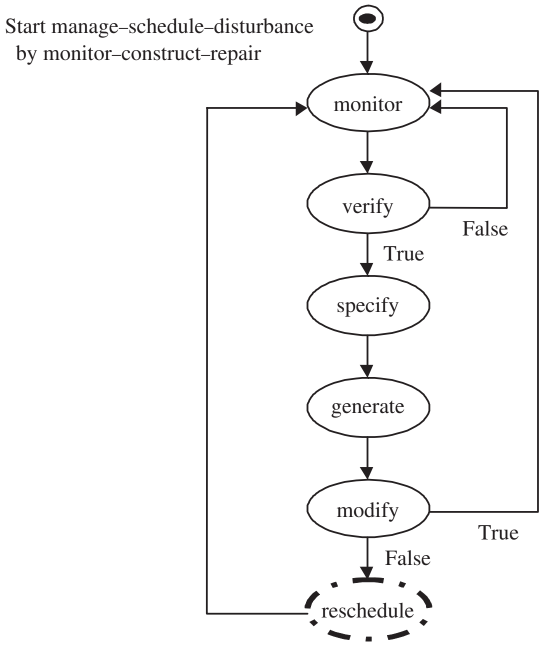

Figure 6. Control structure for monitor–construct–repair.

General characterization of managing–schedule–disturbance task

Goal: Given a set of units and resources assigned to
a schedule, monitor execution, find constraint violation
and apply a fix-action to satisfy constraint.

Typical Example: Disturbance management in steel-making.

Terminology:

System data: data that initiate disturbance in a process
Discrepancy: abnormal behaviour in schedule execution
Fix: an ordered list of possible actions to remove or at least minimize discrepancy
Constraint: a control or something that limits process behaviour.

Input: Complaints about disturbance affecting a schedule.

Output: A set of instructions to minimize the impact of schedule
disturbance in steel-making.

Features:
Managing schedule disturbance is traditionally a manual activity and
in principle lacking any structure. The activity straddles both analytic
and synthetic task and demands balanced attention.

Figure 7. Default method for MSD in steel-making.

3.2.3.1. *Control structure.* Figure 6 shows the method control structure for the monitor–construct–repair task. It has a graphical view of the method of control. The control structure assumes there is an observation that can verify the existence of a discrepancy and then the steps are followed. Figure 7 shows a generic default method applicable to MSD in steel-making. The method is data driven; this is represented by the transfer function 'receive' (an external agent, a human user or a subsystem has the initiative). Whenever the system receives external data, the controller checks for abnormal behaviour in the schedule execution. The system checks the observed values by actively seeking new data (through 'obtain' transfer function).

3.2.4. *Domain knowledge specification*

The domain knowledge is static in the sense that it presents a description of the facts about the domain without knowing how this knowledge might be used in problem solving. In this sense, domain knowledge is task dependent and domain specific. A scheduling task in general is relatively weak in providing domain knowledge because of the dynamic nature of the scheduling task as compared with other synthetic tasks, e.g. configuration designs.

CommonKADS does not prescribe a fixed formalism for describing domain structure. However, 'frames', 'is a hierarchy' and 'rule sets' are examples of domain structures used in CommonKADS. To acquire domain knowledge, each item of declarative knowledge was classified into four types: concepts, attributes, relations and rule types.

The starting point for domain knowledge modelling is the analysis of an interview transcript to generate a set of concepts, relations and attributes. Using the transcript of an interview in Adesola *et al.* (2000), the following domain knowledge

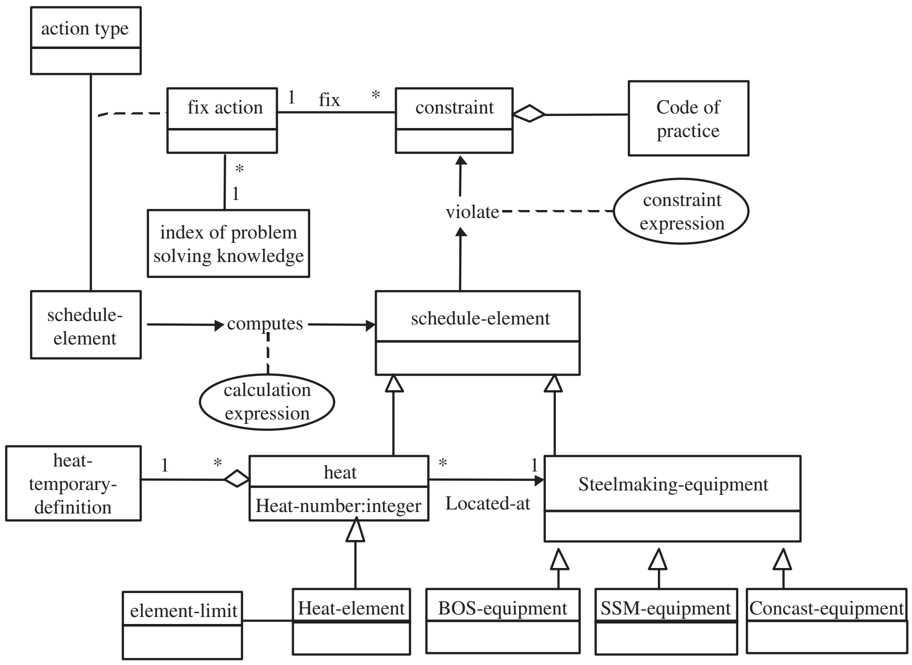

Figure 8. Typical domain knowledge types in MSD through monitor–construct–repair.

was elicited. Figure 8 shows the domain schema drawn from elicited domain knowledge using XPat. The natural language analysis (Vescovi *et al.* 1993) technique provides a good first guidance for understanding the meaning of text in the transcript. The main problems with this approach, however, are that it is time consuming and one needs different scenarios for analysing different types of sentences.

4. Knowledge model validation

This section presents the validation of the reasoning process through paper-based simulations, as adapted from CommonKADS. The validation process consists of three components. The first concerns what is happening and the entities involved in domain-specific terminology; this was validated for accuracy. The second describes the associated inference functions and knowledge roles required, determining the actions; this was validated for completeness. (Abstract names of data objects indicate their role in the reasoning process.) The third provides for an additional explanation and comments about actions which are taken; this was validated for consistency. Therefore, accuracy, completeness and consistency are used as the measurement criteria for the validation; they are also supported by existing literature on knowledge analysis. The knowledge model is validated with three case studies that reflect the required system behaviour. A paper trace in terms of the knowledge model constructs is generated. A set of questionnaires was designed from the measurement criteria to validate the knowledge model. Three experts validated the knowledge model, which lasted 12 man-hours over three days. During this time changes were made to the structure and contents of the knowledge model. Informal discussion about the behaviour of the model was captured on audiotape and transcribed; this was later used to support changes to the paper trace where appropriate.

The case studies chosen represent the most common categories of disturbance in steel-making scheduling. The inference structure for MSD is validated with domain experts to confirm that it is sufficiently detailed. The validation also indicates that it is easy to find domain knowledge that could act as static roles for the inference structure. (These are more or less stable over time; they specify collection of domain knowledge that is used to make inferences.) Table 1 shows the three case studies for the knowledge model validation; there are two scenarios for each case study. For a complete description of all three cases with the scenarios, see Adesola (2002). Appendix A shows a paper-based simulation result for steel OOS BOS vessel B (Scenario 1) case study result as a sample.

Each scenario describes a specific disturbance, the conditions and actions. The paper simulation is performed in a tabular form with three columns. The first column identifies what happens in the domain, the entities involved including the scheduler, the shop floor operator and the systems used. In the second column, the knowledge model (inference functions and knowledge roles) identifies necessary variables and rules to generate possible actions. The knowledge model realizes the required problem solving through the sequence of inferences defined. The third column provides explanation and comments about action to validate the knowledge model. Each expert was given: (1) the three case studies, (2) six paper simulation tables and (3) a set of semistructured questionnaires. Each expert was required to study and cross-check items (1) and (2). Each column of the paper simulation table is validated against the measurement criteria. For example, the domain data are validated for accuracy, a knowledge model is validated for completeness and the explanation is validated for consistency. After cross-checking,

Case studies	Description
A	Out-of-specification in the BOS vessel
B	Out-of-specification in the secondary steel-making (SSM) plant
C	Steel temperature too hot BOS/SSM plant

Table 1. Case studies for knowledge model validation.

experts were asked to complete the validation questionnaire and comment on their overall experience. Table 2 presents the validation results.

The result of paper-based simulation (table 2) indicates the model match problem-solving behaviour. The experts scored 84% (average) for accuracy of domain data, which indicates the domain data reflected typical systems data for a specific category of disturbance in steel-making scheduling. For the knowledge model, the experts scored 77% to indicate that the knowledge model completely captures the inferencing process in the problem-solving behaviour to MSD. The result of the explanation (79%) indicates there is consistency in the relationship expressed between the domain data and the analysis in the knowledge model.

5. Discussion

There is now an overall consensus that the process of building a knowledge-based system may be seen as a modelling activity (Studer 1998). A generic inference structure for disturbance management in steel-making scheduling has been constructed. Many components have been identified with a potential for reuse. The methodology applied in this research follows an academic approach that pursues a framework for capturing human knowledge and for conversion into a system for reuse combined with parallel validation through case studies in the industrial environment.

A methodology (XPat) to capture process knowledge has been applied. The method aims to improve the natural knowledge elicitation technique by making it easier for experts to express and display their expertise within a flexible and structured process. From the interpretation of an interview transcript, ten categories of disturbances in steel-making were elicited from experts. Eleven knowledge items were defined for each category of disturbances. These knowledge items constitute PSK for MSD in steel-making. The knowledge items include implementation flowcharts for each PSK. This part of the research is prone to bias due to interpretation by researchers. The bias is minimized by careful design of the questionnaire and through additional observation. The benefit of developing the knowledge model this way is that it is easy to trace each function through the inference function. The knowledge model is reusable and adaptable. One of the main advantages of model-based knowledge engineering is the concept of reusability. Potentially, a combination of model elements can be reused. It is intuitively clear from the knowledge model in the research that large parts of the model are not specific to steel-making. Parts of the task and the inference knowledge can reoccur in other domains and/or tasks.

The knowledge model has been validated with three case studies. The approach employed for validation was repeated 'walk-through' paper simulation with domain experts. Where necessary, feedback from the walk-through tests was used iteratively

Case studies	Scenario	Knowledge model simulation	Measurement criteria	Expert A	Expert B	Expert C	Score E/12 × %
A Steel OOS BOS vessel	1. Vessel B	Domain data	Accuracy	4	4	4	100%
		Knowledge model	Completeness	4	3	4	91%
		Explanation	Consistency	3	4	4	91%
	2. Vessel C	Domain data	Accuracy	4	3	3	83%
		Knowledge model	Completeness	4	3	4	91%
		Explanation	Consistency	3	3	4	83%
B Steel OOS SSM plant	1. Flusher B	Domain data	Accuracy	4	3	3	83%
		Knowledge model	Completeness	4	2	3	75%
		Explanation	Consistency	4	3	4	91%
	2. RH Degasser	Domain data	Accuracy	4	4	3	91%
		Knowledge model	Completeness	2	3	3	66%
		Explanation	Consistency	3	3	3	75%
C Steel temperature too hot BOS/SSM plant	1. Vessel B	Domain data	Accuracy	3	3	3	75%
		Knowledge model	Completeness	2	2	2	50%
		Explanation	Consistency	2	2	2	50%
	2. RH- Degasser	Domain data	Accuracy	3	3	3	75%
		Knowledge model	Completeness	4	3	4	91%
		Explanation	Consistency	4	3	3	83%
				61 = 61/72 84%	54 = 54/72 75%	59 = 59/72 81%	58 = 58/72 80%

A score of 4 means 'strongly agree' and 1 means 'strongly disagree'.

Table 2. Knowledge model validation results.

to modify and extend the knowledge model. The case study has shown that the knowledge model has accurately captured PSK and rules. The knowledge model is generic to steel-making. Although the knowledge model has not been tested in other industries, it is expected that it can provide a basis for analysing management of schedule disturbance in other sectors. The knowledge model reflects the expert reasoning process.

However, as with any other research, the methodology has some limitations. Its weakness is its applicability to other domains; since time and resources limited the research, it has not been tested widely. With more time and resources, the methodology should be tested in a complex environment, like managing airline gate assignment.

In future, a prototype decision support system will be developed (for offline use) to implement the knowledge model and exploit the inference structure in other application areas where disturbance management is critical to the business.

6. Conclusions

This paper has identified human expertise as the dominant factor in manual scheduling. It has demonstrated how the result of knowledge elicitation through the XPat methodology can be used to develop a knowledge model for MSD in steel-making. This proves the hypothesis of this research.

The main problem is that MSD in steel-making is not formalized. This research has formalized elicited knowledge for decision support to MSD in steel-making. The approach is novel and supports a 'middle-out' route for completing the knowledge model in CommonKADS. The approach demonstrates that it is possible to construct a generic inference structure from PSK identified by flowcharts. The inference structure is sufficiently detailed for implementation to provide the reasoning process for MSD.

The knowledge model has been validated with a case study in out-of-specification steel—BOS vessel. The case study shows that the knowledge model has accurately captured the PSK and rules. The knowledge model is generic to steel-making, and although it has not been tested in other industries, it is expected that it can provide a basis for analysing the management of schedule disturbance in other sectors.

In conclusion, this paper has presented the development of a knowledge model to manage schedule disturbance in steel-making. In future, it can be implemented to develop a decision support system for the disturbance management.

Acknowledgements

Research was carried out as part of a doctoral thesis at Cranfield University and was sponsored by EPSRC and Corus UK Ltd. The authors thank the employees of Corus UK for help and support during the knowledge elicitation.

Appendix A: Case Study A. Paper-based simulation result for scenario 1, steel out-of-specification (OOS) BOS at vessel B

Steel chemistry is OOS in a BOS vessel B. The scheduler received the information via the operator at the BOS vessel. The quality code is 1522; heat number is 2461. The customer will not accept OOS steel. The number of the ladle in sequence is 8; the heat position in sequence is 3; the process route is vessel–flush–degas. Figure 10 shows an annotated inference structure for MSD with data about OOS steel in a BOS vessel B.

Domain	Knowledge Model	Explanation
Receive system data Scheduler: [Look up to identify the problems] Operator: 'B-Vessel heat is OOS' System: BC System — Quality Code Data Merlin Schedule — Heat Location Vax Mgmt System — Analysis Summary	MANAGE–SCHEDULE–DISTURBANCE: monitor: system-data; Heat location = Vessel B	Scheduler monitors system data, receives operator feedback for which the manage schedule disturbance task is started. The heat located at vessel B is OOS
Obtain additional data Scheduler: [Asked operator what element is OOS] Operator: [Value of Carbon and Nickel is too high] System: Vax Mgmt System — Heat Status and Analysis Summary (AS)	OBTAIN: system-data; Heat carbon-element-analysis = false Heat nickel-element-analysis = false	Carbon and nickel are the two elements out of specification as the element analysis value indicates false
Compare analysis summary with customer limit Scheduler: [Check specification] Operator: [Awaiting instruction] System: BC System — QC 10 and QC 02 for Quality code 1522 Vessel B Heat no. 2461 Heat/sequence of heats = 3 of 8	verify: truth-value; Heat within-customer-limits = false Heat element-controllable = false Heat carbon-element-analysis = 0.2 Heat carbon-element-analysis-customer-lower-limit = 0.02 Heat carbon-element-analysis-customer-upper-limit = 0.07 Heat nickel-element-analysis = 0.1 Heat nickel-element-analysis-customer-lower-limit = 0.0 Heat nickel-element-analysis-customer-upper-limit = 0.08	This inference checks whether element analysis is within customer limits and controllable, it returns a truth value — true or false. It checks each element against a code of practice specification for customer acceptance limits. In this case, element OOS includes controllable and uncontrollable elements: carbon and nickel exceed the customer limit

Element	Analysis	Customer limit	
	summary	Lower limit	Upper limit
C	0.2	0.02	20.07
C	0.2	0.02	20.07
NI	0.1	0.0	0.08

Temperature: not affected

Specify PSK Scheduler: [Specify PSK] Operator: [Awaiting instruction] System: 'Call Vessel out-of-specification PSK'	specify: fix: Vessel::OOS-problem-solving-knowledge	This inference states the precise PSK to fix out-of-specification steel in the BOS vessel. The inferencing method is forward reasoning. A PSK is specified to fix out-of-specification steel in the vessel
Generate possible actions Scheduler: [Generate possible actions] Operator: [Awaiting instruction] System: 'Check condition and generate action' 'Calculate time available for rectification' Condition: Time (i.e. time on the clock) = 20:20 Required concast delivery time = 21:30 Q. Is there time available for rectification? Yes	generate: actions; Basis for time to rectify in the vessel Heat tap-to-open-time = 65 (for Quality 1522) Heat time-until-required-at-concast = [delivery-time less current-time] Heat.time-available-from-tap = [delivery-time less heat tap-to-open-time] Heat.time-available-for-rectification = [time-until-required-at-concast less time-available-from-tap]	This inference produces action to solve the OOS problem in the vessel. This is achieved by computing all possible combinations in an algorithm as the rules for calculating time indicate. For example, depending on the state of the plant, the following questions are asked: Is there time available for rectification? Yes Although there is no time to rectify in the Vessel the element OOS is uncontrollable.

	Rules for calculating extra time per sample number
	IF at vessel and sample = 1 THEN
	extra-time = 10
	ELSE IF at Vessel and sample = 2 THEN
	extra-time = 5
	ELSE extra-time = '0'
	END
	Heat-time-available-for-rectification = Heat-time-available-for-rectification + extra-time
		The scheduler instructs the operator to reblow in the vessel. The scheduler checks for quality that can be made from heat-need-outlet, and found there is no quality that can be made from heat-need-outlet. The scheduler instructs the operator to tap as an alternative quality

	The Rule IF time-available-for-rectification = > 10 THEN there is sufficient-time-to-rectify in the vessel, tap-as-normal and continue- to-the-next-station ELSE	

Table 3. Results from a paper simulation process.

Domain	Knowledge Model	Explanation

	IF time-available-for-rectification < 10
	THEN
	there is insufficient-time-to-rectify in the vessel, ask-the-caster-operator-to-slow-down-casting-for-sufficient-time, rectify in the vessel, tap-as-normal and continue-to-the-next-station
	ELSE create heat-needs-outlet
	END
	Comments
	Note that a judgement may need to be made concerning the actual time for rectification at present this is set to 10 min, e.g. 10 min modify: schedule;	

Display modification instruction Scheduler: [Advice shopfloor about possible actions] Operator: [Receive possible action] System: 'Display problem statement, advice and explanation'	Problems element C is out of spec for Customer Limit at vessel B. Controllable: true; Analysis: 0.2. upper limit: 0.07. lower limit: 0.02 element NI is out of spec for Customer Limit at vessel B. Controllable: false; Analysis: 0.1. upper limit: 0.08. lower limit: 0.0	This inference seeks to adapt the schedule by displaying relevant actions as instructions to advice shop floor operators in order to maintain stability in the steelmaking process. In this case, although there is time to rectify in the vessel, the element out of spec is uncontrollable and the customer will not accept steel with OOS elements. This inference display problems, advice and explanation. The heat will have to be rescheduled. The scheduler informs weekly planner by complete the production planning and report sheet
	Advice 'Create Heat Needs Outlet, find alternative quality and available caster for heat-needs-outlet. Tap heat as alternative quality'	
Display rescheduling instructions	OBTAIN: rescheduling instructions; Not applicable in this case	This transfer function is only required if the scheduler cannot manually modify schedule as illustrated in the above two scenarios

Table 3. Continued.

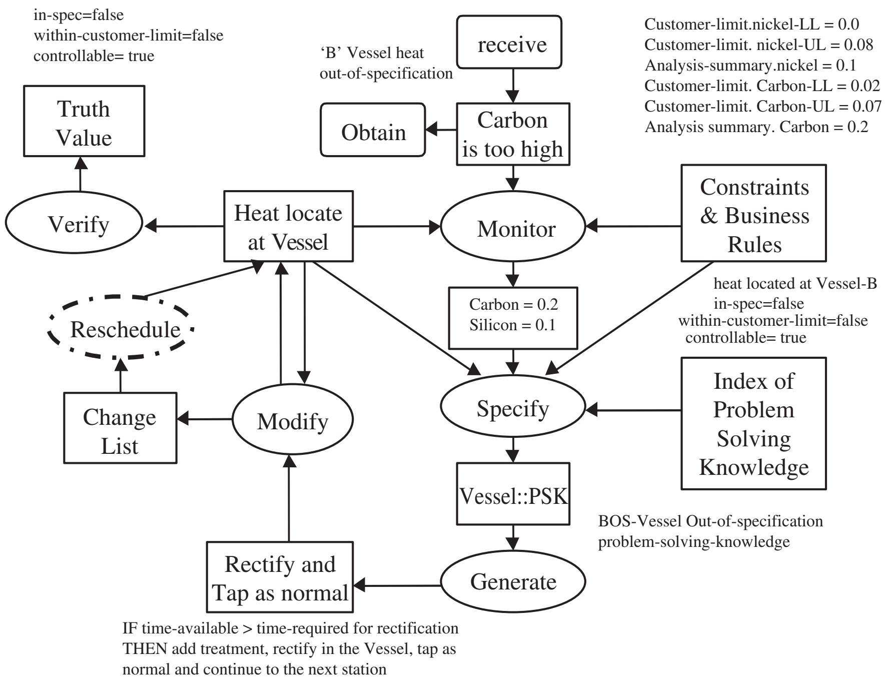
in-spec=false
within-customer-limit=false
controllable= true
Truth Value
Verify
'B' Vessel heat out-of-specification
receive
Obtain
Carbon is too high
Customer-limit.nickel-LL = 0.0
Customer-limit. nickel-UL = 0.08
Analysis-summary.nickel = 0.1
Customer-limit. Carbon-LL = 0.02
Customer-limit. Carbon-UL = 0.07
Analysis summary. Carbon = 0.2
Heat locate at Vessel
Monitor
Constraints & Business Rules
Reschedule
Carbon = 0.2
Silicon = 0.1
heat located at Vessel-B
in-spec=false
within-customer-limit=false
controllable= true
Change List
Modify
Specify
Index of Problem Solving Knowledge
Vessel::PSK
BOS-Vessel Out-of-specification problem-solving-knowledge
Rectify and Tap as normal
Generate
IF time-available > time-required for rectification
THEN add treatment, rectify in the Vessel, tap as
normal and continue to the next station

Figure 9. Annotated inference structure for steel chemistry out-of-specification (OOS) steel in a BOS vessel.

Condition:

Time is 16:45; required concast delivery time is 17:55.
There is time to contact the customer.
Customer will not accept OOS steel.
There is no quality that can be made from heat-need-outlet.

Action: Create heat-needs-outlet: check the 24 h (hard copy of slab or bloom machine) schedule for qualities that can be made from heat-needs-outlet. Tap heat as alternative quality.

For practical purposes, the paper simulation process is conducted in tabular form (Table 3). The reasoning process used in MSD is described. The case study is an example of schedule disturbance covered in the validation process. The knowledge components of the inference structure, i.e. the dynamic roles, e.g. the system data, the schedule, the discrepancies, etc., have been instantiated with domain specific objects (figure 9). (These dynamic roles are run-time inputs and outputs of inferences. They have different instantiations at each invocation.)

The case study reflects the required system behaviour. After several iterations and refinement, the inference structure was validated with domain experts to confirm that the inference structure was sufficiently detailed. The validation also indicates that it was easy to find domain knowledge that could act as static roles for the inference structure. A paper trace in terms of the knowledge model constructs is generated. (These static roles are more or less stable over time; they specify collection of domain knowledge which is used to make inferences.)

References

Adesola, B., 2002, A knowledge model for decision support to manage schedule disturbance in steelmaking. PhD thesis, Cranfield University.

Adesola, B., Roy, R. and Thornton, S., 2000, XPat: a tool for manufacturing knowledge elicitation. In R. Roy (ed.), *Industrial Knowledge Management — Micro KM Approach* (Berlin: Springer).

Breuker, J., Wielinga, B., Someren, M. V., De Hoog, R., Schreiber, G., De Greef, P., Bredeweg, B., Wielmaker, J., Billaut, J., Davoodi, M. and Hayward, S., 1987, *Model-driven Knowledge Acquisition: Interpretation Models*. Technical Report, ESPRIT Project P1098 (Amsterdam: University of Amsterdam).

Brown, M. C., 1988, The dynamic rescheduler conquering the changing production environment. In Proceedings of the 4th IEEE Conference on AI Applications, San Diego, CA, USA, pp. 175–180.

Chandrasekaran, B. and Johnson, T. R., 1993, Generic tasks and task structures: history, critique and new directions. In J. M. David, J. P. Krivine and R. Simmons (eds), *Second Generation Expert Systems* (London: Springer), pp. 233–272.

Cowling, P. and Rezig, W., 2000, Integration of continuous caster and hot strips mill planning for steel production. *Journal of Scheduling*, **3(4)**, pp. 185–208.

Dorn, J., 1995, Case-based reactive scheduling, In R. Kerr and E. Szelke (eds), *Artificial Intelligence in Reactive Scheduling* (London: Chapman & Hall), pp. 32–50.

Dorn, J. and Shams, R., 1991, An expert system for scheduling in a steelmaking plant. In Proceedings of the 1st World Congress on Expert Systems, pp. 395–404.

Dorn, J. and Shams, R., 1996, Scheduling high-grade steelmaking. *IEEE Expert Special Issue on AI in Steelmaking*, **11**, 28–35.

MOKA Consortium, 2001, Managing engineering knowledge. In M. Stokes (ed.), *MOKA: Methodology for Knowledge Based Engineering Applications* (Professional Engineering Publishing, UK).

Numao, M. and Morishita, S., 1991, Cooperative scheduling and its application to steelmaking processes. *IEEE Transactions on Industrial Electronics*, **38**, 150–155.

Schreiber, G., Akkermans, H., Anjewierden, De Hoog, R., Shadbolt, N., Van de Velde, W. and Wielinga, B., 1999, *Knowledge Engineering and Management: The CommonKADS Methodology* (Cambridge, MA: MIT Press).

Studer, R., Benjamins, V. R. and Fensel, D., 1998, Knowledge engineering, principles and methods. *Data and Knowledge Engineering*, **25**, 161–197.

Tansley, D. S. W. and Hayball, C. C., 1993, *Knowledge Based Systems Analysis and Design: A KADS Developer's Handbook* (Englewood Cliffs: Prentice-Hall).

Vescovi, M., Iwasaki, Y., Fikes, R. and Chandrasekaran, B., 1993, How things are intended to work: capturing functional knowledge in device design. In Proceedings of the 13th International Joint Conference on Artificial Intelligence.