Human Factors and Ergonomics in Manufacturing, Vol. 18 (6) 603–620 (2008)

Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/hfm.20129

The Role of Algorithm and Result Comprehensibility of Automated Scheduling on Complacency

Julien Cegarra
CLLE, University of Toulouse, France

Jean-Michel Hoc
CNRS-University of Nantes, Nantes, France

ABSTRACT

Several studies have stressed that even expert operators who are aware of a machine's limits could adopt its proposals without questioning them (i.e., the complacency phenomenon). In production scheduling for manufacturing, this is a significant problem, as it is often suggested that the machine be allowed to build the production schedule, confining the human role to that of rescheduling. This article evaluates the characteristics of scheduling algorithms on human rescheduling performance, the quality of which was related to complacency. It is suggested that scheduling algorithms be characterized as having result comprehensibility (the result respects the scheduler's expectations in terms of the discourse rules of the information display) or algorithm comprehensibility (the complexity of the algorithm hides some important constraints). The findings stress, on the one hand, that result comprehensibility is necessary to achieve good production performance and to limit complacency. On the other hand, algorithm comprehensibility leads to poor performance due to the very high cost of understanding the algorithm. © 2008 Wiley Periodicals, Inc.

1. INTRODUCTION

In the design of human–machine cooperation, several authors have noted that expert operators, even aware of a machine's limits, could adopt its proposals without questioning them. This failure has been termed *complacency* (Hoc, 2001; Parasuraman, Molloy, & Singh, 1993; Smith, McCoy, & Layton, 1997) and has been mainly studied in supervision tasks that imply automation or computers (for example, car driving: Hoc et al., 2006; airline pilots: Layton, Smith, & McCoy, 1994; air traffic control: Metzger & Parasuraman, 2001). However, other tasks that are only partially automated, such as those found in manufacturing scheduling, are also affected by this problem. Manufacturing scheduling can be defined as the elaboration of a plan for resources (machines and human operators) based on the organization within a set period of time of the realization of production jobs, taking into account temporal constraints (waiting periods, precedence, etc.) and constraints related to the use and availability of the necessary resources (Lopez & Roubellat, 2007). Scheduling is, on the one hand, considered to be a complex task and a well-defined problem. On the other hand,

Correspondence to: Julien Cegarra, Centre Universitaire Jean-François Champollion, Place de Verdun, 81012 Albi cedex 9, France. Phone: +33-5-63-48-64-02; e-mail: Julien.Cegarra@univ-jfc.fr

many disturbances can affect the validity of a schedule and may imply rescheduling. Thus, different authors have suggested that scheduling should be allocated to a computer due to the combinatorial requirements of the problem. Likewise, rescheduling should be allocated to a human operator because this requires certain skills, such as the negotiation of due dates with customers. This allocation is labeled *hybrid* and allows production performances to be better than those achieved by the human or the machine alone (Chen & Hwang, 1997; Haider, Moodie, & Buck, 1981; Sanderson, 1989).

However, this method of allocation could lead to a complacency failure as the human operator could accept computer-generated schedules even if they considered the result to be suboptimal from their point of view. In this article we will present the importance of the complacency problem in scheduling (in relation with production performance). We will also suggest different factors to reduce this failure by manipulating diverse algorithm (schedule generation) characteristics. Two characteristics will be detailed: algorithm comprehensibility and result comprehensibility. An experiment has been designed to evaluate these two factors with regard to human rescheduling performance. Finally, we will discuss prospects for scheduling in terms of human–machine cooperation.

2. COMPLACENCY IN HUMAN–MACHINE SCHEDULING

2.1. Definition of Complacency

For some time, a number of difficulties in the relationship between humans and machines have been stressed. A review of these can be found in Wiener and Curry (1980), Parasuraman and Riley (1997), and Hoc (2000). In the main, machines have been described as leading to a loss of expertise on the user side, as well as to a loss of adaptability, complacency, and miscalibrated trust. The loss of expertise is related to the lack of practice when a function is automated and the loss of adaptability to the lack of feedback returned to the human. This article will primarily focus on complacency and secondarily on trust.

Like other failures in human–machine cooperation, the complacency phenomenon has been set within the context of attention and vigilance in supervision tasks. It has been described as an unjustified assumption of satisfaction in the situation faced, although some improvements could apply (Layton et al., 1994; Parasuraman et al., 1993; Smith et al., 1997). This phenomenon is related to a low level of suspicion and is often correlated with overtrust in automation. Most automation complacency studies have studied complacent behavior in terms of performance outcomes (e.g., detection of automation failures) and assumed that this reflects poor monitoring of the raw information sources. Moray (2003) noted the need for measuring not only performance but also the rate of inspection of the information sources. In this way, Bahner, Huper, and Manzey (2006) identified the behavioral consequence of complacency as a sampling rate below the optimal rate, implying an attention defect.

The reasons for complacency can be various, but we consider they mainly rely on the management of a balance between the human investment in terms of cognitive costs and the results obtained in terms of performance. This balance leads to a satisfactory human performance rather than to an optimal performance (Hoc & Amalberti, 2007). When a function is delegated or allocated to a machine in a multitask situation, attention is shifted to those tasks that are not automated. Thus, the human operator neglects the information necessary to perform the automated function, does not supervise the function, and does not try to improve its results, even if it is possible.

2.2. The Specific Case of Scheduling

Complacency has been previously described in the context of attention and vigilance, where the human operator is considered as a supervisor of a machine who has to detect automation failures. In scheduling, there are two main differences with this traditional view: (a) the human scheduler is not only a passive monitor of automation but is also greatly involved in its correction; and (b) not all automation errors are crucial and a human is necessary to modify computer-generated schedules in order to introduce some flexibility.

These differences deserve an example illustrating the human scheduler's involvement in evaluating and correcting computer-generated schedules. In scheduling situations, many data are not easily accessible to computer scheduling tools. For example, such influences as weather can bring about a change in production, so that even a few degrees difference in temperature can require a decreased machine cadence. For this reason, Jüngen and Kowalczyk (1995) noted that the good performance of a scheduling algorithm (in relation to its model of the process) does not automatically imply its use by the human scheduler, especially because some decisions could be contradictory to the human point of view:

> When the resulting set of constraints is finally satisfied it often happens that the human rejects the solution because it does not satisfy some (implicit) requirements (e.g. the expert might find the generated schedule to be too vulnerable to bad weather). On the other hand, a schedule that violates some constraints (e.g. the duration of the whole project is slightly exceeded) can be accepted as a good one. (p. 67)

In relation with these two differences, it is possible to define complacency in scheduling as an unjustified assumption of satisfaction in which a human accepts a suboptimal (production) performance because of the cognitive cost of evaluating or correcting the machine's proposal.

Moreover, the traditionally suggested workaround of complacency does not apply very well to scheduling situations. Smith et al. (1997) studied the use of replanning software capable of proposing new flight plans in order to avoid stormy weather. They suggested that complacency could be prevented by offering the human operators (in their case, airline pilots) several alternatives (instead of only one) from among which to choose. However, this suggestion can lead to several problems in scheduling:

- Human operators may face cognitive limits when different alternatives have to be compared. In this case, Sanderson (1989) stressed that operators are required to explore and to memorize an important number of data, leading to a high cognitive workload.
- The calculation of different solutions may require a very long computing time. If a change to a schedule requires a long time, this may increase the cost of interacting with the machine. In this way, Valax and Cellier (1992) noted that the more a scheduling computer tool acts as a constraint (e.g., requiring long computing times for each update), the less operators supply small but costly changes in production to this tool (and these are not integrated into the calculations).

For these reasons, trying to prevent the complacency problem by increasing the cost of the interaction (by displaying several alternatives) is probably not the correct approach. Furthermore, instead of suggesting different solutions, which lead to an increase in the cost of interacting with the machine, it may be possible to use an algorithm that generates only one relevant solution (from the human operator's point of view). In this case, one should attempt to minimize the cognitive costs of the schedule evaluation and modification to limit the complacency phenomenon.

3. CHARACTERIZING ALGORITHMS FROM THE HUMAN POINT OF VIEW

In the scheduling literature, there are diverse classifications of algorithms. For example, it is possible to distinguish between exact algorithms and approximate algorithms. However, such a classification is not really relevant from the human point of view as humans do not represent complexity in the same way as formal models (Dessouky, Moray, & Kijowski, 1995). To our knowledge, no classification of scheduling algorithms using dimensions relevant from the human point of view has been elaborated. Therefore, we suggest considering two main levels of comprehensibility in scheduling: result comprehensibility (the schedule itself, especially through its graphical representation) and algorithm comprehensibility (the way the algorithm elaborates the schedule).

3.1. Result Comprehensibility

Result comprehensibility involves facilitating the evaluation of computer-generated schedules by grouping jobs in the Gantt chart, thus supporting schedulers' visual cues.

Different graphical representations can highlight different constraints. In a classical Gantt chart, each job uses the same color, as illustrated in Figure 1. This allows the schedulers to immediately detect if precedence constraints are satisfied, as a job cannot start before the previous machine's job has been completed.

Gibson and Laios (1978), in the only existing experimental study to compare graphical representations of scheduling problems, suggested the evaluation of this representation with others as a form of *modified machine planning* (cf. Figure 2). In this representation, the

Figure 1 The machine planning board. Bars of a length proportional to the required processing time represent job operations. Different colors indicate different jobs.

Figure 2 The modified machine planning board. Bars of a length proportional to the required processing time represent job operations. Different colors indicate different machines.

colors indicate the machine and not the job, as in the classical Gantt chart. Gibson and Laios' results stressed that this modified version allowed machine utilization to be increased as a machine's availability is more easily detected.

However, they did not take fully into account the problem of precedence constraints relative to flow shop scheduling. In a previous experiment, we have shown that visually grouping the jobs, using colors for successive jobs in the Gantt chart, as in Figure 1, is a support for human scheduler strategies (Cegarra & Hoc, 2003). The reason is that it facilitates the evaluation of the satisfaction of precedence constraints by allowing the direct perception of jobs in groups or chunks.

Moray (2003) has suggested that complacent behavior relates to an inadequate monitoring of the automation. In supporting schedulers' visual cues, result comprehensibility could facilitate the evaluation of the computer-generated schedule. In this way, it could allow better performance by facilitating the monitoring of the automation and reducing complacency.

3.2. Algorithm Comprehensibility

Algorithm comprehensibility involves limiting the algorithm procedure to only one step, thus supporting schedulers' understanding of the algorithm procedure.

In a computer-generated schedule, the scheduling algorithm satisfies many constraints. The graphical representation only highlights some of these constraints; for example, there is no information about satisfied due dates, or late jobs, in Figure 1. So to prevent human schedulers having to check all constraints manually, it could be possible to train them to understand this algorithm to easily know which constraints are satisfied in the final schedule. This was also noted by McKay and Wiers (2001, p. 173), who considered that: "The schedulers need to know immediately from the Gantt chart why jobs are where they are and previous experience with sophisticated algorithms indicated that human schedulers must be able to easily understand a generated schedule and it cannot be magic and mirrors." This implies that one should resort to simple algorithms to allow human schedulers to understand them (Wiers & van der Schaaf, 1996). This could allow schedulers to monitor and to correct the automation more efficiently and finally to decrease complacency.

Conversely, algorithm comprehensibility could also suggest another conclusion. Taking into consideration a study by Moray, Dessouky, Kijowski, and Adapathya (1991), algorithm comprehensibility could also imply a very high cognitive workload. Then, as cognitive resources are limited, schedulers could invest fewer cognitive resources in the rescheduling task than when the algorithm is not comprehensible. In this way, Davis and Kottemann (1995) noted that when describing the scheduling rule, participants required more time to resolve problems than when they did not know this rule. Finally, this could lead to complacent behavior: the cognitive cost required to modify the schedule will be very high and schedulers will accept a lower level of performance than they could obtain otherwise. This is in line with a study by Bi and Salvendy (1994), which showed that a high workload is usually associated with a low level of performance of scheduling.

It has been suggested that to increase result comprehensibility with regard to the computer-generated schedule leads to decrease complacency. In the case of algorithm comprehensibility, there are two contradictory points of view: (a) this type of comprehensibility could facilitate the monitoring of automation and decrease complacent behavior; and (b) this comprehensibility could lead to a high cognitive workload and, in the end, imply more complacent behavior.

Our experimental study aims to test these two characteristics (algorithm and result comprehensibility) and two explanations of algorithm comprehensibility.

4. EXPERIMENTAL STUDY

4.1. Algorithms

Two main characteristics of algorithms have been suggested: result comprehensibility and algorithm comprehensibility. In this experiment, three algorithms have been selected, as shown in Table 1, to test these characteristics by comparing algorithms in pairs. These three algorithms are detailed in the next section. To identify complacent behavior it is necessary to compare performances with a standard; in other words, the best performance that operators can achieve. In scheduling, Moray et al. (1991) demonstrated that, from a mathematical point of view, humans are not able to attain optimal performance. For this reason, algorithms are compared in pairs, one having a characteristic that could lead to complacent behavior and the other able to determine the maximum human-performance obtainable. As we will see later, by comparing Earliest Due Date and Moore–Hodgson—both algorithms having the same level of result comprehensibility—it is possible to evaluate the role of algorithm comprehensibility. And by comparing Moore–Hodgson and Earliest Due Date + Shortest Processing Time—both having the same level of algorithm comprehensibility—it is possible to evaluate the role of result comprehensibility.

In Table 1, a comprehensible algorithm that does not have a comprehensible result has not been considered. The reason is that such an algorithm is difficult to design as the incomprehensibility of the result requires a complex algorithm; as such, it implies that an algorithm is not comprehensible. So, only the three algorithms shown in Table 1 have been selected. They are detailed in the following sections.

4.1.1. Earliest Due Date (EDD). The earliest due date is a simple algorithm that consists in producing jobs in the order of their planned due date, from the earliest through to the latest (cf. Figure 3). This algorithm is not very efficient when the number of late jobs needs to be minimized. As can be noted in Figure 3, a late job with an early due date can also produce a very high number of late jobs if there is a high degree of tightness (in other words, too many jobs to be produced in too little time). So, this algorithm is not complex enough to solve day-to-day scheduling. For this reason, several experimental studies that compare

TABLE 1. Selected Algorithms in Relation to the Presence or Absence of Two Characteristics of the Algorithm (Algorithm and Result Comprehensibility)

	Algorithm's Characteristics	
Algorithms	Result Comprehensibility	Algorithm Comprehensibility
Earliest Due Date (EDD)	+	+
Moore–Hodgson (MH)	+	–
Earliest Due Date + Shortest Processing Time (EDD+SPT)	–	–

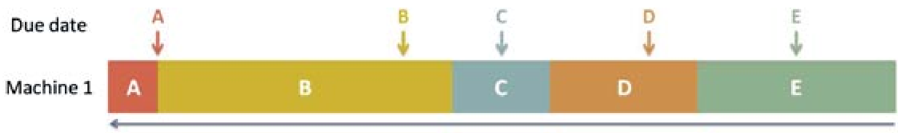

Figure 3 The earliest due date (EDD) algorithm: from the earliest to the latest due date.

the performance of scheduling algorithms with that of human schedulers have shown that the human generally outperforms a simple algorithm such as EDD (Nakamura & Salvendy, 1988; Tabe, Yamamuro, & Salvendy, 1990). Moreover, as Green and Appel (1981) noted, schedulers themselves consider these algorithms to be too simple to solve most scheduling problems. This also highlights that it is possible to consider the EDD algorithm as having good algorithm comprehensibility as it is relatively simple for the schedulers (only one step).

4.1.2. Moore–Hodgson Algorithm (MH). In comparison to the EDD algorithm, the MH algorithm is a combination of several rules. It consists, at first, in scheduling using an EDD algorithm. Then, in the case of late jobs, the first of these is moved to the end of production (this algorithm is detailed in Figure 6). In doing so, the total number of late jobs can be reduced (cf. Figure 4 where there is only one late job with MH, whereas EDD leads to four late jobs).

In an experimental study, Moray et al. (1991) noted that this algorithm is extremely demanding on schedulers in terms of cognitive workload: "This task [scheduling task having time pressure], for which the MH [Moore–Hodgson] gave the correct schedule, was so difficult when the operators knew the rule that their performance collapsed completely, and the task became impossible" (p. 621). For this reason, the MH algorithm is considered not comprehensible, whereas the EDD algorithm is considered comprehensible.

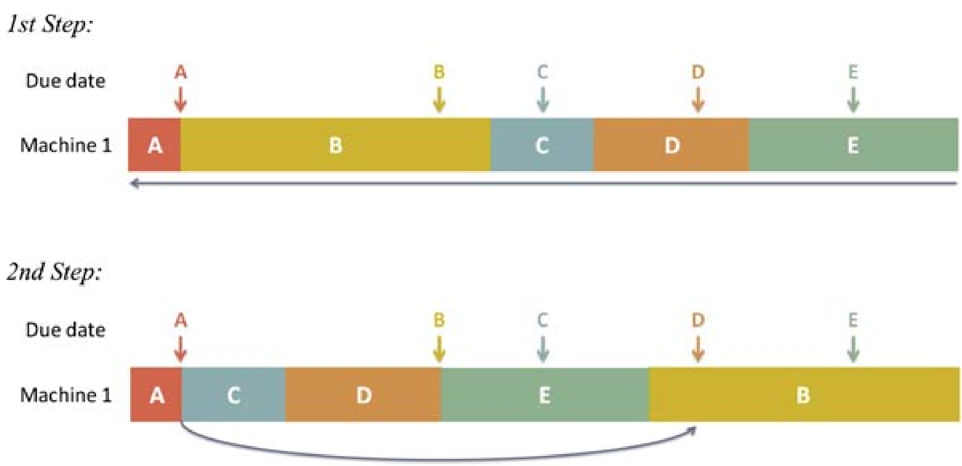

Figure 4 The Moore–Hodgson (MH) algorithm. The first step is the EDD algorithm. The second step consists of reporting the longest jobs at the end.

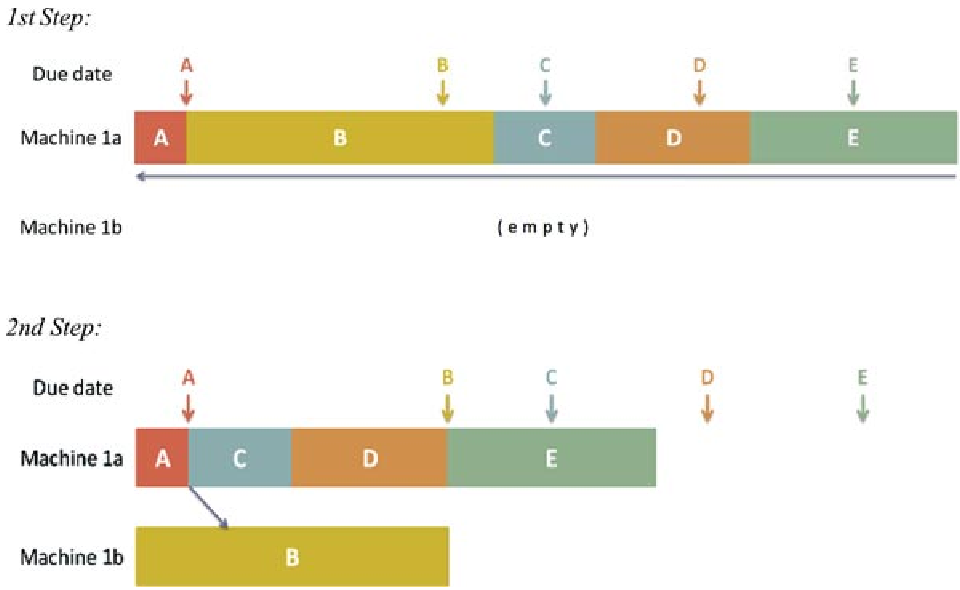

Figure 5 The Earliest Due Date + Shortest Processing Time (EDD+SPT) algorithm. The first step is the EDD algorithm. The second step consists of allocating jobs to the machine (of the same category) that will carry out the job in the shortest time.

4.1.3. Earliest Due Date + Shortest Processing Time (EDD+SPT). As is the case with the MH algorithm, the EDD+SPT algorithm is a combination of rules. The first step consists in sequencing operations from the earliest due date to the latest. There is also a second step, which leads to a decrease in algorithm comprehensibility because EDD+SPT requires different steps, as does the MH algorithm (cf. Figure 5). This second step consists of selecting the machine (and not the operation) that will carry out the operation in the shortest time (SPT; Priore, de la Fuente, & Pino, 2001). As parallel machines are considered as identical, the jobs will be dispatched to one machine or the other depending on the jobs already queued. In terms of the visual display of the schedule, this means that jobs being produced for the same group could be dispatched to machines that are not visually grouped in the Gantt chart.

As stressed previously, the visual grouping of jobs in the Gantt chart is a support to human schedulers' strategies. An algorithm that allows jobs to be displayed in groups is considered as having a comprehensible result. So EDD+SPT has a result that is less comprehensible than the EDD and the MH algorithms. In terms of algorithm comprehensibility, EDD+SPT and MH have a lower level of algorithm comprehensibility than that of the EDD algorithm (as noted in Table 1).

Three algorithms have been described. Previous studies allowed us to characterize each algorithm as having result comprehensibility or algorithm comprehensibility (Green & Appel, 1981; Moray et al., 1991; Nakamura & Salvendy, 1988).

4.2. Participants and Scenarios

In most scheduling field studies there is only one human operator to schedule. Moreover, scheduling situations are very diverse, leading to difficulties in generalizing results from

(1) Arrange the blocks in order according to their due dates (the first job being the block closest to its due date and so on).

(2) Consider the first job on the list.

(3) Check the time needed to complete the first job. If it is less than or equal to its due date then leave it where it is in the list and proceed to consider the next job on the list in the same manner. Otherwise, if its completion time is greater than its due date, then look at all the jobs you have considered so far and drop the job with the longest processing time at the end of the queue. (It's best not to do this one at all.) Now proceed to think about the next job in the order in the same manner.

(4) When all jobs have been considered, go to Step 3. Otherwise, chip away at the jobs not dropped to the end in the order you have established.

Figure 6 The description of the Moore–Hodgson algorithm in the experiment (adapted from Moray et al., 1991).

experts in charge of very different situations (Cegarra, 2008). For this reason, most experimental studies in scheduling have been carried out with students. This is the case with this study, which compares different algorithms with students of production management (all have the same level of practice and no more than a few weeks of practice). Nine students volunteered to participate in the experiment. They had training in scheduling and were familiar with the Gantt chart. Three groups of three participants were defined.

Each participant of a group learned one specific algorithm (EDD, MH, or EDD+SPT) during a training phase to get accustomed to the algorithm. In this training phase, each participant in a group was given the description of the algorithm procedure on paper (cf. Figure 6 for an example with the MH algorithm). Participants had to individually use a computerized Gantt chart tool to replicate algorithm functioning. This training phase was repeated until the participant correctly reproduced the algorithm in different conditions.

In the experimental phase, six schedules were displayed successively on the screen of the tool. The participant did not elaborate these schedules as they were automatically calculated using the algorithm that the participants had learned. Instead, for each schedule, participants were informed of a disturbance in production (e.g., the arrival of a rush job). The task was to reschedule, taking this disturbance into account. The participants had to satisfy the constraint related to the disturbance and, at the same time, preserve most constraints already satisfied by the algorithm. Their goal was to minimize the number of late jobs. After rescheduling, the next schedule (and therefore the next disturbance) was presented. Six schedules were presented in successive but random order:

- A due date change for a specific job. This is the simple scenario.
- An arrival of a rush job during a heavily loaded production period. This is the complex scenario.
- An arrival of a rush job with a flexible (uncertain) due date. This is the uncertain scenario.
- Delays in the arrival of materials, leading to delays in a lot of jobs. This is the contradictory scenario.
- A change in job priorities. This is the change scenario.
- No disturbance. This is the control scenario. In this scenario, the participant only corrects the computer-generated schedule. It allows one to determine the performance of correcting the algorithm without any disturbance (as will be detailed in Section 5.1).

The relevance of these scenarios relates to a cognitive typology detailed in another paper (Cegarra, 2008). They allow the study of rescheduling strategies while taking into account disturbances that are relevant from a cognitive point of view, especially at varying complexity levels. Indeed, when studying complacency, the complex scenario is the most important. The reason is that it requires the deepest analysis of the computer-generated schedule in order to modify the schedule *a minima* due to the high number of constraints to be taken into account. However, taking into account a higher number of scenarios that are relevant to field studies could allow us to improve the generalization of results.

4.3. Performance Measures

In order to evaluate the participants' performances and to compare them in relation to the algorithm with which they had to interact, we selected two measures: behavioral performance and industrial performance.

4.3.1. Behavioral Performance. As complacency also relates to a cognitive cost, it is important to have an evaluation of the mental workload engaged in the task. Taking into account workload analysis methods in terms of their disturbance to the task, or in terms of their capability to measure the dynamics of the mental workload, we decided to favor measures directly obtained from the scheduling task (and not, for example, from a dual-task paradigm). In this way, the workload was evaluated by the ratio between the total time spent in each scenario and the number of actions. The obtained value indicates the average time required to make a decision. Differences in this variable, which depend on the participant's group, will indicate differences in the cognitive cost of interacting with the algorithm (and its result).

Unlike the measure of industrial performance (as detailed next), behavioral performance is not a measure of the outcome but of the rescheduling process. In this process, the participants will correct scheduling algorithm failures and also take into account any disturbance. However, it would be unfair to compare time-based values of rescheduling because algorithms differ in terms of the initial scheduling performance (meaning more or less time to reschedule depending on the initial algorithm performance). So, to study only behavioral performance (B) while still taking into account the disturbance, the control scenario (where there is no disturbance) was used to evaluate the cost of correcting the algorithm. This value was then subtracted from the other scenarios to evaluate behavioral performance in rescheduling independently from the algorithm scheduling performance.

$$B = \frac{\begin{array}{l}\text{[Time to reschedule in a scenario} \\ \text{(algorithm and disturbance)]}\end{array} - \begin{array}{l}\text{[Time to reschedule in the control scenario} \\ \text{(algorithm)]}\end{array}}{\begin{array}{l}\text{[Number of actions to reschedule} \\ \text{(algorithm and disturbance)]}\end{array} - \begin{array}{l}\text{[Number of actions to reschedule in the control} \\ \text{scenario (algorithm)]}\end{array}}$$

This calculation implies that algorithm and disturbance rescheduling are considered to be independent. To this effect, the scenarios were designed to prevent overlapping as much as possible in the rescheduling actions required for the algorithm and the disturbance.

4.3.2. Industrial Performance. This performance measure refers to that prescribed to the participants in the experimental setting: to minimize the number of late jobs. As the problem is the same for the three different algorithms, any significant difference in the number of late jobs will result from differences in the interaction with the algorithm. This measure will be the main variable used to demonstrate the presence of complacent behavior (i.e., schedulers not trying to improve industrial performance even if a higher level of performance is possible).

4.4. Hypotheses Summary

Three main hypotheses can be suggested from the previous theoretical discussions:

1. In all scenarios, an algorithm that has an incomprehensible result (EDD+SPT) will lead to more complacent behavior (i.e., a lower level of performance) than an algorithm with a comprehensible result (MH at the same level of algorithm comprehensibility).
2a. An algorithm that is comprehensible will facilitate automation monitoring and ultimately limit complacency. Thus, we could hypothesize that the comprehensible algorithm (EDD) will lead schedulers to achieve a better performance than an algorithm that is not comprehensible (MH).
2b. However, as indicated previously (see Section 3.2), a contradictory alternative hypothesis can be formulated for algorithm comprehensibility. In the complex scenario, in particular, an algorithm that is comprehensible (EDD) may lead to a high cognitive workload and finally to more complacent behavior than when the algorithm is not comprehensible (MH).

5. RESULTS

5.1. Behavioral Performance

When looking at behavioral performance from a descriptive point of view, Figure 7 shows that the MH algorithm required the least amount of time to make decisions. This is particularly the case in the simple, contradictory, uncertain, and complex scenarios, although the results are less obvious in the case of the change scenario. However, it is not possible to identify an overall significant effect of the type of algorithm across all scenarios ($F(2,6) = 1.393$; *ns*; $p > .31$), nor of the type of disturbance across all algorithms ($F(4,24) = 1.544$; *ns*; $p > .22$).

When looking precisely at the effect of result comprehensibility (comparing MH to EDD+SPT algorithms), Figure 7 shows that the MH algorithm allowed a better behavioral performance than the EDD+SPT algorithm in most scenarios. However, it is not possible to note any significant effect of any scenario, as is shown in the summary table (cf. Table 2). So, the problem of result comprehensibility remains and will be discussed in light of industrial performance results in the next section.

When studying algorithm comprehensibility, Figure 7 shows that in most scenarios (except the change scenario) the EDD algorithm leads to a lower level of performance than the MH algorithm. This is particularly the case with the uncertain and complex scenarios. Detailed analyses indicate that it is not possible to demonstrate a significant effect of algorithm comprehensibility on behavioral performance in relation to all but one scenario. In the case of the complex disturbance, the EDD algorithm required about 16 seconds on average for each decision and the MH algorithm required only 2 seconds and this difference is significant ($F(1, 6) = 51.136$; $p < .001$). Moreover, the observed values are contradictory

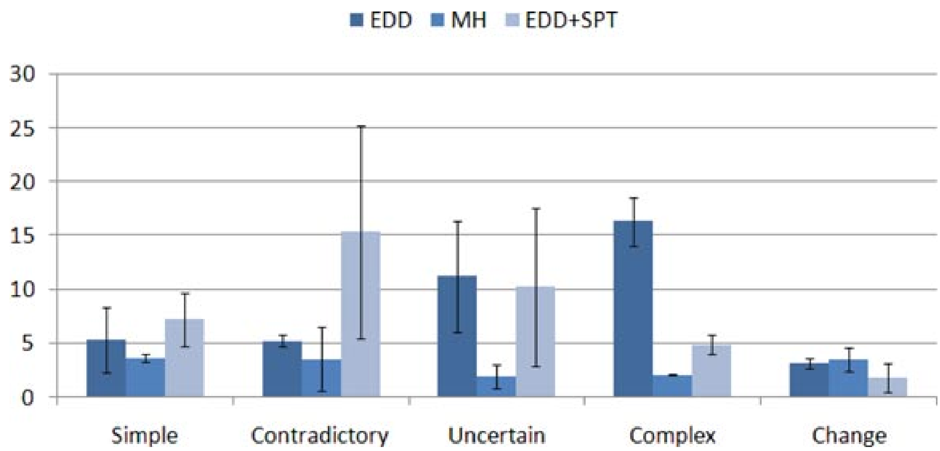

Figure 7 Behavioral performance: ratio (in seconds) between the total rescheduling time and the number of actions, depending on the disturbance and on the algorithm (the control scenario was used as a base to correct the values). Error bars represent standard error of the mean (SEM).

TABLE 2. Summary of Results for All Scenarios and Measures Depending on the Algorithm and Result Comprehensibility

Performance Scenario	Result Comprehensibility		Algorithm Comprehensibility	
	Behavioral	Industrial	Behavioral	Industrial
Simple	$F(1, 6) = 1{,}285$; *ns*; $p > .30$	$F(1, 6) = 2{,}4$; *ns*; $p > .17$	$F(1, 6) = 0{,}524$; *ns*; $p > .49$	$F(1, 6) = 0{,}15$; *ns*; $p > .71$
Contradictory	$F(1, 6) = 1{,}958$; *ns*; $p > .21$	**$F(1, 6) = 10{,}125$; $p < .02$**	$F(1, 6) = 0{,}04$; *ns*; $p > .84$	$F(1, 6) = 1{,}125$; *ns*; $p > .32$
Uncertain	$F(1, 6) = 1{,}259$; *ns*; $p > .30$	**$F(1, 6) = 10{,}137$; $p < .02$**	$F(1, 6) = 1{,}571$; *ns*; $p > .25$	$F(1, 6) = 0{,}051$; *ns*; $p > .82$
Complex	$F(1, 6) = 1{,}973$; *ns*; $p > .20$	**$F(1, 6) = 18$; $p < .01$**	**$F(1, 6) = 51{,}136$; $p < .001$**	**$F(1, 6) = 6{,}125$; $p < .05$**
Change	$F(1, 6) = 1{,}4$; *ns*; $p > .28$	$F(1, 6) = 1{,}442$; *ns*; $p > .27$	$F(1, 6) = 0{,}052$; *ns*; $p > .82$	$F(1, 6) = 0{,}231$; *ns*; $p > .64$
Control	*	**$F(1, 6) = 40{,}5$; $p < .001$**	*	$F(1, 6) = 4{,}5$; *ns*; $p > .07$

Note: Significant results are in bold. An asterisk indicates there is no value in these cases as they are used as a base and subtracted from each scenario in behavioral performance.

to Hypothesis 2a. In the complex scenario, when the algorithm is comprehensible, the level of behavioral performance is lower than when the algorithm is not comprehensible. This means that the amount of time taken to make a decision was less in the MH algorithm, even though it was considered to be harder to understand than the EDD algorithm.

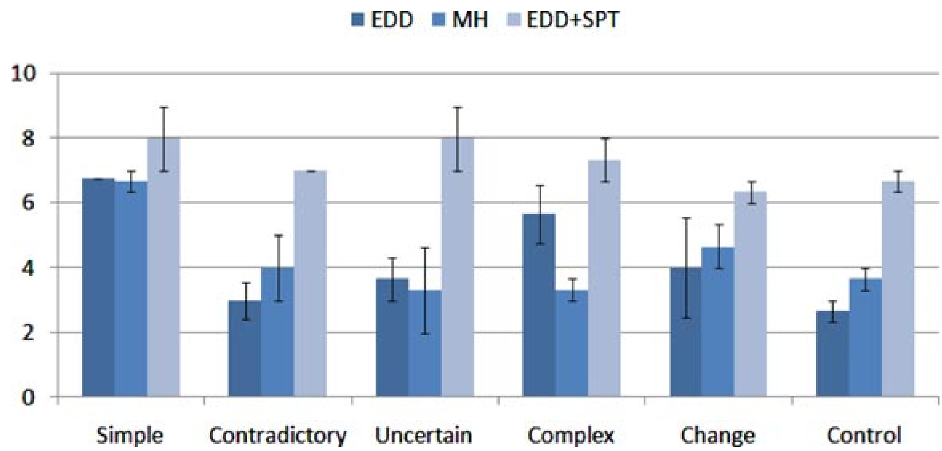

Figure 8 Industrial performance: the number of late jobs at the end of the rescheduling time, depending on the scenario and on the algorithm. Error bars represent SEM.

5.2. Industrial Performance

From a descriptive point of view, Figure 8 shows that the EDD+SPT algorithm leads to the highest number of late jobs (industrial performance) and that the MH and EDD algorithms have similar results. In terms of scenarios, Figure 8 shows that the simple scenario leads to the highest number of late jobs. There is also an overall significant effect of the algorithm type across all scenarios ($F(2, 6) = 14.665$; $p < .004$) and an effect on the scenario across all algorithms ($F(5, 30) = 6.981$; $p < .001$).

When studying result comprehensibility, Figure 8 shows that in all scenarios, the EDD+ SPT algorithm leads to a higher number of late jobs than does the MH algorithm. However, it was not possible to demonstrate a significant effect for two scenarios: simple ($F(1, 6) = 2.4$; *ns*; $p > .17$) and change ($F(1, 6) = 1.442$; *ns*; $p > .27$). The other scenarios each show that result comprehensibility has a significant effect: contradictory ($F(1, 6) = 10.125$; $p < .02$), uncertain ($F(1, 6) = 10.137$; $p < .02$), complex ($F(1, 6) = 18$; $p < .01$), and control ($F(1, 6) = 40.5$; $p < .001$) scenarios. These results indicate that an algorithm with low result comprehensibility will lead to a lower level of industrial performance in most rescheduling scenarios, thus supporting Hypothesis 1.

When studying algorithm comprehensibility (comparing the MH to the EDD algorithm), Figure 8 does not show any clear difference in the number of late jobs, except for the complex scenario where there were significantly fewer late jobs in the MH algorithm (approximately three late jobs on average) than in the EDD algorithm (about five late jobs on average) ($F(1, 6) = 6.125$; $p < .05$). In terms of behavioral performance, this highlights the fact that when participants understand an algorithm for a complex situation, this leads to a lower level of performance. In the discussion, we will elaborate further on these results.

6. DISCUSSION

6.1. Result Comprehensibility

Results indicate that an algorithm with low result comprehensibility will lead to a lower level of industrial performance in most rescheduling scenarios. Complacency is usually associated with supervision failure, indicating that the human operator does not detect critical changes or errors in the automated system. In this experiment, when the result is not comprehensible (EDD+SPT), the Gantt chart does not highlight some important information for the evaluation of the schedule, such as precedence constraints. Even so, participants did not significantly increase their investment in rescheduling (in terms of cognitive costs) when comparing the algorithm with a comprehensible result (MH). Because humans favor the achievement of satisfactory performance rather than optimal performance, participants who faced result incomprehensibility invested less attention than necessary in satisfying other constraints (such as those related to due date). Finally, this led to a significantly poorer industrial performance when the algorithm result is not comprehensible (as suggested by Hypothesis 1).

Many studies have noted the importance of graphical representation in determining human scheduling strategies (e.g., Cegarra & Hoc, 2003; Dessouky et al., 1995; Gibson & Laios, 1978; Higgins, 1996; Sanderson, 1991). Thus, it is very important that result comprehensibility is taken into account because of the consequences it could have on schedule evaluation. In fact, many authors have noted that schedulers spend from 80% to 90% of their time taking into account all the constraints of the problem, and only 10% to 20% actually building the schedule (see Sanderson, 1989). So, it is particularly important to increase result comprehensibility in order to decrease the cost of looking for relevant constraints, which could lead to complacency.

To decrease complacent behavior, one could introduce an ecological interface that is especially designed to display relevant constraints to the human operator (Vicente, 2002). Recently, Higgins (2001) successfully designed an ecological interface for scheduling in a small-job shop. This study could, therefore, be used as a starting point to better understand how human schedulers make the most of scheduling interfaces and to prevent complacency due to poor result comprehensibility. Since the study by Gibson and Laios (1978), no comparisons of different graphical representations of scheduling problems have been carried out. Moreover, the link between complacency and interface design has not yet led to many experimental studies (but see, for instance, Furukawa & Parasuraman, 2003).

6.2. Algorithm Comprehensibility

Behavioral performance indicates that the amount of time taken to make a decision was less in the MH algorithm, even though it was considered to be harder to understand than the EDD algorithm. To explain this result, one has to refer to previous studies with the MH algorithm.

Moray et al. (1991) noted that humans do not come across the MH algorithm by themselves; indeed, even those being trained in this algorithm had difficulties in applying it. This highlights the fact that participants in the MH algorithm group did not use any knowledge about the algorithm that they had acquired during training. So, one can consider that participants with a comprehensible algorithm (EDD) tried, in part, to take into account their knowledge of this algorithm. Consequently, they required more time to make decisions and their available cognitive resources decreased. On the other hand, participants confronted with the incomprehensible algorithm (MH) did not try to use any knowledge of the algorithm and had more free use of resources to engage in the rescheduling activity. The complex scenario was particularly relevant with regard to this result since many different jobs (a high number of constraints) have to be taken into account simultaneously, requiring the most cognitive resources. In other words, the more complex algorithm is easier (in terms of cognitive cost) to interact with. Moreover, results indicate that when participants understand an algorithm, this also leads to a lower level of industrial performance. This means that participants accepted a lower level of performance because of the increased cognitive cost needed to modify the schedule.

Initially, understanding the algorithm could seem a way of increasing performance. Our results underline another conclusion: understanding an algorithm leads to a decrease in level of performance. This allows us to reject Hypothesis 2a and to favor Hypothesis 2b; instead of supporting human–machine cooperation, the comprehensibility of the algorithm leads to complacent behavior. So, in this case, complacent behavior is not due to the human being insufficiently aware of algorithm functioning, but rather being too aware of its functioning and, subsequently, giving up.

In our experiment, participants learned the detailed algorithm functioning. However, schedulers do not necessarily require a full understanding of the algorithm, but only the most informative points, such as algorithm limits. Davis and Kottemann (1995) designed an experiment to evaluate the effect of describing the performance of an algorithm (using such phrases as “the heuristic has historically outperformed 90% of your peers”) and outcome feedback (comparing subjects’ performance with the performance of the algorithm). They noted that performance improved with outcome feedback and that algorithm description had no overall effect. However, in practice, it is extremely difficult to measure scheduling performance because of the number of potentially contradictory objectives (Gary, Uzsoy, Smith, & Kempf, 1995; Wiers, 1997). Indeed, it is sometimes impossible to design a schedule that can fully satisfy all of them (Dessouky et al., 1995). An alternative approach is to describe the algorithm functioning in terms of algorithm failures. For example, it could schematically mean describing the earliest due date algorithm by its poor performance when a long job is late and then is followed by many other jobs that are almost completed by their due date (as exemplified by the MH algorithm). Thus, schedulers are not trained to the correct functioning of the algorithm but rather to its drawback; its *validity domain*. This may allow for evaluating the support coming from a more efficient algorithm training, while not trying to train the human operator to the precise functioning of the algorithm.

7. CONCLUSION

There is only partial automation in scheduling situations and human operators are necessary for rescheduling. For this reason, it has been suggested that the complacency phenomenon be extended beyond the classical sense of the cost of supervision to take into account the cost of evaluating and correcting a machine’s schedule. This means the complacency phenomenon is multifaceted:

- Complacency is traditionally associated with a decrease in automation supervision, when the operator has manual tasks to attend to, while facing a high cognitive workload. In this way, complacency has been attributed to the human tendency to place too much trust in automated systems.
- Our study indicates that complacency could also result from the high cost of interacting with the automation, implying a high cognitive workload. In this case, the human operators probably do not place too much trust in the automation (even if minimal trust is indeed required to cooperate). Instead, they accept a level of satisfactory performance that is lower than they can possibly attain.

Our results stress that result comprehensibility is one necessary factor for limiting complacency because of the importance for schedulers to be able to immediately identify relevant constraints for their rescheduling. More systematic studies, with larger sample sizes, could be carried out to develop interfaces that will specifically take this into account. Our results also indicate that algorithm comprehensibility implies complacency because it adds too high a cognitive workload and leads to poor industrial performance. A further suggestion is to train schedulers about the algorithm validity domain in order to identify if more efficient training could lead to an increase in performance (see also Bahner et al., 2006). These two suggestions could allow our knowledge about human–machine cooperation in scheduling to be extended even further. Indeed, the current lack of empirical studies is noted by several authors (Hoc, Mebarki, & Cegarra, 2004; Moray, Hiskes, Lee, & Muir, 1995; Sanderson, 1989).

ACKNOWLEDGMENTS

We thank our two anonymous reviewers for giving instructive and critical feedback, thus helping to improve the article. This research is supported by COST Action (European Cooperation in the field of Scientific and Technical Research) A29: Human and Organisational Factors in Industrial Planning and Scheduling (HOPS).

REFERENCES

Bahner, J. E., Huper, A.-D., & Manzey, D. (2006). Complacency in automated fault management: How to keep operators alert towards possible failures of automated aids. In R. N. Pikaar, E. A. P. Koningsveld, & P. J. M. Settels (Eds.), Proceedings of the 16th World Congress on Ergonomics (International Ergonomics Association '06) (pp. 3969–3974). Maastricht, The Netherlands: Elsevier.

Bi, S., & Salvendy, G. (1994). Analytical modeling and experimental study of human workload in scheduling of advanced manufacturing systems. The International Journal of Human Factors in Manufacturing, 4(2), 205–234.

Cegarra, J. (2008). A cognitive typology of scheduling situations: A contribution to laboratory and field studies. Theoretical Issues in Ergonomics Science, 9(3), 201–222.

Cegarra, J., & Hoc, J. M. (2003). From human heuristics to the design of scheduling interfaces. In Proceedings of the 1st Multidisciplinary International Conference on Scheduling: Theory and applications (pp. 236–239). University of Nottingham: Nottingham, UK.

Chen, M. B., & Hwang, S. L. (1997). A decision support system for production scheduling and control. Human–Computer Interaction, 2, 15–18.

Davis, F. E., & Kottemann, J. E. (1995). Determinants of decision rule use in a production planning task. Organizational Behavior and Human Decision Processes, 63(2), 145–157.

Dessouky, M. I., Moray, N., & Kijowski, B. (1995). Taxonomy of scheduling systems as a basis for the study of strategic behavior. Human Factors, 37, 443–472.

Furukawa, H., & Parasuraman, R. (2003). Supporting system-centered view of operators through ecological interface design: Two experiments on human-centered automation. In Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting (pp. 567–571). Santa Monica, CA: Human Factors and Ergonomics Society.

Gary, K., Uzsoy, R., Smith, S. P., & Kempf, K. G. (1995). Measuring the quality of manufacturing schedules. In D. E. Brown & W. T. Scherer (Eds.), Intelligent scheduling systems (pp. 129–154). Boston: Kluwer Academic.

Gibson, R., & Laios, L. (1978). The presentation of information to the job-shop scheduler. Human Factors, 20(6), 725–732.

Green, G. I., & Appel, L. B. (1981). An empirical analysis of job shop dispatch rule selection. Journal of Operations Management, 1(4), 197–203.

Haider, S. W., Moodie, C. L., & Buck, J. R. (1981). An investigation of the advantages of using a man-computer interactive scheduling methodology for job shops. International Journal of Production Research, 19, 381–392.

Higgins, P. G. (1996). Interaction in hybrid intelligent scheduling. International Journal of Human Factors in Manufacturing, 6, 185–203.

Higgins, P. G. (2001). Architecture and interface aspects of scheduling decision support. In B. L. MacCarthy & J. R. Wilson (Eds.), Human performance in planning and scheduling: Fieldwork studies, methodologies and research issues (pp. 245–279). London: Taylor & Francis.

Hoc, J. M. (2000). From human-machine interaction to human-machine cooperation. Ergonomics, 43, 833–843.

Hoc, J. M. (2001). Towards a cognitive approach to human-machine cooperation in dynamic situations. International Journal of Human-Computer Studies, 54, 509–540.

Hoc, J. M., & Amalberti, R. (2007). Cognitive control dynamics for reaching a satisficing performance in complex dynamic situations. Journal of Cognitive Engineering and Decision Making, 1(1), 22–55.

Hoc, J. M., Mars, F., Milleville-Pennel, I., Jolly, E., Netto, M., & Blosseville, J. M. (2006). Evaluation of human-machine cooperation modes in car driving for safe lateral control in bends: Function delegation and mutual control modes. Le Travail Humain, 69, 153–182.

Hoc, J. M., Mebarki, N., & Cegarra, J. (2004). L'assistance à l'opérateur humain pour l'ordonnancement dans les ateliers manufacturiers. Le Travail Humain, 67, 181–208.

Jüngen, F. J., & Kowalczyk, W. (1995). An intelligent interactive project management support system. European Journal of Operational Research, 84, 60–81.

Layton, C., Smith, P. J., & McCoy, E. (1994). Design of a cooperative problem-solving system for en-route flight planning: An empirical evaluation. Human Factors, 36, 94–119.

Lopez, P., & Roubellat, F. (Eds.). (2007). Production scheduling. London: ISTE.

McKay, K. N., & Wiers, V. C. S. (2001). Decision support for production scheduling tasks in shops with much uncertainty and little autonomous. In B. L. MacCarthy & J. R. Wilson (Eds.), Human performance in planning and scheduling: Fieldwork studies, methodologies and research issues (pp. 165–177). London: Taylor & Francis.

Metzger, U., & Parasuraman, R. (2001). The role of the air traffic controller in future air traffic management: An empirical study of active control versus passive monitoring. Human Factors, 43(4), 519–528.

Moray, N. (2003). Monitoring, complacency, scepticism and eutactic behaviour. International Journal of Industrial Ergonomics, 31, 175–178.

Moray, N., Dessouky, M. I., Kijowski, B. A., & Adapathya, R. (1991). Strategic behavior, workload, and performance in task scheduling. Human Factors, 33, 607–629.

Moray, N., Hiskes, D., Lee, J., & Muir, B. M. (1995). Trust and human intervention in automated systems. In J. M. Hoc, P. C. Cacciabue, & E. Hollnagel (Eds.), Expertise and technology (pp. 183–194). Hillsdale, NJ: Lawrence Erlbaum Associates.

Nakamura, N., & Salvendy, G. (1988). An experimental study of human decision-making in computer-based scheduling of flexible manufacturing systems. International Journal of Production Research, 26, 567–583.

Parasuraman, R., Molloy, R., & Singh, I. L. (1993). Performance consequences of automation-induced "complacency." The International Journal of Aviation Psychology, 3, 1–23.

Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. Human Factors, 39, 230–253.

Priore, P., de la Fuente, D., & Pino, R. (2001). Learning-based scheduling of flexible manufacturing systems using case-based reasoning. Applied Artificial Intelligence, 15(10), 949–963.

Sanderson, P. M. (1989). The human planning and scheduling role in advanced manufacturing systems: An emerging human factors domain. Human Factors, 31, 635–666.

Sanderson, P. M. (1991). Towards the model human scheduler. International Journal of Human Factors in Manufacturing, 1, 195–219.

Smith, P. J., McCoy, E., & Layton, C. (1997). Brittleness in the design of cooperative problem-solving systems: The effects on user performance. IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, 27, 360–371.

Tabe, T., Yamamuro, S., & Salvendy, G. (1990). Knowledge elicitation in scheduling FMS: Towards a hybrid intelligent system. International Journal of Industrial Ergonomics, 5, 17–27.

Valax, M. F., & Cellier, J. M. (1992). Aides à l'organisation du travail dans les ateliers: problèmes du décalage entre prévision et réalisation. In G. De Terssac & P. Dubois (Eds.), Les nouvelles rationalisations de la production (pp. 121–137). Toulouse, France: Cépaduès.

Vicente, K. J. (2002). Ecological interface design: Progress and challenges. Human Factors, 44, 62–78.

Wiener, E. L., & Curry, R. E. (1980). Flight deck automation: Promises and problems. Ergonomics, 23, 995–1011.

Wiers, V. C. S. (1997). Human-computer interaction in production scheduling. Analysis and design of decision support systems for production scheduling tasks. Doctoral thesis, University of Eindhoven, The Netherlands.

Wiers, V. C. S., & van der Schaaf, T. W. (1996). A framework for decision support in production scheduling tasks. Production Planning & Control, 25(2), 533–544.