CHAPTER THIRTEEN

Architecture and Interface Aspects of Scheduling Decision Support

Peter G. Higgins

13.1 INTRODUCTION

The gap between classical scheduling theory and actual industrial practice has been discussed throughout this book. When it comes to the complexity and uncertainty of many manufacturing environments, the scheduling activity addressed by classical researchers and the activities followed by industrial practitioners are disparate. Both constitute goal-directed allocation of resources over time to perform a collection of tasks; however, their specific goals and decision-making activities differ markedly. In classical theory, goals are simple quantitative measures (e.g., resource utilisation and tardiness) and scheduling activities are algorithmic procedures (e.g., heuristics such as SPT or sub-optimising methods, for instance, branch-and-bound).

In real plants, scheduling is a social activity; schedulers are subjected to the social dynamics operating within the organisation (see also Crawford in Chapter 5). Rich and assorted, intrinsic and extrinsic values and goals of the system of people of which they are part, affect them (Gault, 1984). Their behaviour is situated activity and is therefore embedded in the particular work environment (Bødker, 1991 and see Wäfler in Chapter 19). They have to make effective decisions in circumstances where the state of the system cannot be clearly predicted, information regarding jobs, materials and resources are ill-defined and scheduling goals are diverse. Scheduling under ‘perplexity’ rather than combinatorial complexity characterises many schedulers’ worlds, especially those working in small job shops. Perplexity, derived from the Latin word perplexus meaning involved, is an apt descriptor:

Perplexity: inability to determine what to think, or how to act, owing to the involved, intricate, or complicated conditions of circumstances, or of the matter to be dealt with, generally also involving mental perturbation and anxiety (Oxford English Dictionary).

Confusion and uncertainty are the hallmarks of perplexity (McKay, et al., 1990). Schedulers operating in the ‘real world’ of manufacturing must continually surmount environmental perplexity. As McKay discusses in another chapter, decision strategies are often complex and hinge on the schedulers’ awareness of the subtle relationships between factors that comes from an intimate knowledge of the plant, products, and processes. Indeed, McKay expresses amazement that
Schedulers make so few mistakes in coping with the amount of data and degree of complexity usually present.

In managing perplexity, schedulers do not use clearly defined, easily articulated, models of the scheduling process. Their knowledge develops over time as they continuously gather information as social beings (McKay, 1987). It does not stem from consciously elaborated principles or procedures, but is a matter of habit, experience and practice, which matures over years of experience in the work environment (Polanyi, 1962). This presents a basic problem for software development, as the automated construction of schedules requires all the constraints and rules to be prescribed. Instead of trying to elicit schedulers' knowledge to place in the software, developers of computer applications should give schedulers a leading role in the decision-making process. Then they can take advantage of the special abilities of humans (Sheridan, 1976);

- their ability to recognise patterns in the data, identifying what is, and what is not, essential from the current context (Sharit, 1984; Papantonopoulos, 1990);
- their ability to handle unexpected events and information that may be diverse, inexact, or conflicting (Higgins, 1996a);
- their ability to formulate general rules from specific cases (Sharit, 1984; Meister, 1966).

To tease out the form of a suitable methodology for designing tools that suit perplex environments, the chapter opens with a theoretical discussion on scheduling activity as purposive-rational action. It is argued that a methodology for developing scheduling tools must produce a decision architecture that encompasses the manufacturing system and the problem-solving operations of human decision-makers, without requiring complete knowledge of the scheduling domain. Constraint management is then used as a unifying concept for discussing the efficacy of prevailing scheduling tools and for expounding a new decision architecture and a new methodology for their development. Using data collected from a field study, the elements of a hybrid intelligent production scheduling system (HIPSS) based on the new architecture and methodology are then presented.

### 13.2 THEORETICAL FOUNDATIONS

While the consciousness of schedulers may form through shared activity with other people, their behaviour is purposeful as it is directed towards the accomplishment of specific tasks. The focus is a functional psychology relating perception and action. Schedulers are usually practical persons who have a keen understanding of their resources, the capabilities of machines and work practices. Their competence comes from a deep understanding of the work domain, evolved through experience gained under a variety of circumstances (; Dutton and Starbuck, 1971; Brödner, 1990 Higgins, 1999a; 1999b). They acquire much of their knowledge by solving pressing problems (De Montmollin and De Keyser, 1986). They learn to deal with a vast array of factors arising in the working environment, which may be unpredictable and are not easily placed into a theoretical scheduling context.

Gault (1984) puts forward a theory of Social OR Action that places purposeful action into the context of operations research. He focuses on the actions of people
in organisations inhabited by purposeful beings. Schedulers are subjected to the social dynamics operating within the organisation. Individual goals and relationships with other persons are subject to change. That is, schedulers in practice cannot isolate themselves from other activities within the organisation. They have to cope with complex systems of changing problems that interact with each other. Ackoff (1979a, 1979b) sees planning under these conditions as a process of the management of messes, not the solving of problems. He therefore sees it as pointless to seek optimal schedules. Scheduling becomes a process of synthesis more than analysis. Gault argues that scheduling support should help actors improve the quality of their actions as the complex array of consequences associated with any action includes many that schedulers experience subjectively. Merely focusing on improving the performance of a few quantifiable measures may be detrimental to the performance of the subjective dimensions, which may, in turn, be more important for organisational viability. For instance, the mix of challenging but stressful work and simple but undemanding work affects a skilled operator’s performance.

Having decided upon a goal to meet, schedulers can be understood to form intentions, make plans and carry out actions (Norman, 1986). Their intentions derive from their value structures and internal goals (Rasmussen, 1985). To know how to shape a scheduling system, in which humans actively partake in the making of decisions, we need to know how humans perceive the scheduling process. How do they interpret signals from the environment and work out appropriate actions? What mechanisms do they apply in generating descriptions of system purpose and form, explanations of system functioning and observed system states, and predictions of future system states (Rouse and Morris, 1986)? Obtaining answers to these questions depends on using formal methods for describing human decision-making processes in scheduling. A systems-oriented method of analysis is therefore required that encompasses both the manufacturing system and the problem-solving operations of the human decision-maker.

As it is difficult to observe in short field studies the full complexity and subtlety of the decision strategies of schedulers operating in dynamic environments, scheduling tools have to be able to be developed without thorough knowledge of all scheduling behaviours. A methodology is needed for developing tools that assist schedulers to follow strategies they find effective, without having to encompass the actual scheduling practices in the software. While the methodology must be capable of handling incomplete domain knowledge, the decision architecture of a scheduling tool should permit refinement as knowledge of the domain improves when the tool is used in the field. That is, the design of the decision architecture should support iterative refinement of domain knowledge.

13.3 DESIGNING THE HUMAN-COMPUTER INTERACTION PROCESS

Schedulers in practice require tools that help them to do their work more effectively. Primarily, a computer aid must support schedulers to apply accustomed methods developed through years of experience. The interaction process must cope with the ways they tackle scheduling problems. Using their intuition and
knowledge, schedulers should be able to guide searches for suitable schedules in directions they would like to follow. While not interfering with their perception of scheduling problems, the computer system should over time help them extend their knowledge of scheduling. These are all attributes that the 1991 SIGMAN (Special Interest Group in Manufacturing of the American Association for Artificial Intelligence) workshop on interactive scheduling identified as desirable (Kempf et al., 1991).

As scheduling is event-driven, the interaction process between human and computer must allow schedulers to experiment flexibly and opportunistically with different strategies (Woods, 1988; Woods and Roth, 1988; Sanderson, 1989). For example, the software must include functions for undoing and redoing all actions so users can manoeuvre freely in the decision space. Software development has to shift from a focus on OR models and solutions to the provision of decision support for domain experts, who are at the core of the decision-making process. The decision architecture is therefore designed so schedulers are active participants in schedule construction. Unlike conventional, interactive scheduling systems that present completed schedules to users, there is constant interactivity between human and computer as they share the decision activities, though under the control of the user.

When making decisions, schedulers usually go beyond the restricted set of information displayed by Gantt charts. Gantt charts by themselves are inadequate vehicles for displaying all the information that schedulers may want to use. A Gantt chart should be seen as the product of a decision making process; it is the plan for running the shop. While its form may well suit the exposition of a plan, it may not be suitable as the primary display for decision-making. The difference is communication as against discovery. A Gantt chart’s purpose is to communicate the plan of work to those people who have to use it. The imperative for communication is simplicity (Bertin, 1981). However, a display for decision-making must show comprehensively all data a scheduler may use to discover relationships, in a form that helps them to visualise abstract relationships between data. That is, it reveals relationships formed by the interplay of the data.

Realising the inadequacies of a standard Gantt chart as an interface for decision making, designers of interactive scheduling systems populate their screens with extra objects, either added to the Gantt chart or placed in additional windows or pop-up boxes. For example, users of LEKIN\(^1\) can see more details of a job by double clicking any of its operations. The operation becomes encircled, the due date and expected completion date of the job are marked on the time line, and a box appears showing the routing of all the operations for the job and its contribution to the performance measure.

A ‘hybrid intelligent’ production scheduling system (HIPSS) that brings human and computer ‘intelligence’ together in the process of decision making is

\(^1\) A flexible job-shop scheduling system developed for educational purposes by Pinedo and Chao (1999).
distinctly different to other interactive systems. Unlike other interactive systems, a HIPSS allows users a broad degree of action. It preserves the schedulers’ initiative to evaluate situations and to make decisions. The focus of the development of a HIPSS is on what the scheduler is doing with the computer and not what the computer is doing. In using a HIPSS, schedulers should feel engaged in scheduling activity, and not engaged in managing a computer.

To develop a ‘hybrid intelligent’ system requires a systematic approach to software development. The methodology for developing a HIPSS addresses decision making in complex systems in which there are many competing and conflicting goals. It:
1. has schedulers at the centre of the decision-making architecture;
2. uses Cognitive Work Analysis (CWA) to identify the information a HIPSS needs to display;
3. uses configural display elements that replace schedulers’ memory storage with external memory and supports schedulers to make visual inferences thereby reducing the need for any mediating inferential process.

13.3.1 Human-centred architecture

By locating humans centrally in the decision architecture, software can be designed that is not tightly bound to an overly restrictive perspective of the problem. The design emphasis moves away from including all the scheduling constraints and conditions within the software. The focus moves to the development of software that supports expert schedulers in applying their vast knowledge of the physical domain and its operational constraints to the decision-making process. Figure 1 gives one example of a human-centred architecture that allows schedulers to freely make decisions. They have full control over the application of the scheduling rules and can accept or reject advice at will from the knowledge-based adviser. Just as skilled artisans can freely wield their tools to accomplish purposeful actions, schedulers can employ a HIPSS as a cognitive tool. Schedulers can move freely within the decision space, seeking patterns within data, recognising familiar work situations, and exploring decision-making strategies under novel circumstances (Higgins, 1996a).

A human-centred architecture does not presume abandonment of classical OR methods, but instead accepts Morton and Pentico’s (1993) proposition that ‘All useful approaches should be pursued.’ The human can act as an intermediary between the real-world manufacturing environment and the abstract world of operations research, by:
• dealing with the stated and unstated conflicting goals;
• resolving how to use information that is incomplete, ambiguous, biased, outdated, or erroneous;
• grouping jobs to meet the specific criteria for applying selected heuristics.
In identifying human capabilities, Nakamura and Salvendy (1994) stress the proficiency of schedulers to adapt to changing production goals and priorities, to make decisions on realistic criteria and to consider multiple goals. Schedulers can also recognise conflicts between goals and decide how to resolve them. However, to be proficient users of a HIPSS as a cognitive tool, schedulers have to completely understand its functions and behaviour (Woods and Roth, 1988). In using a HIPSS as a tool, schedulers must be able to see the connection between their own intentions, or actions, and the effects produced by them.

**13.3.2 Cognitive work analysis**

The development of a HIPSS for a particular industrial setting depends upon an analysis of the sociotechnical system: the manufacturing system and the problem-solving operations of human decision-makers. Cognitive Work Analysis (CWA) provides the requisite methodology (Rasmussen, 1986; Sanderson, 1998; Vicente, 1999). It incorporates two different types of analysis: Work Domain Analysis (WDA) and Control Task Analysis (CTA). WDA uses an abstraction hierarchy (AH) - a generic framework for describing goal-oriented systems - to describe a system in a way that distinguishes its purposive and physical aspects. WDA is event independent and is quite separate from Control Task Analysis (CTA), which is a subsequent event dependent analysis of the activity that takes place within a work domain.
The basic features of CWA are most clearly demonstrated using a supervisory control example. While it is not a scheduling example, it is easy to explain the meaning of the AH using a system bound by physical laws. The AH in Figure 13.2 is a representation of the work domain for a heat exchanger, a component that is common in chemical processing and thermal power plants. The links show how to instantiate the design from the level of functional purpose to physical devices. The functional purpose of the heat exchanger is the efficient transfer of heat from one fluid to another. At the level of priorities or values are the intentional constraints: through the application of the physical principles of heat transfer, the designer places constraints on the behaviour of the system. The purpose-related function is the transference of heat between the primary and secondary circuits. Below this are the physical functions and devices. At the physical level, there are tubes, fluids and sensors. The sensors measure the appropriate state variables. The nodes and links are clear and unambiguous.

Control-room operators (i.e. the supervisory controllers) monitor the operation of the heat exchanger, troubleshoot, and intervene where necessary to modify its performance. Their activities are the subject of CTA. Just as the AH is a formal descriptor for WDA, Rasmussen’s (1986) decision ladder provides the rigour for representing each set of activities in CTA (Vicente and Rasmussen, 1990; Vicente, 1999). As a template of generic activities on which to overlay supervisory control activities, it acts as a prompt for identifying the information processing activities in the CTA (see Figure 13.2). It consists of a series of nodes, alternating between data processing activity and states of knowledge. States of knowledge are the outputs or
products of the information processing activities engaged in by a decision-maker. The decision ladder is not used to describe the decision process - the actual cognitive processes of the decision-maker - but to describe the products of information processing. For Vicente, ‘The goal is to find out what needs to be done, not how it can be done’.

In monitoring the operation of the heat exchanger, the operators’ activities may start at the bottom left of the ladder. Some event alerts them to monitor the condition of the heat exchanger. They scan all the instruments, observing the various temperature and flow rate values. Their ‘reading’ of the temperatures and flows depends on the instrument panel transforming the electrical signals from the thermocouples and flowmeters into information displayed at the level of physical function in the WDA. To identify the state of the system, they need to perceive the energy flows in and out of the heat exchanger. To do this they must know the thermodynamic relationships. The interpretation of imbalance of energy flows may be that the system is undergoing a thermal perturbation: it is not in equilibrium. If the goal state is ‘maintain equilibrium,’ the operators may need to define a task to bring the system into equilibrium: for instance, decrease the temperature of the primary circuit fluid exiting the heat exchanger. The procedural steps may include increasing the flowrate of the fluid in the secondary circuit and concomitantly decreasing its entry temperature.

As the decision ladder provides a systematic means for analysis of the operators’ decision activities, the chances of oversight are lessened. Even from this brief explanation of the decision ladder, it is clear that operators may benefit from a display that allows them to directly perceive energy flows. The procedural steps for the defined task may be automated if they are routine.

The decision activities do not have to use every node, climbing the ladder on the left and then descending on the right. Arcs can shunt across the ladder, reflecting short-cuts in the information processing activities. They are not in set positions but vary with the specific application. Consider the case of operators trouble-shooting a problem with the heat exchanger. They see an unusually low reading of the ‘temperature 2 out’ (see left side of Figure 13.2), while the other instruments are within their normal range. An operator who associates this pattern with past experience right away goes to the task ‘to check the thermocouple electrical circuit, (see right side of Figure 13.3).’ Before the task can be executed, the procedural steps must be drawn up. This stage may be bypassed, if the problem arises so frequently that the trouble-shooter can readily recall the procedure. The decision to check the electrical circuit depends upon previous experience of this type of problem. If the operators have not come across this situation before (or if on checking the electrical circuit it had been found to be functioning normally), their decision activities would return to the left side of the ladder and move upwards to more fully identify the state of the system. That is, decision making moves to observation at a higher level of abstraction: the flow of energy in the primary and secondary circuits.

A well-designed computer interface, therefore, must depict the different levels of abstraction shown in the WDA in a way that supports all the activities shown in the decision ladder.
In the process control industry, the WDA is founded on the physical laws that constrain the system: for the heat exchanger, these are the principles of fluid flow and heat transfer. In production scheduling, the WDA describes the properties of the system that are associated with the scheduling activities portrayed by the CTA. However, in describing a system from a scheduling perspective, at the level of priorities or values there are no physical laws to call upon.

13.4 APPLICATION OF COGNITIVE WORK ANALYSIS TO SCHEDULING

In job-shop scheduling, the WDA is based on a means-ends analysis of constraints in regard to meeting desired outcomes. The control activities of schedulers relate to the management of constraints. By choosing to place a particular job on a machine,
a scheduler places a temporal constraint on the machine. While the machine
deals with the job, it is unavailable for other jobs. Some constraints are physical,
set by physical laws, while others are intentional, set to meet the intentions of the
designers and users. From the schedulers’ perspective, a machine’s physical
constraints are rigid and they define its capabilities. Allocation of a job to one
machine in preference to another, where there are several similar machines, is an
act of intentionally setting the variable physical constraints on one machine at a
chosen time to the informational constraints defining the job. The setting of
constraints is the consequence of the scheduling activity of an agent, either human
or computer. The scheduling process is therefore the subject of CTA and not of
WDA.

To demonstrate the application of CWA to scheduling, Higgins (1992, 1999a)
used information he had collected from a field study of a printing company that
mainly produces continuous stationery: typical products are invoice and cheque
forms. In using his example, we will only discuss it here in sufficient detail to
elucidate the technique of CWA.

13.4.1 Work Domain Analysis

The printing company has two types of offset press - The Akira presses and the
Trident press, which are web and sheet-fed, respectively. Only the Akiras could
print fan-fold jobs, although any press could run the sheeted jobs. The Akiras,
however, being web presses must be fed paper in continuous form. After printing,
the paper is cut into sheets. Cutting may take place on the press itself or as a
separate operation on the Bowe cutter. The input to the Trident is stacked sheet,
which the shop obtains in sheet form or converts from continuous paper. The
Akiras vary in the number of colours they can print. Web presses operate at high
speed and can print on both sides of a continuous roll of paper. They generally
have in-line binding and cutting features. After impressing the image on the page, a
web press may either place perforations across the sheet to produce fan-fold forms
or cut the paper into individual sheets. On fan-fold forms made for track-fed
computer printers, the press also places perforations on each side of the sheet and
makes holes required by the sprockets. The final product is either fan-fold paper or
stacked sheet. Sheet-fed presses operate more slowly than web presses, but are
faster to set up (known as ‘make ready’ within the trade): accordingly, they suit
short production runs.

The four Akiras are alike, except for the number of colours - one, two, four
and six — that they can print and for the ancillary attachments that provide extra
functionality. Each colour requires a set of three cylinders: a plate cylinder, a
blanket or offset cylinder and an impression cylinder. A principal constraint in
offset printing is the cylinder size. The size of the impression cylinder depends
upon the required depth of paper for the job, as it must be an exact multiple of the
sheet’s depth. The ancillaries place additional constraints on the allocation of jobs
to presses. For instance, all jobs using ink that needs ultraviolet (UV) fixing must
go onto Akira 3, the six-colour press. For machine parts that are swapped between
machines, the configuration of one machine affects other machines. For example,
across the four Akira presses, only six colours can be printed concurrently on 297mm-depth sheet, as each colour requires a cylinder set and the cylinder size is an exact multiple of the sheet’s depth.

Where a job has multiple parts, they can run either consecutively or separately on one machine. Equally, the split can be between machines. A collator then joins the parts. Which collator is used depends upon the specifications for the job, as they differed in their capabilities.

The production process consists of the major physical functions shown in Figure 13.4. Finishing and special finishing can be broken down into separate procedures. The arcs show the order of operations. Figure 13.5 shows the major resources used to perform these functions. The closed and open arrows on the arcs show the routing for fan-fold and sheeted paper, respectively. The dashed arcs from Akiras 1, 2 and 4 signify that their links will be the same as those shown for Akira 3.
Figure 13.5 Major physical resources in printing.

Arrows in Figure 13.6 show the relations, from the ends to means, between the physical functions and physical resources. For simplicity, the four Akiras are lumped into a single generalised representation. Note that for some operations different resources can be used (e.g., folding, collating and finishing). In some cases, they are alternatives (e.g., in many cases it is immaterial which collator is used). For others it depends upon the specific operation, for example, the Hunkeler can die-cut the paper and glue transparent window on envelopes and Akira 3 is the only resource that can apply ultraviolet light to cure special inks. Arrows from the ‘Special Finish’ to the Hunkeler and the generalised Akira indicate these ends-means relations.

The ends-means relationships between physical functions and physical resources in Figure 13.6 form the levels of ‘physical function’ and ‘physical device’ in an AH for the WDA. For any particular job, the purpose-related level of the AH consists of the specification for the manufactured components. In other words, the job specifies the purpose of the system. The attributes of a job form the intentional constraints on the physical parameters (material, geometry, batch size, etc.) for the manufacture of the final product. The purpose-related function of the manufacturing system changes with the change in jobs. Hence, in production scheduling the WDA focuses on the configuration of the manufacturing system to meet the purpose-related function, that is, the production of a specified job.
13.4.2 Focusing on the Presses

To reduce the complexity of the scheduling problem for the sake of a manageable discussion, only the formation of an AH for the Akira presses is examined. While Figure 13.7 only shows the Akira presses, the AH for a full WDA for the manufacturing environment includes all the other physical resources as physical devices.
For each job, links from the set of constraints, that is, the specified attributes, to particular physical functions represent the feasible alternatives. All constraints on the purpose-related function that may map to constraints on the physical function must map to a single aggregated node for a link to exist. For example, if a job requires four colours, there must be four links from the front-of-bill (FOB) and back-of-bill (BOB) nodes at the purpose-related level to four-colour nodes within the aggregated physical function ‘print,’ which in turn map to four separate applicators within the ‘ink’ device as shown in Figure 13.7. To produce a four-colour job, there are only links to Akira 2 and Akira 3, as Akira 1 and Akira 2 do not have enough colour applicators. A job allocated to a machine has an ends-means chain instantiated between the purpose-related function and the machine as shown in Figure 13.9.
The hard technical constraints of the machines, which are causal constraints, project up to the purpose-related function level. Means-ends links only form where the causal constraints match the requirements of the purpose-related intentional constraints. For instance, the number of colours required for the job must be within the constraint boundary of the number of colours that the press is capable of producing. As the physical system can be ‘redesigned’ by changing the set up of the machines, intentional constraints at the purpose-related level set constraints on the physical device. For instance, a press’s cylinder size is changed to meet the constraints set by the depth of paper required by the job.

The functional purpose of production control at the company was to ‘maximise the long-term financial return’ of the company. To meet this purpose, manufacture was organised to ‘maximise short-term financial viability’ and ‘maximise repeat custom.’ These are shown at the priority/values level of abstraction. How links form between the nodes the priority/values level and the lower levels will be shown next.
As the configuration of the physical device must meet the purpose-related end, the AH is redesigned for each job as the purpose-related function changes. That is, each job has a different AH. The AHs for different jobs can therefore be considered metaphorically as a batch of cards. Each card shows the potential configurations for the processing of the batch defined by the job. Moving from one card to another denotes the changeover of jobs. If in changing from one job to another, the machine’s configuration must change, then the values of the constraints change between cards. The order of the cards sets the structural sequence of jobs (Figure 13.9): the temporal order of processing. Scheduling activity is therefore a process of ordering the cards and is therefore the subject of control task analysis (CTA).

13.4.3 Control task analysis

Control task analysis of scheduling behaviour identifies the processes in setting the intentional constraints identified by WDA. Sanderson (1991) was first to use the decision ladder for the control task analysis of the control structure for scheduling. She emphasises that the decision ladder provides a template for ‘a rational and complete analysis of the type of information in which humans must engage to solve scheduling problems,’ and as such it does not reflect the way that humans internally represent scheduling (if at all!).

Figure 13.10 shows a decision ladder for the decision-making activity associated with developing and maintaining a production schedule. The type of information processing varies with the position on the decision ladder. The arcs
link nodes that cycle between the recognition of a state of knowledge and information processing activities. Progressing up the ladder on the left, the application of knowledge characterises the decision behaviour in analysing the state of the manufacturing system. Having identified the consequences of the system state, decision behaviour moves down the right side of the ladder. Schedulers use their knowledge to plan scheduling actions, that is, actions affecting the intentional constraints. In contrast, arcs that shunt from left to right, near the bottom of the ladder, are associated with rule-based behaviour; that is, the recognition of a familiar pattern in the data triggers the application of an accustomed routine. Which arcs form depends upon the domain, the scheduler’s expertise and the presence of ‘stimuli’ for invoking particular behavioural responses.

Figure 13.10. The decision-making activity in developing and maintaining a production schedule.
In executing a procedure, schedulers search for key attributes among the data. For instance, to compose a schedule for a single machine using an EDD policy, they are likely to search all available jobs for the one that has the earliest due date. They would then most likely place it at the head of the queue. Again they would search the remaining jobs for the one that has the earliest due date. They would then place it next in the queue. They would repeat this procedure for the remaining jobs. The information extracted by schedulers from the data during the execution of the policy differs from the information used to select a suitable policy. A crude example demonstrates this. Schedulers are more likely to select EDD over SPT where the processing times for all jobs are similar but with the due dates broadly spread, than for the converse situation. In making this decision they link a ‘set of observations’ to a control ‘task’ as shown in Figure 13.10. The data display and representational form for supporting this activity may be different to that required for carrying out the procedural steps.

A greater challenge for schedulers is making decisions in circumstances where no known heuristics seem to apply. With no policies clearly pertinent, they try drawing some meaning from the ‘set of observations.’ Their identification of the present state of the system depends upon them making a connection between the data presented and their perception. By changing the presentation - by modifying, adapting or transforming the form in which data is presented - a mapping may emerge. If schedulers find a pattern in the data for which they know a policy that is efficacious, then they may enact this policy; on the decision ladder there would be an arc between the ‘system state’ and ‘task.’ If a scheduling policy is still indeterminate, schedulers may have to contemplate what ‘target state’ they are trying to satisfy (e.g., minimise average tardiness), and then define a heuristic that meets it (‘task’). Where they do not know what performance criterion to follow, decision-making moves to the higher knowledge-based level of performance criteria. If their behaviour were describable as pure rational action, then they would trial various functional performance criteria until they found one that suitably matches the performance goals.

It is clear from the decision ladder that decision-making by schedulers extends far beyond a focus on scheduling heuristics. Developers of software for decision support need to widen their purview from the automation of procedural steps, which concerns only the bottom right side of the ladder, to the other control tasks, such as helping schedulers identify which goal they should currently pursue.

13.4.4 Multiple decision ladders

Because schedulers seek different goals at different times, CTA of scheduling requires a series of connected decision ladders. Let us look how activities map onto multiple decision ladders. Assume that a scheduler, in observing insufficient jobs queued at a particular press, associates this state with a goal state, ‘extend the queue length,’ without referring explicitly to the ultimate goal, ‘no press idle time.’ The scheduler’s task is to add jobs to the queue by executing a procedure that minimises changes to the press set up. The elements of the decision are mapped on activity 1, the leftmost decision ladder shown in Figure 13.11. While executing this procedure, the scheduler is alert (activity 2) for adverse consequences. The
observation process is that of scanning patterns in job attributes. As long as the scheduler does not find any distinctive patterns in the data not associated with the current goal, the procedure for activity 1 continues. If, however, the scheduler identifies a pattern associated with a significant change to the system state, then activity 2 progresses and activity 1 halts. If the new system state includes tardy jobs for which meeting due date is critical, then decision activity leaps to the ultimate goal ‘meet the due dates of selected jobs.’ The goal state is redefined and the task becomes meeting major set-up constraints, but relaxing minor set-up constraints where necessary to allow inclusion of jobs that must meet their due date. If a known procedure exists, then activity leaps to its execution. During the execution of the new procedure, the scheduler again is on the alert for adverse effects. In scanning jobs allocated to other machines, the scheduler may, for argument sake, identify that the current scheduling procedure violates the set-up configuration across all parallel machines for the immediate processing of unannounced premium jobs. This triggers activity 3. To resolve the relative importance of competing goals, decision activity then moves to the top of the ladder. On figuring out appropriate performance criteria, the scheduler may then define a target state, associate a task to meet the target, and then carry out a procedure.

In Figure 13.11 switching between the different activities seems to be serial. But, scanning for a pattern that triggers a new activity occurs concurrently with the search for jobs that meet the constraints of the current procedure. Therefore, the left side of the decision ladder (activities dealing with analysis) along with the associative leap to the ‘ultimate goal’ (e.g., activity 2 in Figure 13.11) runs concurrently with the extant activity (e.g., activity 1). The right side of the ladder (planning and execution) is activated only when the scheduler decides to direct activity towards the new goal.

In the above example, the focus is on the utilisation of a particular machine. Schedulers also focus on jobs. For instance, as new jobs arrive, they may consider where to place them in the current schedule. In considering possible scheduling choices, their attention would be on the requirements of the jobs (job attributes) and relating them to the operational constraints (machine capabilities and current set-up configuration). This requires an understanding of the structural means-ends relations between the job attributes and constraints associated with the different physical devices.
13.4.5 Goal Structure links Control Task Analysis to Work Domain Analysis

The many goals that schedulers seek may seem unrelated. They are, however, bounded by rational action. A structural relationship between goals can be found by mapping the actual operational objectives to goals at higher levels of abstraction. In developing the structure, cognitive engineers need to find the principal goal of the company. The principal goal and the schedulers’ operational objectives are linked. Without any links there would be no effect on company performance by scheduling behaviour. This does not imply all operational objectives are the most effective means for attaining the principal goal. Between the operational objectives and the principal goal are intermediate goals. Some intermediate goals may clearly be substantiated from field data, whereas others may have to be inferred. Instead of having to extract all the rules of the domain, cognitive engineers only have to find a logical basis for structurally linking the goals.

The goal structure for the printing company is shown in Figure 13.12. At the lowest level of the hierarchy are goals A - the immediate operational objectives that directly guide the building of the schedule. These include low set-up time (2A) for each machine, particular jobs meeting their due dates (13A), priority given to favoured customers (15A), little change to a machine’s set-up between discounted jobs (9A), matching an operator’s ability to the work task (14A), and invoicing all jobs in the month the required raw materials are purchased (8A). Above this are goals B, to which schedulers may also directly attend; however, the extent of the focus varies between goals and schedulers. They are more abstract than operational objectives: for instance, ‘fully utilise all machines’ (1B) and ‘maximise return on
discounted jobs’ (5B). At the higher levels are goals that indirectly relate to shopfloor parameters. The focus is on short-term financial viability and customer patronage. At the top of the goal structure is the *raison d’être* of the company, the maximisation of long-term financial return. It depends upon short-term financial viability (1D) and customer patronage (2D). Short-term financial viability, in turn, depends upon productivity (1C) and cash flow (2C). By aiming to maximise customer satisfaction (3C), the goal of maximum repeat custom is achieved.

The higher a goal is up the hierarchy, the less directly it relates to immediate shopfloor activity. High-level goals tend to be attained through satisfaction of low-level goals, rather than by the direct attention of the scheduler. Nonetheless, schedulers sometimes directly considered them when making scheduling decisions. Directed arcs into a goal do not depict direct causation. Instead, they indicate a tendency of a goal to move with its underlying subgoals towards satisfaction. Underlying goals at times may move away from satisfaction, to improve performance at the higher level. For example, changing a machine’s configuration is an activity that is directed away from ‘low machine set-up time’ (goal 2A). This seemingly reduces ‘machine utilisation’ (goal 1B). Yet, if no jobs are available for the current configuration the machine becomes idle and consequently utilisation reduces. A dotted arc within a level of the hierarchy denotes that the linked goals are constrained by the relationship that ties them. For example ‘the quality of jobs is to the standard the customer expects’ (goal 9B) sets the upper bound on ‘maximise processing speed’ (goal 2B). Therefore, in pursuing goals that constrain other goals, experienced schedulers consider adverse effects on other parts of the goal structure.
Normally the ultimate goal in the decision ladders comes from the bottom level (i.e., goals A) of the goal structure in Figure 13.12. To realise a goal, an objective is set, a scheduling policy is defined and operational steps are executed; these are equivalent to the ultimate goal, target state, task and procedure, respectively, in the decision ladder. The relationship between a decision ladder, the goal structure and the AH is shown in Figure 13.13. In the relationship between the goal structure, decision ladder and abstraction hierarchy, the apex of the goal structure coincides with the functional purpose level of the AH, and the level immediately below coincides with the highest-level priorities in the AH.

The ultimate goal may often consist of a combination of goals shown in the goal structure. For instance, to satisfy combined goals of utilisation and tardiness, the procedure schedulers would follow would be distinctly different to the procedure for either goal by itself. As schedulers step through the procedure they scan the job tags for those that meet the desired constraints. For each job schedulers scan, they observe whether the feasible means-ends links (Figure 13.7) in the AH meet the current procedural requirements. From the subset that meets the requirements they select a job, that is, they instantiate a particular means-ends chain (Figure 13.8). During the scanning process they see patterns among job attributes that may trigger the consideration of other goals. In effect, they compare the AHs for many jobs, grouping them in a logical structural sequence (Figure 13.9).
13.5 DEVELOPING A HIPSS FROM CWA

The information on the scheduling constraints and decision activities, identified using CWA, forms the basis for developing a HIPSS (hybrid intelligent production scheduling system). A minimal form of a HIPSS merely displays the constraints identified in the AH: the constraints at the purpose-related level (job attributes) and the constraints at the physical function and device level (machine constraints). All control tasks in developing a schedule would be completely manual. By including feasible means-ends links shown in the AH, the computer shows which machines can process it. That is, the computer helps schedulers match the intentional constraints (job attributes) to the physical constraints (machine constraints).

The appropriate form of the data display varies with the type of decision being made. The presentation and control of information between the computer and human may vary with the recognition-action cycles in a decision ladder. On the left of the ladder, the decision behaviour is that of seeking dominant patterns in the data. The computer should support schedulers recognising current goals and procedures from patterns in the data. Observation and identification activities in the decision ladder are linked to the abstraction hierarchy of the work domain. Goal-relevant constraints from the work domain are mapped onto salient perceptual properties of the display: principles relevant to their design are covered by Ecological Interface Design (Vicente and Rasmussen 1990, 1992; Vicente, 1999), representation aiding (Woods, 1991) and configural displays (Bennett and Flach, 1992). Effortless extraction of information from the data depends upon the form of its representation. Each representation has its own set of constraints and intrinsic and extrinsic properties that emphasises some relationships and properties at the expense of others. Therefore, the form of data representation for showing the state variables at various levels of abstraction must be well grounded in theory.

Table 13.1 Legend for the labels in Figure 13.12.

<table>
<thead>
<tr>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Horizontal line links all the elements of a JSO into a unified image.</td>
</tr>
<tr>
<td>2</td>
<td>Horizontal position of the vertical bar signifies cylinder size. The height of the bar signifies paper depth.</td>
</tr>
<tr>
<td>3</td>
<td>The cylinder size is the same as label 2, but the depth of paper is a third.</td>
</tr>
<tr>
<td>4</td>
<td>Signifies an increase in width between jobs.</td>
</tr>
<tr>
<td>5 &amp; 6</td>
<td>The bars (labelled 5) map to the press icons labelled 6, with the current choice indicated by the tallest bar.</td>
</tr>
<tr>
<td>7</td>
<td>The length of the thick bar depicts the processing time.</td>
</tr>
<tr>
<td>8</td>
<td>The length of the thin bar depicts the set-up time.</td>
</tr>
<tr>
<td>9 &amp; 10</td>
<td>The horizontal position of the vertical line indicates which colour is to be printed.</td>
</tr>
<tr>
<td>11</td>
<td>The bar touching the baseline signifies that the colour is to be printed on the front of the bill, whereas the position shown for label 9 indicates the colour is to be printed on the back of the bill.</td>
</tr>
</tbody>
</table>
An example of a display for the printing case is the Jobs Window in Figure 13.12. It shows the detailed information for jobs queued at a particular machine as symbolic objects; Table 13.1 provides an explanation of the different elements. From the order of jobs in the Jobs Window the computer produces a Gantt chart (Figure 13.13). In the display each job is represented by a job specification object (JSO), which denotes the characteristics of the final product. It shows the intentional constraints associated with the purpose-related function, ‘process paper,’ (i.e., the job’s attributes). The features of a JSO are such that it:

- Clearly and distinctively shows the job attributes;
- Clearly displays, unambiguously, the values of the attributes;
- Supports the scanning of jobs to locate those jobs having a particular attribute value; and
- Clearly displays patterns in attributes across jobs.

Each element in the JSO signifies a particular job attribute and is visually quite distinctive, thereby enabling schedulers to clearly distinguish between elements depicting different attributes. Their design is guided by theories relating substitutive and additive scales and global and local features (Higgins, 1996a; 1996b). For each job, the element that refers to the same attribute has some features in common with the JSOs for all other jobs. These ‘Global’ features allow identification of all elements denoting the same attribute. As the global features for each element are distinctly different to the surrounding elements, the scheduler can easily identify the element that signifies a particular attribute. For each element, the scheduler can easily recognise the value of the referent attribute using ‘local’ features within the element that changes with the value. By scanning job attributes, at the purpose-related level of the AH, schedulers can observe both the patterns across jobs and also the system constraints (the current machine configurations).
To the right of each JSO is a sign (labelled 5) that shows possible means-ends links. The short, vertical lines indicate alternative machines that can produce the number of colours required for the job, whereas the large line shows the machine to which the job has been allocated (the means-ends link currently instantiated). Schedulers can perceive the work domain at different levels of abstraction. At the purpose-related function level, the HIPSS shows the constraints that specify the function, ‘process paper’, in the abstraction hierarchy (AH) in Figure 13.7. The list of objects, in effect, forms a sequence of abstraction hierarchies in Figure 13.9. The Jobs-Window display in the HIPSS is equivalent to a plan view of a pack of abstraction hierarchies. Peering downwards, metaphorically, the user can see the work domain constraints for each card in the pack. For each job, the JSO provides affordances to the different levels of the associated AH. Figure 13.14 shows some of the links between the screen objects and the AH. Curved broken lines depict the association between the constraints in the JSO and the AH. Users can also observe how the intentional constraints at the purpose-related level place constraints on the physical devices. The cylinder element of the JSO shows the depth - an intentional
constraint set by the job specification - and the cylinder size, which is a configuration requirement for the press. Solid arcs having arrow ends show the link between the depiction in the Jobs Window and the constraint on the physical device.

![Diagram](image)

**Figure 13.14.** The relationship between the abstraction hierarchy and the signs in the Jobs Windows in the HIPSS for printing.

Set-up time is not derived from the mapping of the intentional constraints at the purpose-related function level to the physical-device level of the AH. Instead, it is a product of the juxtaposition of AHs. In juxtaposing JSOs, differences in the local features of the elements signify a configuration change in moving from one AH to another. A change from one cylinder size to another is a major set-up, which is clearly observable, as labels 2 and 3 in Figure 13.15 show. A graphic object (labelled 1) signifies a minor set-up occurring when the width of paper decreases from one job to another.
Figure 13.15  Set-up time is a factor that arises when the value of particular constraints differs between abstraction hierarchies.

To move the problem solution closer to meeting a set of goals, schedulers shuffle the job objects in the Job Window until a satisfactory pattern is obtained. For example, to maximise the number of jobs before a cylinder change, jobs would be rearranged so that the vertical line through the cylinder element of the JSOs would be unbroken.

Human Scheduler  Computer
Figure 13.16 Human scheduler recognising relevant policy and computer carries out the procedure.

In familiar situations, behaviour is rule-based as schedulers apply familiar scheduling heuristics. The HIPSS can help schedulers execute rule-based procedures, shown on the right side of the ladder. As they group jobs, the computer shows which machines the string can be allocated (Figure 13.17). Where possible the steps of procedures can be automated (Figure 13.16). The scheduler first selects a string of jobs and then chooses a heuristic to apply.

Figure 13.17. As a string of jobs is collected, the collected objects become shaded (labelled A) and the permissible presses become shaded (labelled C). The number of spokes on the blue ‘collector’ (labelled B) shows how many have been collected. When the ‘collector’ is right clicked, the job numbers are displayed in a pop-up box, listed in the sequential order of collection.

13.5.1 Schedulers at the centre

By designing a HIPSS using CWA, it is possible to position schedulers centrally in the decision-making architecture as depicted in Figure 1. In the case of the HIPSS for printing, schedulers use the Jobs Windows (see Figure 13.12 for a detailed view of a window) to choose classification strategies for forming groups and then place them into machine queues. They can reflect upon the characteristics, that is, the attributes, of jobs and the calls that these make upon the shop. The job attributes, and patterns among attributes across jobs, act as stimuli. Using information that is displayed and other domain knowledge, they can seek patterns in the data on which to draw inferences about possible scheduling strategies. In an opportunistic way, they can try various groupings, make amendments and backtrack on previous decisions. They can select sets of jobs to move as one within a queue or to another machine. In ordering the jobs within a set, they can apply various OR heuristics.

The interface is designed so users can attend to constraints at times appropriate to them. The HIPSS is designed so that it does not needlessly get in the way of decision-making. Users can explore potential decisions in their own way, guided
by their understanding of the scheduling process. The HIPSS passes messages unobtrusively to users. For example, the black bars denoting constraints on permissible presses (label 5, Figure 13.12) are always visible and can be considered at any time. For soft constraints graphical signs have particularly efficacy. The graphic object (labelled 1 and 4 in Figure 13.15 and Figure 13.12, respectively) denotes a soft constraint that occurs when a job has a lower value of width than its immediate predecessor and the predecessor uses the same cylinder. The message draws attention while not being disruptive. While the knowledge-based adviser warns the user when soft constraints are infringed, it disallows violation of hard constraints (see label 4 in Figure 13.12). Of course, if a user attempts to violate hard constraints the activities of the HIPSS will not remain in the background. It then sends an intrusive warning through a pop-up message box. For example, as the string of jobs shown in Figure 13.17 is collected, permissible presses are shaded. This message is non disruptive. However, if the scheduler attempts to allocate the string to an inappropriate press (the unshaded press 4), then the computer pops up a warning in a standard message box, to which the scheduler is compelled to attend.

13.5.2 Improving scheduling performance

Although schedulers using a HIPSS may follow familiar practices, the HIPSS should be designed to encourage schedulers to improve scheduling behaviour. Often they are preoccupied with immediate operational objectives (e.g., low press set-up time across the shift), which are at the lowest level of the goal structure (Figure 13.18). Occasionally, their attention may be drawn to high-level goals. If their focus could be generally raised to higher-level goals, then the scheduling performance relating to these goals may improve. Goals at the level above the immediate operational objectives tend to be directed towards functional goals. Some goals at this level are commensurate with traditional OR goals. For the printing example, percentage utilisation is a suitable measure of performance for the goal ‘fully utilise all machines’ (1B). Similarly, average tardiness is a suitable measure for the goal ‘all jobs delivered on their due date’ (8B).
Schedulers using the HIPSS in Figure 13.13 can track the average tardiness at each press (labelled 12 in Figure 13.12) and across all presses (in the window for unallocated jobs in Figure 13.13). Inclusion of measures for other functional goals would further improve the HIPSS. The HIPSS gives schedulers the opportunity to seek ultimate goals that encompass those functional goals for which performance is displayed. In formulating a procedure (the lower right side of the decision ladder) to move a schedule towards an ultimate goal, schedulers must be able to experiment with various strategies. This HIPSS does not show which jobs to move, and where they should move, for performance to improve. The Jobs Window in Figure 13.19 is for a HIPSS that addresses this issue. This HIPSS was designed for a system having only a few attributes (release date, due date and processing time). Each job’s tardiness, priority weight and weighted tardiness are shown graphically. (Refer to Table 13.2 for a description of the labels). Schedulers can see where jobs need to move in the queue for due date to be met. Placing the mouse cursor on job 03 causes a bar to appear between jobs 11 and 06. This bar shows where the job has to move for it to be finished on time. A similar bar shows the location on the Gantt chart.

There may be various arrangements that would give the same value of overall tardiness. By observing the distribution of tardiness schedulers can adjudge other aspects of performance qualitatively. From experience, they may, for example, prefer a balanced distribution of tardiness. Or perhaps they may prefer to concentrate the tardiness in only a few jobs. Having established the immediate objective, they can try out various strategies for meeting it.
Table 13.2 Legend for the labels in Figure 13.19.

<table>
<thead>
<tr>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transparent rectangle signifies a non-tardy job</td>
</tr>
<tr>
<td>2</td>
<td>The length of the horizontal line shows a non-tardy job’s earliness</td>
</tr>
<tr>
<td>3</td>
<td>The width of the rectangle shows the job’s contribution to the performance measure.</td>
</tr>
<tr>
<td>4</td>
<td>The height of the rectangle shows the weight used for the job.</td>
</tr>
<tr>
<td>5</td>
<td>The horizontal line shows the unweighted measure: tardiness for this case.</td>
</tr>
<tr>
<td>6</td>
<td>The bar shows where a job indicated by the cursor for the mouse (labelled 7) has to be placed so that the job will not be tardy.</td>
</tr>
<tr>
<td>7</td>
<td>The cursor for the mouse. Note that job 03 is tardy.</td>
</tr>
<tr>
<td>8</td>
<td>The performance measure for the machine</td>
</tr>
</tbody>
</table>

Figure 13.19 The HIPSS Jobs Window for a simple scheduling model.

13.6 CONCLUSION

For those production environments that must rely on the knowledge and skills of human schedulers, scheduling software should support their decision-making. A hybrid intelligent human-computer production scheduling (HIPSS) paradigm that supports human schedulers operating in environments characterised by uncertainty and instability has been advanced. Its contribution to scheduling practice is a methodology for addressing decision making in complex systems in which there are many competing and conflicting goals.

The decision architecture of a HIPSS locates schedulers at the centre of the decision-making process, where they can act as intermediaries between the real-world manufacturing environment and the abstract world of operations research, by:

- dealing with the stated and unstated conflicting goals;
- resolving how to use information that is incomplete, ambiguous, biased, outdated, or erroneous;
• grouping jobs to meet the specific criteria for applying selected heuristics.

Just as in the process industry, where the operator in the control room has instruments for displaying the state variables and performance measures of the plant and has alarms to warn of critical constraint violations, a HIPSS provides an environment for schedule control, with features for showing the state of the schedule, for indicating performance and for warning the violation of constraints. As the understanding of scheduling factors in the domain is improved, new indicators and automated procedures, which are under the command of the schedulers, may be incorporated in a hybrid intelligent production scheduling system. By being actively involved in decision-making, the human can deal with contingencies and other aspects of scheduling jobs that are difficult to vest in a computer decision-maker. The use of intelligent human decision-makers with vast local knowledge also obviates the need for an exhaustive knowledge base.

For humans to play a coherent and active role in schedule construction, they must have ready access to all the information they use to make decisions. Cognitive Work Analysis (CWA) provides a useful formalism for designing a HIPSS. Work Domain Analysis (WDA) identifies the system constraints that should, where possible, be displayed by the computer. Using Control Task Analysis (CTA) cognitive engineers can construct decision ladders associated with the different scheduling goals. The details in the goal structure and the decision ladders vary between schedulers as the particular problem-solving technique a person applies depends upon experiential familiarity with the task. Therefore, it is important to design a HIPSS that has many degrees of freedom to allow for wide-ranging problem-solving strategies. Where cognitive engineers can construct decision ladders for prototypical behaviours, then there is the potential for determining the states of knowledge and control tasks that schedulers may use in making decisions. Cognitive engineers can then consider how best to represent the states of knowledge (e.g., patterns in data, system constraints, goal states and measurement of goal performance) and to automate control tasks (e.g., procedures for ordering jobs). Where possible, information is displayed using graphical elements that are configured by the value of the parameters they represent. The graphics are designed so there are visual cues to significant factors and visual pattern recognition replaces pattern-matching activity in the schedulers’ memory.

The extent of the analysis and the scope of the design for a HIPSS may vary with the domain and the degree of support sought. A minimal HIPSS replicates the constraints in the WDA. The computer displays the job attributes and machine attributes and warns when there is violation of technical constraints. All the control tasks associated with the decision ladders are undertaken by the schedulers. For relatively simple and fixed manufacturing systems, the path between subgoals and the control tasks for each subgoal may be well understood. Under these circumstances, it may be easy to automate decision processes. More complex environments may depend upon human schedulers identifying the relevant goals from patterns in the data and then recognising which scheduling procedure is applicable: frequent procedures may be automated.

As HIPSS may be developed in stages, ‘hasten slowly’ may be the appropriate motto!
13.7 REFERENCES


