International Journal of Production Research vol. 42 No 18, September 15, 2004, pp. 3975 – 3994

Development of a Knowledge Model for Managing Schedule Disturbance in Steelmaking

R.Roy[1], B.A. Adesola[1], and S.Thornton[2]

[1]Department of Enterprise Integration, School of Industrial and Manufacturing Science,

Cranfield University, Bedford, MK43 0AL, UK. Email: {r.roy or b.adesola}@cranfield.ac.uk

[2]Corus UK Ltd., Teesside Technology Centre, P.O. Box 11, Grangetown, Middlesborough,

Cleveland TS6 6UB, UK. Email: Steve.Thornton@corusgroup.com

Abstract

This paper presents the development of a knowledge model, which describes the reasoning process in managing schedule disturbance (MSD) in steelmaking. Literature review shows the lack of research in developing a knowledge model for decision making in steelmaking. In this paper the knowledge model distinguishes three knowledge categories: the task knowledge, the inference knowledge and the domain knowledge. The knowledge is captured for ten most common types of disturbances in steelmaking. It is observed that a common inference model exists for the disturbance management. A knowledge elicitation methodology called XPat [eXpert Process Knowledge Analysis Technique] combined with CommonKADS approach was used to capture process knowledge for managing schedule disturbance in steelmaking. 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.

Keywords:

Managing schedule disturbance, steelmaking, application of knowledge modelling, knowledge engineering.

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 steelmaking of a twelve-hour shift by a human scheduler takes at least two hours. This manual scheduling time means that it is difficult to react to unforeseen production events, e.g. rushed order, especially during night shifts and weekends when experts might not be available (Cowling and Reizig, 2000). In addition, when it is necessary in advance to schedule several parallel activities, which 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 lack of continuity in detailed schedule execution. Due to the dynamic nature of the steelmaking process however, it is often difficult to maintain the original short-term schedule. The schedule disturbance management is a manual process and requires many years of experience. The research presented in this paper intends to formalise the knowledge required to manually modify a schedule in order to minimize the impact of any disturbance. The knowledge can then be used in a decision support system to improve the management of schedule disturbance and avoid unnecessary rescheduling.

This paper presents a knowledge model for decision support to manage schedule disturbance in steelmaking (hypothesis). Knowledge modelling is an approach to develop a Knowledge-Based System (KBS). This is a transformation approach to knowledge capture, by modelling ‘what an expert does’. A knowledge model for the

MSD is a semi-formal representation of the tasks involved, the inference mechanism to manage the disturbance and any domain specific knowledge. In section 2, the paper presents an overview of managing schedule disturbance in steelmaking. Section 3 describes the approach for knowledge model development and section 4 presents validation of the knowledge model through case studies. Section 5 in the paper presents a discussion on the research methodology and results and finally section 6 concludes with the limitation of the approach and the future research.

2. Managing Schedule Disturbance in Steelmaking

Scheduling in general is a dynamic activity where several repair actions may be required depending on internal or external influences. Managing schedule disturbances (MSD) is a complex knowledge intensive activity, performed by human experts. This activity encompasses several ranges of dynamic tasks, such as generating alternative actions and making decision. MSD is necessary to ensure reaction in one domain does not affect the rest of the schedule. Disturbance in steelmaking 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 steelmaking 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, rework due to wrong product specifications. MSD in steelmaking 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 *time available* and the state of the overall steelmaking schedule.

The task of managing schedule disturbance is popularly termed as 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 Gantt chart. The use of Gantt chart in general scheduling is widespread. The Gantt chart 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. It is observed 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 problem*' 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 steelmaking can be developed. The overall argument is that MSD in steelmaking 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, compatibility constraints of higher grades of steel impose requirements on the sequence in which orders are produced, (Dorn and Shams, 1991). To achieve certain characteristics, when chemicals are added it may also react with the steelmaking aggregate.

3. Developing a knowledge model for MSD in steelmaking

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 minimise 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 steelmaking.

3.1. The Approach

In steelmaking scheduling, each categories of disturbance are handled differently. It is essential that each categories of disturbance and possible action to modify or reschedule is well understood. In order to model the process of MSD in steelmaking scheduling ten most common categories of disturbances were identified: Steel out-of-specification in BOS plant, Steel out-of-specification in SSM plant, Steel temperature too hot, Steel temperature too cold, Hot Metal Supply (HMS), 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 semi structured interviews with experts from three different plants within Corus. Authors also studied shift logs for any disturbance and analysed previous company documentation. Problem solving knowledge (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 (KMFs) 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 realisation 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 steps identify and capture domain specific knowledge and their sources. Steps three and four 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 have been identified and elicited the next step is to classify the nature of task. To elicit knowledge, simple structured questions were developed based on 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, the authors observed experts during MSD. Figure 1 present a problem solving (procedural) flow chart for steel out-of-specification in the Basic Oxygen Steelmaking (BOS) Vessel.

Different categories of disturbance require different PSK. A PSK in this case consists of eleven knowledge items as listed below.

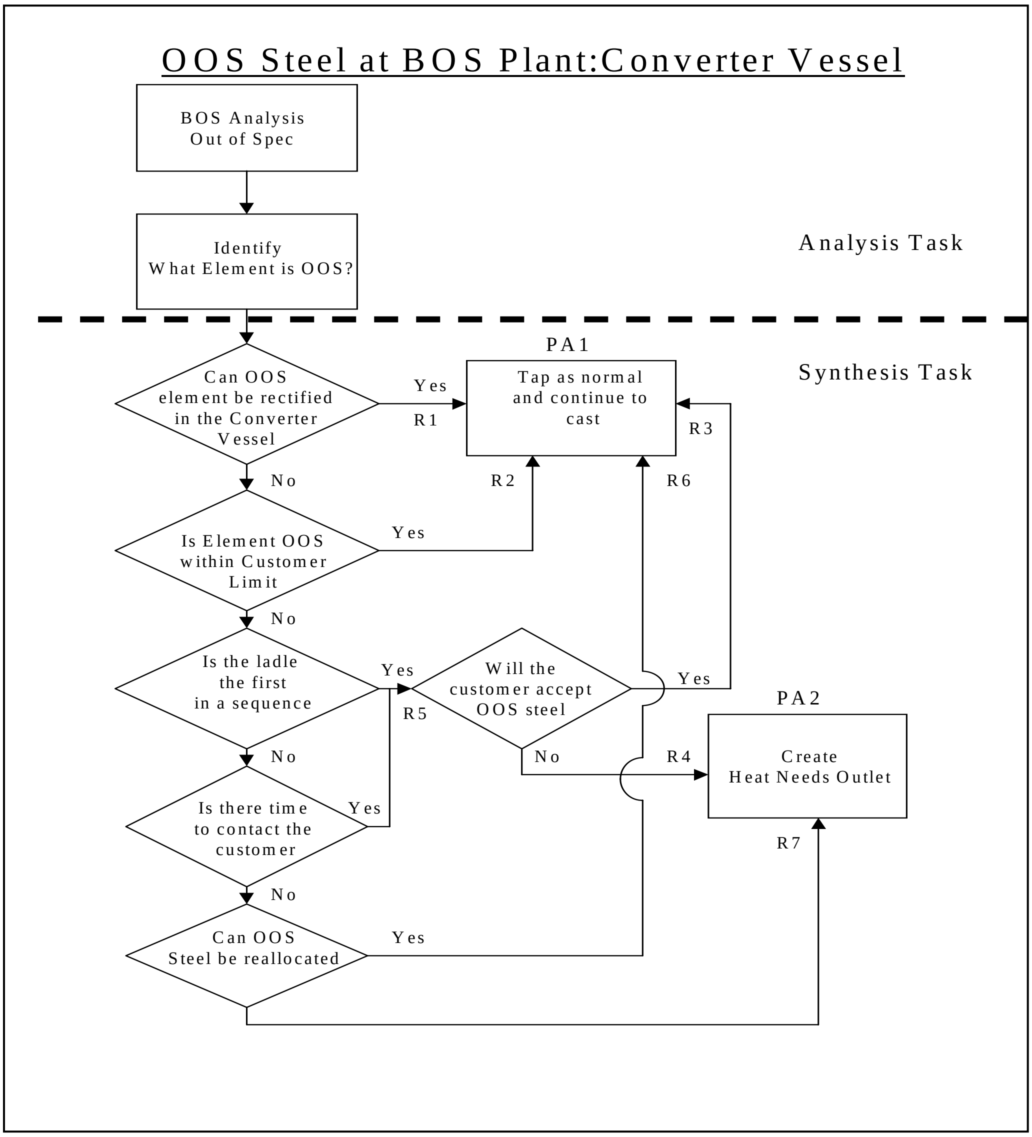

Figure 1: Problem solving (procedural) flowchart for out-of-specification (OOS) Steel in the BOS Vessel

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

Item number nine represents procedural or problem solving flowcharts, which illustrate the flow of reasoning. Together all of 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 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 analytic and synthetic tasks. The nearest inference structure in this case is the configuration design task template in Schreiber et al., (1999). Configuration design method uses a variation of the propose-critique-modify class of method described in Chandrasekaran and Johnson (1993). The "propose" part of the method is similar to the predictive schedule, the "critique-modify" doesn't 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, analytic and synthetic (Breuker, et. al., 1987; Schreiber, et. al., 1999; Tansley and Hayball 1993). It captures typical analytic and synthetic features of the problem-solving knowledge. For each disturbance category, the flowchart method was used to collect procedures used for problem solving. The flowchart indicates how steelmaking schedulers reason during 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 reasoning process onto the knowledge model specification. The '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 the 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 middle-out route was influenced by the *flowchart method* of collecting problem solving knowledge.

3.2.2. Inference Knowledge Specification

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

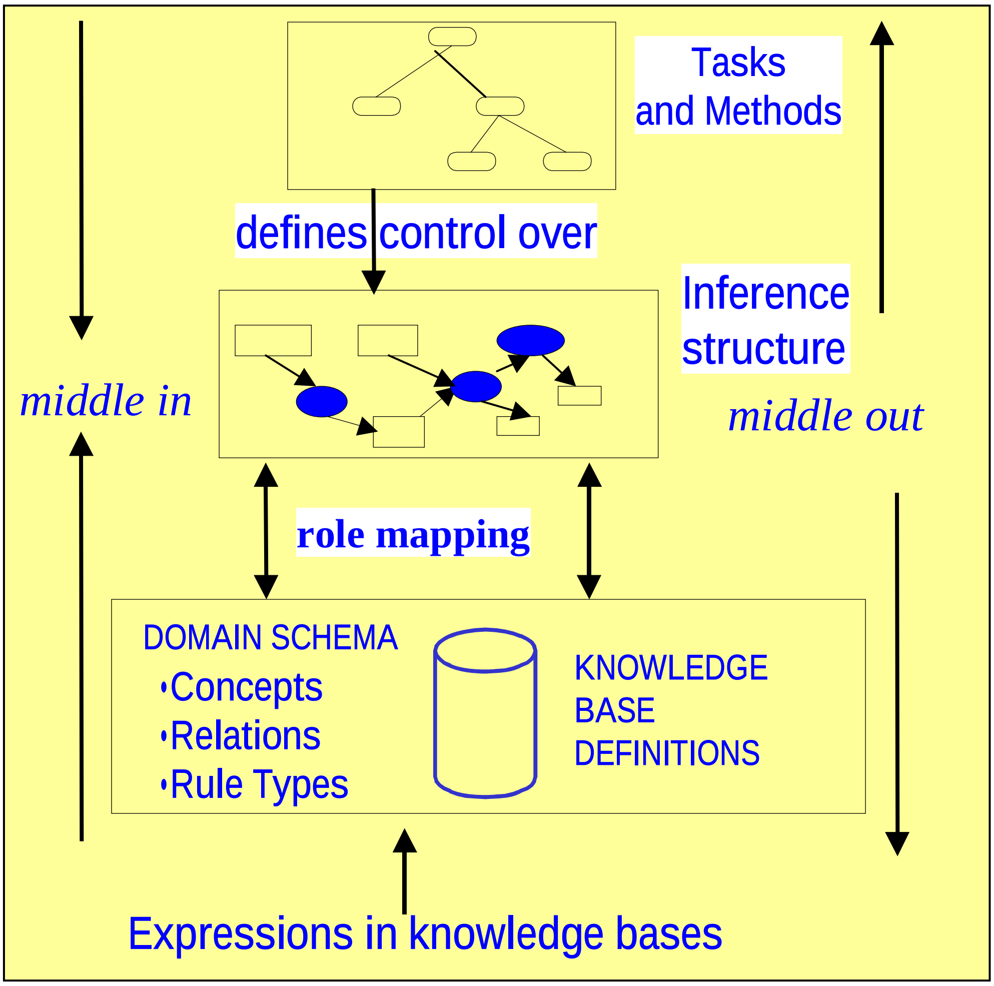

Figure 2: Knowledge Model Specification (Schreiber, et. al., 1999).

The Evolution of Inference Structure for MSD in Steelmaking

Figure 3 illustrate 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 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). Authors studied all ten categories of disturbances (corresponding to ten categories of PSKs) identified during the research. It is observed that a common pattern exists between the PSKs, which suggests steelmaking scheduling experts' reason about the problem solving in a similar way. To achieve a generic inference structure the steps in problem solving were identified in order abstract patterns of behaviour. The steps were then put together to form an inference structure through an iterative approach which involve 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.

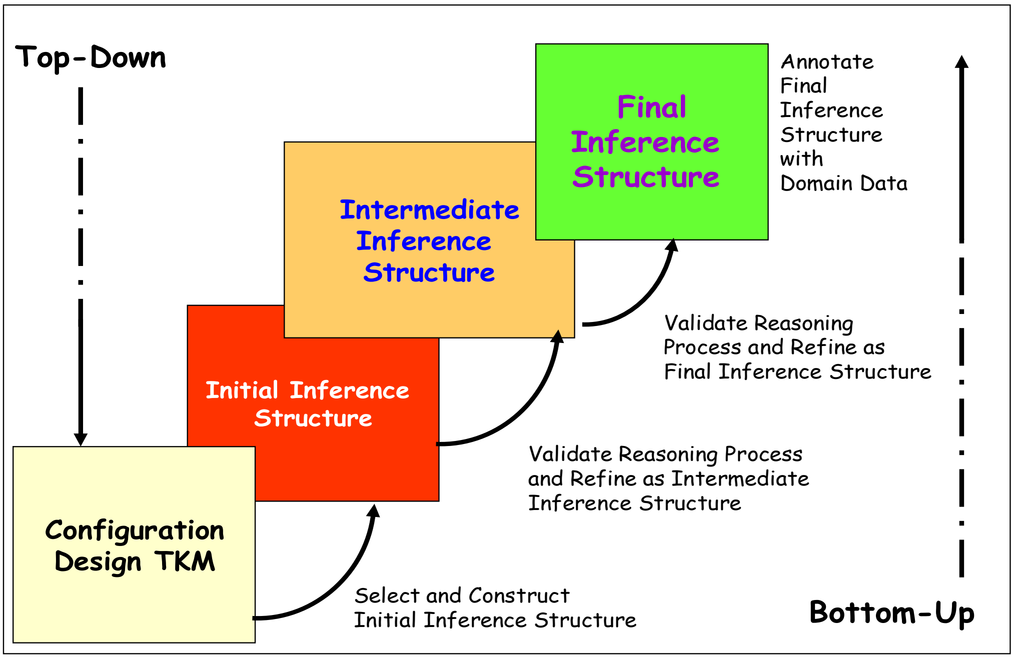

Figure 3: Evolution of Inference structure for MSD in steelmaking

Changes made to the initial and intermediate inference structure include identifying 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.

The Final Inference Structure

The final inference structure in Figure 4 demonstrates a combination of task types. Monitoring is an analytic task to establish 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 minor or 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. In order to identify a specification, a synthetic task may initially contain an analytic task (Breuker, et. al., 1987).

Construction of the inference structure is realised by identifying first, the inference functions for the analysis task, and then synthetic tasks. For the analytic task, the three inferences identified were ‘monitor’, ‘verify’ and ‘specify’. For the synthetic task, the three inferences identified were ‘generate’, ‘modify’ and ‘reschedule’. The flowchart method abstracts the analytic task of monitoring and classifying system behaviour. It provides a top-level view of the problem-solving method.

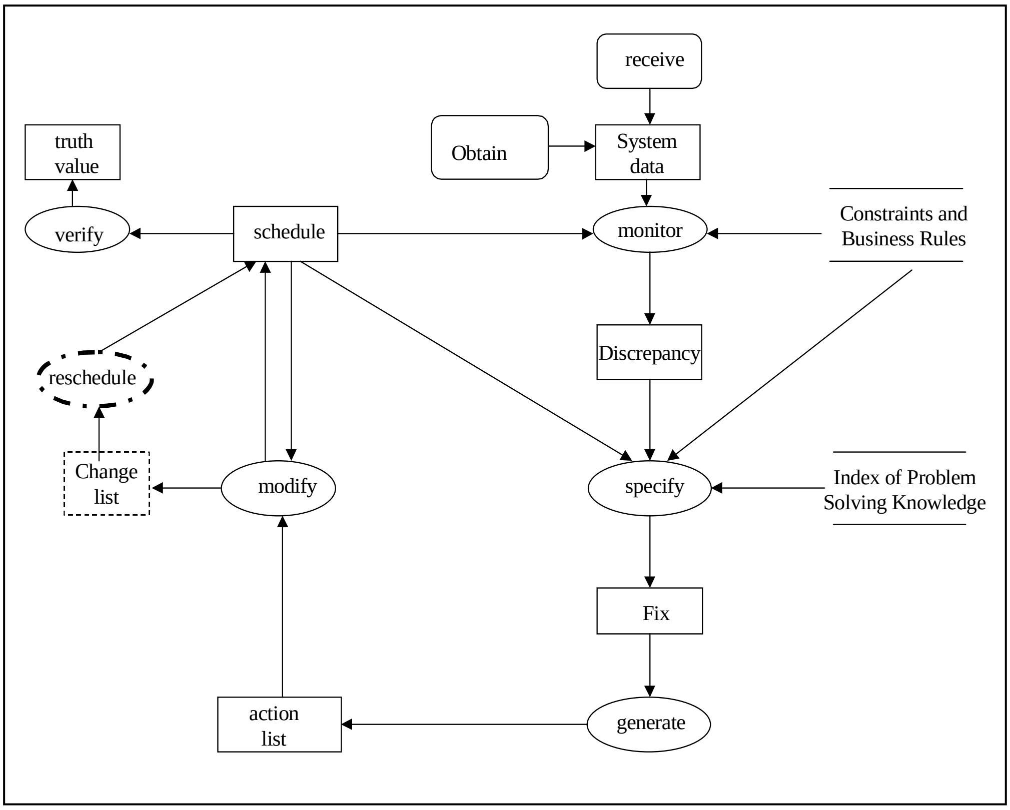

Figure 4: Inference structure for MSD in steelmaking

The human scheduler receives system data (unsolicited message), analyses it and may obtain additional data if he chooses. The 'system data' is a message, informing that there has been a violation. The violation is of 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 steelmaking. 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 realise 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, that is, leaf functions.

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 realise the goals. In this case the task is to manage-schedule-disturbance to maintain sequence of heats, in order to minimise the risk of needing to reschedule orders. The task knowledge is described in a hierarchical fashion. The top-level task[1] MANAGE-SCHEDULE-DISTURBANCE is decomposed into smaller tasks, which in turn is split into even smaller tasks. The task does not include rescheduling task. It only decides whether rescheduling is necessary, therefore rescheduling is shown as dashed dotted line in the

[1] Composite task is a problem solving action it specifies an abstraction level and require decomposition before it can be executed.

diagram (see Figure 5). The lowest level tasks[2] are the leaf functions and are linked to inferences and transfer functions in the inference structure.

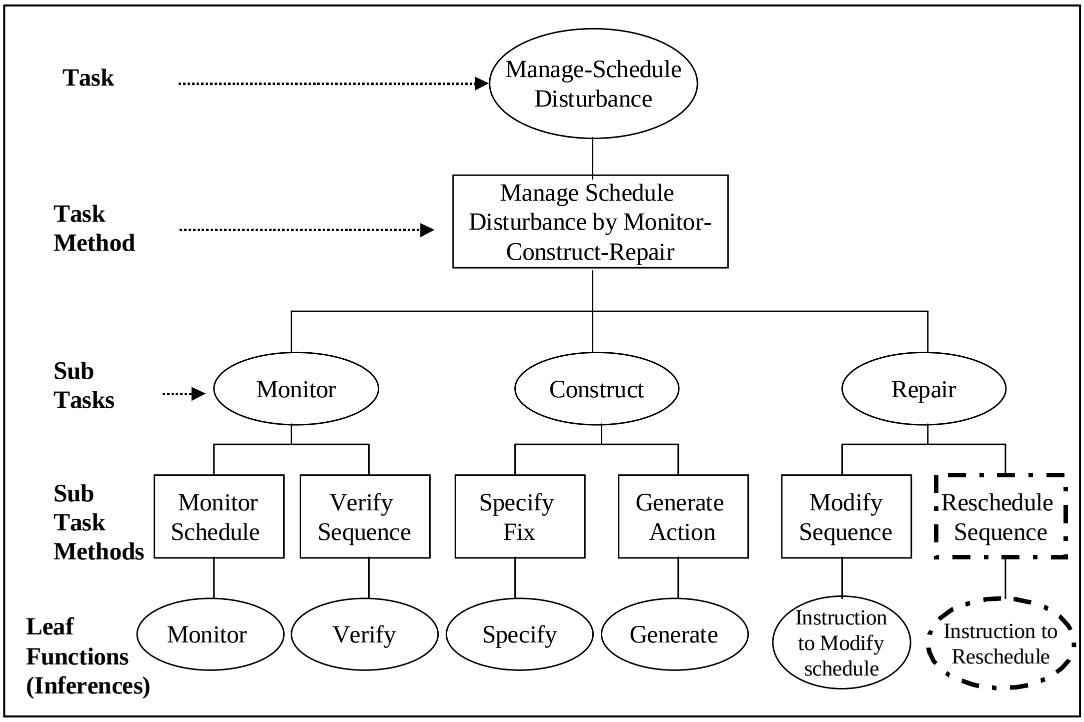

Figure 5: Task Structure for MSD in steelmaking

The task method defines a reasoning goal, in other words, how the task MANAGE-SCHEDULE-DISTURBANCE can be realised through decomposition into sub-functions. In this case, the top-level task is decomposed by a task method MONITOR-CONSTRUCT-REPAIR. The sub task 'monitor' describes the activities that track schedule execution and verify the state of the original schedule. This involves looking up at the Gantt chart and other information sources for systems data. The sub-task 'construct' specifies problem solving knowledge 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 solution to maintain stability of the steelmaking process. For example, in the event of out-of-specification steel in BOS Vessel the scheduler uses his/her expertise to generate possible actions

[2] Primitive task as they have enough knowledge available that they can be solved without further decomposition.

by analysing requirements and trade-offs to balance the objectives of the plant. The sub task ‘repair’ presents possible modification actions to remove constraint violation or generate change list for rescheduling action where it is necessary. The latter action is the last resort.

The sub-tasks are decomposed into sub-task methods. For example, the sub task “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. In order to respond to system data that indicates 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 temperature is too hot in a ladle such that it is not possible to send the steel to a caster, verify sub-task will return false, a Boolean value. This result of verification will be passed onto specify sub-task which will call the appropriate problem solving knowledge (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’ function, as procedures for handling specific problems. In order to present actions to address a specific problem, the sub-task ‘modify’ is employed and the ‘reschedule’ function is used when it is not feasible to ‘modify’ a schedule.

Control Structure

In Figure 6 a graphical representation of the method control structure for MONITOR-CONSTRUCT-REPAIR task is shown. This presents 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. In Figure 7, a generic default method applicable to MSD in steelmaking is shown. The method is data driven, this is represented with the use of 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).

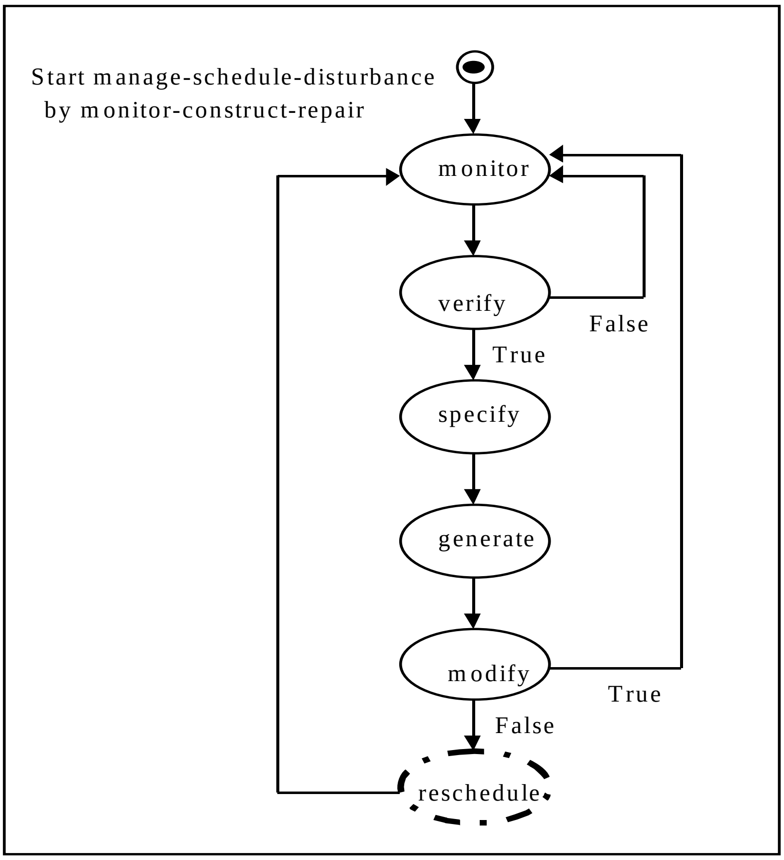

Figure 6: Control Structure for MONITOR-CONSTRUCT-REPAIR

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. Scheduling task in general is relatively weak in providing domain knowledge because of the dynamic nature of scheduling task as compared to other synthetic tasks e.g. configuration designs.

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 steelmaking.

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 minimise discrepancy
Constraint: a control or something that limit process behaviour.

Input: Complaints about disturbance affecting a schedule

Output: A set of instructions to minimise the impact of schedule disturbance in steelmaking.

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 steelmaking

CommonKADS does not prescribe a fixed formalism for describing domain structure. However, 'frames', 'is-a hierarchy', 'rule sets' are examples of domain structures, which are used in CommonKADS. In order 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 interview transcript to generate a set of concepts, relations and attributes. Using the transcript of interview in (Adesola, et. al., 2000) the following domain knowledge were 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, is that it is time consuming, one needs different scenarios for analysing different types of sentences.

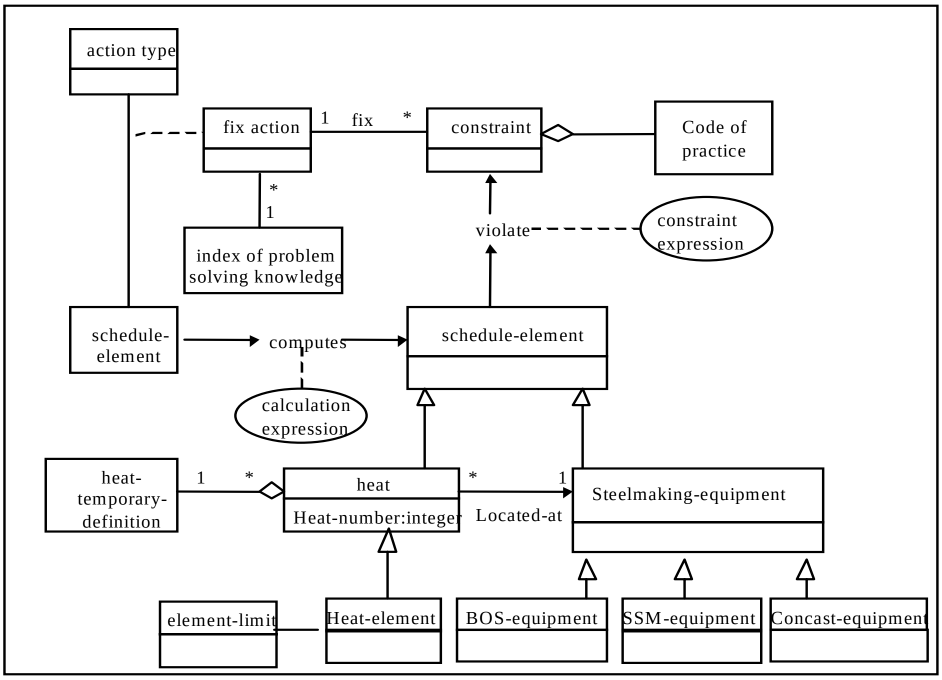

Figure 8: Typical domain knowledge types in MSD through monitor-construct-repair

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 component concerns what is happening and the entities involved in domain specific terminology, this was validated for accuracy. The second component describes the associated inference functions and knowledge roles[3] required, determining the actions; this was validated for completeness. The third component provides for 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 steelmaking 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

[3] Abstract names of data objects that indicate their role in the reasoning process.

easy to find domain knowledge that could act as static roles[4] for the inference structure. . Table 1 shows the three case studies for the knowledge model validation, there are two scenarios for each case study. A complete description of all three cases with the scenarios can be found in Adesola (2002). Appendix A presents a paper based simulation result for 'Steel OOS BOS Vessel-B (Scenario 1)' Case Study result as a sample.

Table 1: Case Studies for Knowledge Model Validation

Case studies	Description
A	Out-of-specification in the BOS Vessel
B	Out-of-specification in the Secondary Steelmaking (SSM) plant
C	Steel Temperature Too Hot BOS / SSM plant

Each scenario describes 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, 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 realises 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 experts were given: (a) the three case studies, (b) six paper simulation tables and (c) a set of semi-structured questionnaires. Each expert was required to study and cross check items (a) and (b) above. Each column of the paper simulation table is validated against the measurement criteria. For example, the domain data is validated for accuracy, knowledge model is validated for completeness and the explanation is validated for consistency. After cross checking, experts were asked to complete the validation

4 These are more or less stable over time, they specify collection of domain knowledge that is used to make inferences

questionnaire and comment on their overall experience. Table 2 presents the validation results.

Table 2: Knowledge Model Validation Results, a score of 4 means 'strongly agree' and 1 means 'strongly disagree'.

Case Studies	Scenario	Knowledge Model Simulation	Measurement Criteria	Expert A	Expert B	Expert C	Score E/12 x %
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	54	59	58
				=61/72	=54/72	=59/72	=58/72
				84%	75%	81%	80%

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 steelmaking 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 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 KBS may be seen as a modelling activity (Studer, 1998). A generic inference structure for disturbance management in steelmaking scheduling has been constructed. Many components have been identified with potential for reuse. The methodology applied in this research follows an academic approach, which pursues a framework for capturing human knowledge and conversion into a system for reuse combined with parallel validation through case studies in the industrial environment.

A methodology, called XPat, to capture process knowledge has been applied. The method aims to improve the natural knowledge elicitation technique by facilitating experts to express and display their expertise within a flexible and structured process. From the interpretation of interview transcript, ten categories of disturbances in steelmaking were elicited from experts. Eleven knowledge items were defined for each category of disturbances. These knowledge items constitute problem-solving knowledge for MSD in steelmaking. The knowledge items include implementation flowcharts for each PSK. This part of the research is prone to bias due to the interpretation by the researchers. The bias is minimised 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, combination of model elements can be reused. It is intuitively clear from the knowledge model in the research that large parts of model are not specific to steelmaking. Parts of the task and the inference knowledge can re-occur in other domain 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 to iteratively modify and extend the knowledge model. The case study has shown that the knowledge model has accurately captured problem solving knowledge and rules. The knowledge model is generic to steelmaking. Although the knowledge model has not been tested in other industries, it is expected that this can provide a basis for analysing management of schedule disturbance in other sectors. The knowledge model reflects expert reasoning process.

However as with any other research, the methodology has some limitations. The weakness of the methodology is its applicability to other domain, since time and resources limited the research; it has not been tested widely. With more time and resources the methodology should be tested in 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 to 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 XPat methodology can be utilised to develop a knowledge model for MSD in steelmaking. This proves the hypothesis of this research.

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

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 problem solving knowledge and rules. The knowledge model is generic to steelmaking. Although the knowledge model has not been tested in other industries, it is expected that this can provide a basis for analysing management of schedule disturbance in other sectors.

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

7. Acknowledgements

This research is being carried out as part of a Doctoral study at Cranfield University and is sponsored by EPSRC and Corus UK Ltd. Authors would like to thank the employees of Corus UK Ltd. for their help and support during the knowledge elicitation.

8. The Appendix: Case Study A

Paper based Simulation Result for Scenario 1, Steel Out of Specification (OOS) BOS at Vessel-B

Steel chemistry is OOS in BOS Vessel-B. The scheduler received the information via the operator at the BOS Vessel. The quality code is 1522, heat number in 2461. The customer will not accept OOS steel. The number of ladle in sequence is 8, heat position in sequence is 3, and the process route is VFD (Vessel Flush Degas). Figure 9 shows annotated inference structure for MSD with data about out-of-specification (OOS) steel in BOS Vessel-B.

Condition: Time is 16:45, required concast delivery time is 17:55.

There is time to contact the contact the customer.

Customer will not accept out-of-spec steel.

There is no quality that can be made from heat-need-outlet.

Action: **“**Create Heat-Needs-Outlet: Check the 24 Hour (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 a tabular form. The reasoning process, employed 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, that is, the dynamic roles[5], e.g. the system-data, the schedule, the discrepancies, etc has been instantiated with domain specific objects (Figure A.1).

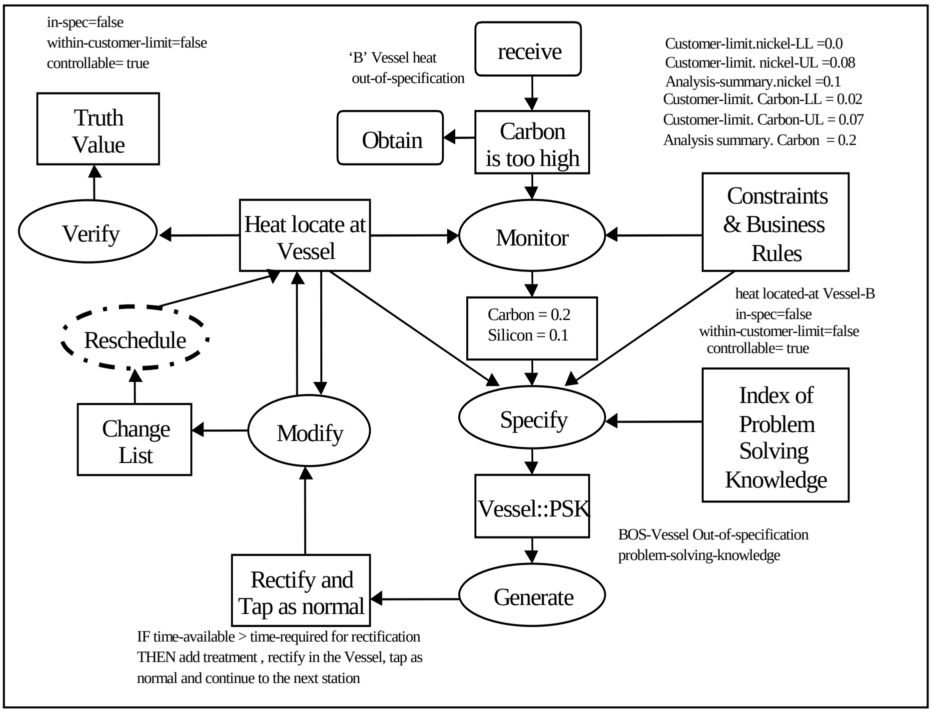

Figure A.1: An Annotated Inference structure for Steel Chemistry Out of Spec (OOS) Steel in BOS Vessel

The case study reflects required systems 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[6] for the inference structure. A paper trace in terms of the knowledge model constructs is generated.

---

[5] These are run-time inputs and outputs of inferences. The dynamic roles have different instantiations at each invocation.

[6] These are more or less stable over time, they specify collection of domain knowledge that is used to make inferences

<table>
<tr><th>Domain</th><th>Knowledge Model</th><th>Explanation</th></tr>
<tr><td>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</td><td>MANAGE SCHEDULE DISTURBANCE:
monitor: <u>system-data;</u>
Heat.location = Vessel-B</td><td>The scheduler monitors system-data, receive operator feedback for which manage schedule disturbance task is started. The heat located at Vessel-B is OOS</td></tr>
<tr><td>Obtain Additional Data

Scheduler: [Asked operator what element is OOS]
Operator: [The value of Carbon and Nickel is too high]
System: Vax Mgmt System – Heat Status and Analysis Summary (AS)</td><td>OBTAIN: <u>system data;</u>

Heat.carbon-element-analysis = false
Heat.nickel-element-analysis = false</td><td>Carbon and nickel are the two elements out of specification as the element analysis value indicate false.</td></tr>
<tr><td>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
<table>
<tr><th rowspan="2">Element</th><th rowspan="2">Analysis Summary</th><th colspan="2">Customer Limit</th></tr>
<tr><th>Lower Limit</th><th>Upper Limit</th></tr>
<tr><td>C</td><td>0.2</td><td>0.02</td><td>0.07</td></tr>
<tr><td>NI</td><td>0.1</td><td>0.0</td><td>0.08</td></tr>
</table>
Temperature: not affected</td><td>verify: <u>truth-value;</u>
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</td><td>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 code of practice specification for customer acceptance limits. In this case element OOS include controllable and uncontrollable elements: Carbon and Nickel exceeds customer limit.</td></tr>
<tr><td>Specify PSK

Scheduler: [Specify PSK]
Operator: [Awaiting instruction]
System: “Call Vessel out-of-specification psk”</td><td>specify: <u>fix:</u>
VESSEL::OOS-PROBLEM-SOLVING-KNOWLEDGE</td><td>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.</td></tr>
</table>

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]
	
	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 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
		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 indicates. For example depending on the state of the plant, the following questions are asked:
		
		Is there time available for rectification? Yes
		
		Although there is <u>no</u> time to rectify in the Vessel the element OOS is uncontrollable. 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

	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 minutes e.g. 10 minutes	
Display modification instruction Scheduler: [Advice shopfloor about possible actions] Operator: [Receive possible action] System: “Display problem statement, advice and explanation”	modify: **schedule**; **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 **Advice** “Create Heat Needs Outlet, find alternative quality and available caster for heat- needs-outlet. Tap heat as alternative quality”.	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.
Display rescheduling instructions	OBTAIN: **<u>rescheduling instructions;</u>** 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.

9. References

ADESOLA, B., 2002, *A knowledge model for decision support to manage schedule disturbance in steelmaking* (PhD Thesis, Cranfield University, UK).

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* (Springer Verlag).

BREUKER, J., WIELINGA, B., SOMEREN, M. V., DE HOOG, R., SCHREIBER, G., DE GREEF, P., BREDEWEG, B., WIELMAKER, J. BILLAUT, J., DAVOODI, M. AND S. HAYWARD., 1987, Model-driven knowledge acquisition: Interpretation models, Technical report, ESPRIT Project P1098, University of Amsterdam, Netherlands.

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

CHANDRASEKARAN, B. AND JOHNSON, T.R., 1993, Generic Tasks and Task Structures: History, Critique and New Directions. In David J.M., Krivine J.P., and Simmons R. (Eds.) *Second Generation Expert Systems* (Springer-Verlag, London), pp. 233-272.

COWLING, P. AND REZIG, W., 2000, Integration of continuous caster and hot strips mill planning for steel production. *Journal of Scheduling,* Vol. 3, Number 4.

DORN, J. AND SHAMS, R., 1991, An Expert System for Scheduling in a Steelmaking Plant. Proceedings of the 1st World Congress on Expert Systems, pp. 395-404.

DORN, J., 1995, Case-based reactive scheduling, In Roger Kerr and Elisabeth Szelke (eds) *Artificial Intelligence in Reactive Scheduling* (Chapman & Hall, London) pp.32-50 .

DORN, J. AND SHAMS, R., 1996, Scheduling High-grade Steelmaking. *IEEE Expert Special Issue on AI in Steelmaking*, 11 (1), pp. 28-35.

MOKA CONSORTIUM, 2001, Managing Engineering Knowledge. In Melody Stokes (Ed) MOKA: Methodology for Knowledge Based Engineering Applications. ISBN 1860582958.

NUMAO, M. AND MORISHITA, S., 1991, Cooperative Scheduling and its Application to Steelmaking Processes. *IEEE Transactions on Industrial Electronics*, Vol. 38, No. 2. pp. 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* (A Bradford Book, The MIT Press, Cambridge, Massachusetts).

STUDER, R., BENJAMINS, V.R. AND FENSEL, D., 1998, Knowledge engineering, principles and methods. *Data and Knowledge Engineering*, 25(1-2), pp.161—197.

TANSLEY, D.S.W. AND HAYBALL, C.C., 1993, *Knowledge Based Systems Analysis and Design: A KADS Developer's Handbook* (BCS Practitioners Series, Prentice Hall).

VESCOVI, M., IWASAKI, Y., FIKES, R, AND CHANDRASEKARAN, B., 1993, How things are intended to work: Capturing functional knowledge in device design. Proceedings of the 13th International Joint Conference on Artificial Intelligence.