# 3 Significant Complexity Factors

As shown in the previous chapter, we have to develop a novel framework to evaluate the complexity of proceduralized tasks. To this end, it is natural to start by identifying what factors make the performance of proceduralized tasks complicated. In other words, instead of many complexity factors pertaining to CR, WR, and CMP, it is necessary to consider different factors that could annoy people by demanding additional cognitive resources for TP. For this reason, many works dealing with causal factors regarding the complexity of a proceduralized task have been reviewed. As a result, a total of nine categories of complexity factors were ddistinguished. In this chapter, I would like to explain the meaning as well as the characteristics of each category.

It is to be noted that, in the course of this literature survey, two basic principles have been applied to the selection of complexity factors. Therefore, before explaining task complexity factors, it is helpful to clarify the basic principles that I have adopted.

## 3.1 Complexity Factors of a Process Control Task

The first principle was that, although this may sound natural, we must focus on complexity factors pertaining to the nature of the tasks being considered. Jonassen (2000) stated that "These cognitive demands are situationally specific. Arguing a case in court, for instance, would demand a different set of cognitive skills from those needed for air traffic controlling (p. 79)."

This means that, before identifying significant factors that make the performance of proceuralized tasks complicated, we should make explicit the task type we are concerned with. In this regard, our interest is in managing the complexity of proceduralized tasks used in a large and safety-critical process control system. Therefore, we have to concentrate on complexity factors related to a process control task.

At the same time, we need to concentrate on complexity factors pertaining to a supervisory control task. For example, Stassen et al. (1990) and Johannsen et al. (1994) specified that a supervisory control generally consists of several subtasks, such as monitoring, interpreting, planning, fault management, and intervention, etc. Here, it is very interesting to consider the definition of Leitch and Gallanti (1992), who articulated that a process control task must deal with a dynamic physical system evolving over time, which consists of five primitive behaviors as listed in Table 3.1.

**Table 3.1** Five primitive behaviors related to a process control task

| Behavior | Meaning |
|---|---|
| Decision | An action generating hypotheses or conclusions that satisfy given constraints or specifications |
| Prediction | An action generating future states from the present state using an implicit or explicit model of a system |
| Identification | An action related to determining unknown or unmeasurable states from known or assumed states |
| Interpretation | An action generating a situational description from observable data |
| Execution | An action related to the actuation of a target system |

It is to be emphasized that process control tasks and supervisory control tasks resemble each other, because primitive behaviors related to process control tasks seem to be directly comparable to those of supervisory control tasks. For example, an intervention behavior is congruent with an execution behavior shown in Table 3.1, and a fault management behavior seems to be comparable with an identification behavior, etc. Accordingly, we need to search the complexity factors that would be related to either process control or supervisory control tasks.

## 3.2 Complexity Factors of a Novice

The second principle was that all kinds of complexity factors should be applicable not to a user's manual but to a procedure that provides practical contents for performing a process control task. This principle is closely connected to the definition of a good procedure – *the procedure should be developed so that even a novice can properly follow it*. Here, it should be noted that the novice in this book implies a human operator who already has a certain level of domain knowledge. Let us consider the following examples.

- Starts a computer using the power button.
- Starts a computer using the power button. It is located on the front panel of the computer. It is round and about the size of a quarter. You can boot the computer by pushing this button.

Here, it is presumed that the first instruction may be unclear for a person seeing a computer for the first time. This is because he or she may have felt frustration in starting the computer due to a lack of basic knowledge regarding, for example, the power button, its location and so on. In contrast, since the second instruction contains very detailed information, it is expected that even someone seeing a computer for the first time would easily boot it up.

Unfortunately, the second instruction is closer to what you would find in a user's manual rather than a procedure. This is because it simultaneously provides two kinds of descriptions that have a different purpose, such as (1) the description of an action to be done (i.e., *starts a computer using a power button*) and (2) additional descriptions about the physical form of the power button. In other words, since we are looking for complexity factors making the performance of proceduralized tasks complicated, we have to pick out those that are meaningful for a novice who has a minimum level of domain knowledge. A person with general knowledge of cooking as well as how to deal with kitchenware is a good example of a novice who is ready to use recipes (i.e., procedures). Similarly, operating personnel of NPPs who have just passed a basic training course are novices who can follow a procedure. Therefore, several factors pertaining to a lack of domain-specific knowledge, such as experience (Thelwell 1994; Maynard and Hakel 1997; Van Eekhout and Rouse 1981; Morris and Rouse 1985) or job training/skill (Li and Wieringa 2000; Leplat 1998), have been excluded from the consideration of task complexity factors. For the sake of convenience, henceforth, a *qualified operator* will be referred to as a person who is ready to follow a procedure, while an *unqualified operator* will refer to an ordinary person without a minimum level of domain knowledge.

## 3.3 Identifying Complexity Factors

With the aforementioned principles in mind, existing works that deal with many kinds of complexity factors have been reviewed. As a result, in total nine categories of complexity factors were identified as epitomized in Table 3.2. Appendix A summarizes all the complexity factors belonging to each category. It is to be noted that the meanings of four categories (i.e., time pressure, temporal characteristics, system characteristics, and personal characteristics) are self-explanatory from the summary in Appendix A. Therefore, more detailed explanations will be provided for the remaining categories.

### *3.3.1 Amount of Information and Number of Actions*

The first category is the amount of information to be processed by a qualified operator. For example, it seems to be clear that a proceduralized task pertaining to operating a huge chemical plant is more complicated than that of a small domestic factory, since qualified operators have to manage more information including process alarms or process parameters, etc. Therefore, it is strongly expected that qualified operators working in the former need to spend more cognitive resources compared to those working in the latter.

**Table 3.2** Categories of complexity factors

| No. | Categories | Description |
|---|---|---|
| 1 | Amount of information | Amount of information to be processed by a qualified operator |
| 2 | Number of actions | Number of actions to be conducted by a qualified operator |
| 3 | Logical entanglement | Logical complexity due to the sequence of actions to be followed by a qualified operator |
| 4 | Amount of domain knowledge | Amount of domain knowledge to be considered by a qualified operator |
| 5 | Level of engineering decision | Amount of cognitive resources to be used by a qualified operator, which is needed to establish an appropriate decision criterion |
| 6 | Time pressure | Time allowed for the performance of a task |
| 7 | Temporal characteristics | Degree of a task arrival, task frequency, task overlap, etc. |
| 8 | System characteristics | Dynamic characteristics of a task due to the nature of the system |
| 9 | Personal characteristics | Aptitude, intelligence, ability, and cognitive style of a qualified operator |

Similarly, the number of actions to be conducted by qualified operators is an obvious factor making the performance of proceduralized tasks complicated, because they need to use cognitive resources to properly conduct each and every action. However, this factor seems to be somewhat *superficial*, because the complexity of the cookie recipe, which includes eight actions (Fig. 1.6), is definitely different from that of an arbitrary proceduralized task that consists of two procedural steps with the same number of actions (Fig. 3.1). It is to be noted that Fig. 3.2 depicts a target system to be managed by the proceduralized task shown in Fig. 3.1.

As depicted in Fig. 3.2, there are four valves contributing to the change of the water level of a reservoir (i.e., Tank 1). First, an influx into Tank 1 is governed by IV 1, which has only two operable states – open and close. Meanwhile, CV 1 regulates the rate of an outflow from Tank 1 by continuously adjusting its open position from 0% to 100%. In addition, in order to prevent the overfill of Tank 1, there are two bypass vales (BV 1 and BV 2), which are normally in a closed state. That is, when the water level is too high, these valves can be used to provide another flow path draining the water from Tank 1. In this regard, three more categories – logical entanglement, the amount of domain knowledge, and the level of an engineering decision – are needed to reflect the *hidden* aspect of the complexity of proceduralized tasks.

**RESPONSE TO THE HIGH WATER LEVEL OF TANK 1**

1 **IF** the water level of Tank 1 is within 50~70%,
**THEN**
- 1.1 Close IV 1[*].
- 1.2 Increase the opening position of CV 1[*] to 10% higher than the current position.

2 **IF** the water level of Tank 1 is over 70%,
**THEN** perform one of the following:
- 2.1 Increase outflow.
  - 2.1.1 Close IV 1.
  - 2.1.2 Increase the opening position of CV 2 to 30% higher than the current position.

**OR**

- 2.2 Provide bypass line.
  - 2.2.1 Open BV 1[*].
  - 2.2.2 Open BV 2.

[*] IV, CV, and BV stand for isolation valve, control valve, and bypass valve, respectively.

**Fig. 3.1** Arbitrary proceduralized task pertaining to controlling the water level of a reservoir

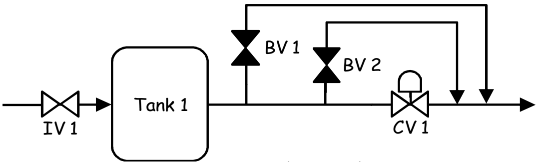



**Fig. 3.2** An arbitrary system including four valves and a reservoir

## *3.3.2 Logical Entanglement*

First, we need to consider the logical entanglement caused by the relationship of the required actions. For example, Kieras and Polson (1985) regarded the number of execution sequences needed to achieve a goal as one of the dominant complexity factors. In addition, similar comments were made by many researchers such as Leplat (1998), Li and Wieringa (2000), Sundstrom (1993), Thelwell (1994), Wood (1986), and Wood and Locke (1990). These comments can be conceptualized as *path-goal multiplicity* (Jacko and Salvendy 1996) or *multiple path-goal connections* (Campbell 1988), which indicates the number of different ways to perform a task. To explain this concept, let us consider Fig. 3.3, which illustrates the sequence of actions in the recipe shown in Fig. 1.6.

As shown in Fig. 3.3, the recipe just provides a single way to bake a cookie. If qualified operators conduct a proceduralized task like this, then they perhaps do not need to use additional cognitive resources to clarify *if they are correctly following the sequence of actions, if they are doing what they need to be done,* etc. In contrast, let us assume that qualified operators have to follow the sequence of actions depicted in Fig. 3.4, which is related to coping with the high water level of Tank 1.

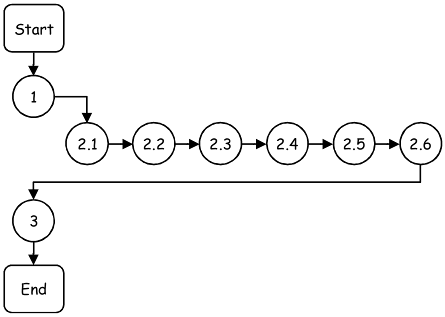
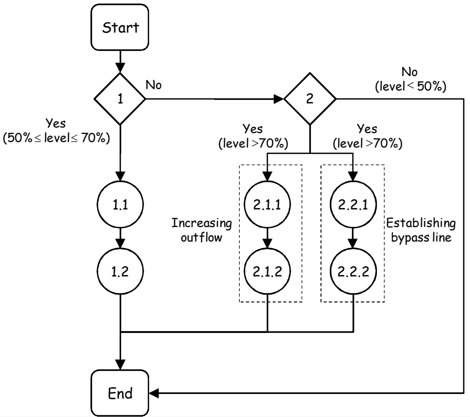



**Fig. 3.3** Sequence of actions to bake chocolate chip cookies

In Fig. 3.4, there are four paths to accomplish this task. First, if the water level of Tank 1 is maintained between 50 and 70%, then qualified operators need to perform two actions (*1.1* and *1.2* in Fig. 3.4). Meanwhile, if the water level of Tank 1 is greater than 70%, then qualified operators have to select either the second or the third path to decrease the water level of Tank 1. The second path consists of two actions designed to increase the rate of an outflow from Tank 1 by opening CV 1. The third path also consists of two actions but seems to be more aggressive, because its intention is to provide additional drain channels by opening two bypass valves, BV 1 and BV 2. The last path is somewhat trivial because it says there is nothing to do if the water level is less than 50%.




| Designation | Action description |
|---|---|
| 1 | Determine the water level of Tank 1 is 50~70% |
| 1.1 | Close IV 1 |
| 1.2 | Increase the opening position of CV 1 to 10% higher than the current position |
| 2 | Determine the water level of Tank 1 is over 70% |
| 2.1.1 | Close IV 1 |
| 2.1.2 | Increase the opening position of CV 2 to 30% higher than the current position |
| 2.2.1 | Open BV 1 |
| 2.2.2 | Open BV 2 |

**Fig. 3.4** Sequence of required actions related to the proceduralized task shown in Fig. 3.1

Consequently, compared to baking a cookie, it seems that qualified operators may use additional cognitive resources to complete this task because they need to pay attention to following the correct sequence of actions with respect to the situation at hand. In general, therefore, it is expected that the greater the number of possible paths to accomplish a proceduralized task, the more cognitive resources will have to be used by qualified operators.

### *3.3.3 Amount of Domain Knowledge*

The next category is the amount of domain knowledge, because it is natural to assume that the amount of domain knowledge for carrying out each action is not equal. Actually, the results of existing studies support this assumption, because they have revealed that qualified operators need to use their knowledge of a system in order to carry out a procedure (Boy and Brito 2000; Spangler and Peters 2001; Wright et al. 1998). In this light, it is very interesting to compare the original proceduralized task shown in Fig. 3.1 with a slightly modified proceduralized task as illustrated in Fig. 3.5.

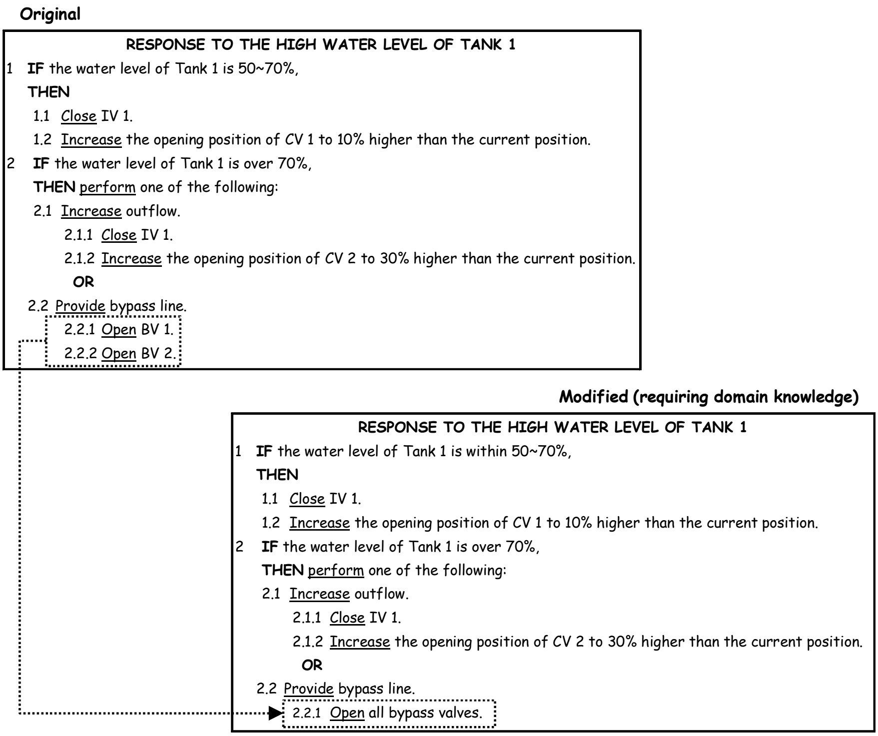



**Fig. 3.5** Actions requiring different levels of domain knowledge

First, as mentioned earlier, the purpose of the required actions that are enclosed by dotted lines in the *original* task is to reduce the water level of Tank 1 by opening bypass valves. To this end, each action showed a dedicated component (i.e., BV 1 and BV 2) as a target to be manipulated by qualified operators. This means that qualified operators probably do not need to use additional cognitive resources to recall (or extract) appropriate domain knowledge, such as a component's configuration because each action is limited to component itself.

In contrast, the required action enclosed by dotted lines in the *modified* task may demand a higher level of domain knowledge because it is related not to a dedicated component but to a set of components grouped to accomplish a desired function. In other words, although the intention of this action is identical to that of two previous actions (i.e., *open BV 1* and *open BV 2*), it probably forces qualified operators to recall a kind of domain knowledge about the configuration of bypass valves, such as *how many bypass valves are there?* In addition, since a complicated process control system will include many components with many different functions, it is generally expected that the greater the number of components, the more domain knowledge will have to possess qualified operators. This strongly implies that the amount of cognitive resources needed to recall the proper domain knowledge will increase in proportion to the number of components included in the process control system being considered.

It should be noted that many researchers have reported a similar concern. For example, Rouse (1978) and Rouse and Rouse (1979) pointed to the number of components as one of the major contributors to task complexity. In addition, Allen et al. (1996), Morris and Rouse (1985), Leplat (1998), and Liao and Palvia (2000) commonly distinguished two kinds of task complexity factors, such as the number of functional relations among components as well as equipment. Moreover, although experimental data have been collected from unqualified operators, Park et al. (2008) observed that their performance seemed to be significantly affected by the amount of domain knowledge they possessed. Consequently, it is reasonable to regard the amount of domain knowledge as one of the dominant factors on the complexity of proceduralized tasks. A more detailed explanation can be found in Sect. 6.3.2.

### *3.3.4 Level of an Engineering Decision*

Another complexity category is the level of an engineering decision, which is related to the amount of cognitive resources used to establish appropriate decision criteria for performing required actions. In order to understand the nature of an engineering decision, it may be necessary to answer two crucial questions: (1) *what is the engineering decision*? and (2) *why do qualified operators need to decide something while they are performing a proceduralized task*?

First, let us consider the following explanations given by Turk (2001) and Ditlevsen (2003), who identified an important feature of engineering decisions, respectively.

> Engineering is based on sound principles of mathematics and physics, however, not every engineering decision is based on calculations and models. Engineers also use intuition, common sense and insight when they design. The origin of such 'feelings' (i.e. intuition, common sense, etc.) could be numerous, perhaps from experiences" (see p. 247 of Turk 2001).
>
> In engineering decisions the usual situation is that it is generally not possible to choose the safe lottery, i.e. the lottery that for sure gives the benefit and never the loss. This can be expressed by saying that among all the possible lotteries of relevance in the considered technical problem only some of the lotteries are realizable. To be able to choose among the realizable lotteries in a rational way the decision maker must, at least partly, put the lotteries in some priority order of preference that points at a most preferred realizable lottery (see p. 167 of Ditlevsen 2003).

The above excerpts state that engineers will use not only domain-specific knowledge but also all kinds of available knowledge (such as feeling, intuition, or common sense, etc.) in order to find a practical solution to an actual problem. In a similar vein, in order to correctly perform what they have to do, qualified operators will do their best to establish proper decision criteria by using all kinds of available knowledge. It is to be noted that, for this reason, the term *the level of an engineering decision* was adopted in this book instead of *the level of a decision*.

Second, in order to explain why qualified operators need to make an engineering decision, let us recall the cookie baking episode in Chap. 1. In this episode, although I followed all the required actions very well, I made several mistakes in the course of baking cookies. This is because I failed to establish correct decision criteria, such as *the nature of a smooth batter*, *what a nicely browned edge* and so on. As a result, I got hard and bitter tasting cookies. This clearly shows that establishing proper decision criteria is crucial for accomplishing required actions.

Actually, Sundstrom (1993) pointed out that qualified operators who are working in a dynamic task environment should constantly update their perception in order to make two kinds of decisions regarding (1) what control tasks need to be accomplished and (2) how they need to be prioritized. Subsequently, Sundstrom identified several task complexity factors including (1) interrelatedness of assessment, choice, and evaluation rules, (2) interconnectedness of operational states, (3) relation between indicators and operational states, (4) the number of assessments, choices and evaluation rules, and (5) the number and relationship between conditions for assessments, choices and evaluation rules. In addition, Kieras and Polson (1985), Schmuck and Gundlach (1989), Svensson et al. (1997), and Thelwell (1994) identified similar complexity factors.

For example, let us consider the sequence of actions illustrated in Fig. 3.4, which has an unusual decision point. That is, qualified operators have to select one of the action sequences, either *increasing outflow* or *establishing a bypass line*, which is more appropriate for decreasing the water level of Tank 1. To this end, qualified operators need a decision criterion by which the proper action sequence can be determined. Unfortunately, settling on a decision criterion is harder than it seems, because qualified operators need to integrate at least two kinds of information (i.e., the trend of the water level of Tank 1 and the open position of CV 1) to assess an ongoing situation. Figure 3.6 will be helpful to illustrate this intricacy.

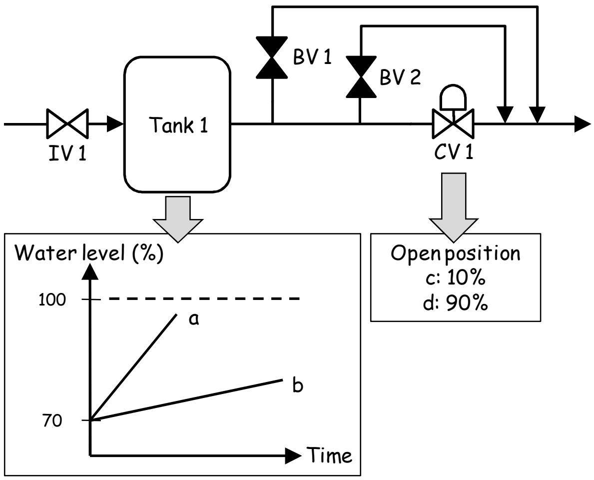



**Fig. 3.6** Hypothetical situations with which qualified operators may be faced

The first situation with which qualified operators may be faced is the combination of {a, d}. This situation represents *the water level is drastically increasing* and *the open position of CV 1 is 90%*. In this situation, most qualified operators might select the action sequence related to *establishing a bypass line*. That is, since CV 1 is already opened to 90%, it is anticipated that this valve will not be able to reduce the water level that will apparently soon reach 100%. In contrast, in the situation of {b, c}, most qualified operators would probably select the action sequence pertaining to *increasing outflow*, because the gradual increment of the water level seems to be successfully compensated by increasing the open position of CV 1.

If qualified operators have to establish an appropriate decision criterion by integrating several kinds of information, it is evident that they may use a considerable amount of cognitive resources. In other words, although there is a difference in the level of depth, determining an appropriate decision criterion can be accomplished by performing a set of high-level cognitive activities (such as identification, interpretation, decision, etc.) that belong to the primitive behaviors of a process control task (Table 3.1). In addition, it is assumed that the amount of cognitive resources demanded from these cognitive activities can be explained by a series of well-defined units of thought (Campbell and Gingrich 1986; Jiang and Klein 2000; Johnson and Payne 1985; Shugan 1980; Sintchenko and Coiera 2002). This assumption seems to be empirically supported, although experimental data have been collected from unqualified operators, because Park et al. (2008) observed that their performance seems to vary with respect to the level of engineering decision. Consequently, we can say that the level of an engineering decision is one of the significant factors complicating the performance of proceduralized tasks. A more detailed explanation about the level of an engineering decision is given in Sect. 6.3.3.

## 3.4 Where Is the Starting Point?

Till now, the nine categories of complexity factors have been discussed from the point of view of a process control task. Roughly speaking, these categories can be regrouped as depicted in Fig. 3.7.

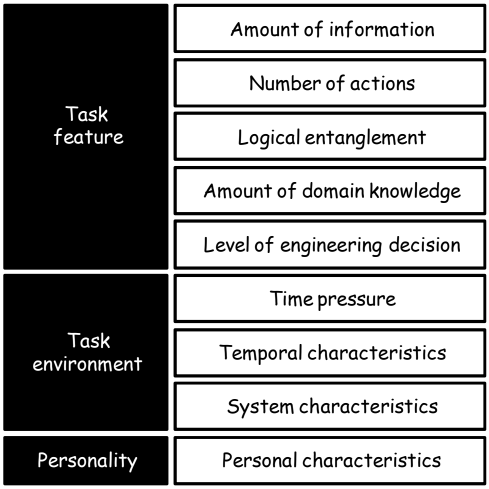



**Fig. 3.7** Three groups of task complexity factors

The first group contains several categories pertaining to task features that can be characterized from a proceduralized task itself. For example, the amount of information as well as the number of actions can be easily obtained after a proceduralized task has been determined. In addition, it is expected that three categories of complexity factors, such as logical entanglement, the amount of domain knowledge, and the level of an engineering decision, can be extracted from the given proceduralized task. This strongly suggests that there will be a deterministic framework by which the effect of complexity factors on the performance of proceduralized tasks can be dealt with.

In contrast, the second group seems to defy easy measurement in a deterministic framework because of the dynamic features of a task environment. That is, it is very difficult to measure the effect of a task arrival rate, which belongs to the category of *temporal characteristics*, on the performance of proceduralized tasks because it would vary in the form of a continuous as well as a cumulative pattern over time. In addition, this effect would likely vary with respect to time constraints (e.g., time pressure). Accordingly, a stochastic framework would be necessary to reflect the varied effects of complexity factors belonging to these categories. Similarly, due to the diversity of personalities, a stochastic framework should be used to ponder the effect of a personality on the performance of proceduralized tasks.

Consequently, in measuring the effect of complexity factors on the performance of proceduralized tasks, it is reasonable to start from easy and tangible features first. Therefore, five categories of complexity factors that are closely related to *task features* are worth considering first. This implies that the systematic framework we are trying to develop can be regarded as a kind of static (as well as objective) complexity measure. That is, if it is possible to characterize all five complexity factors without reference to any dynamic (e.g., temporal characteristics) or subjective (e.g., personal characteristics) constituents, the result of the developed framework would represent the *verbatim complexity* of a proceduralized task to be given to every qualified operator who has to accomplish it.

## References

Allen JA, Teague RC, Carter RE (1996) The effects of network size and fault intermittency on troubleshooting performance. IEEE Trans Syst Man Cybern A: Syst Hum 26(1):125–132

Boy GA, Brito G (2000) Toward a categorization of factors related to procedure following and situation awareness. In: International Conference on Human Computer Interaction in Aeronautics (HCI-Aero’00), Toulouse

Campbell DJ (1988) Task complexity: a review and analysis. Acad Manage Rev 13(1):40–52

Campbell DJ, Gingrich KF (1986) The interactive effects of task complexity and participation on task performance: a field experiment. Organizat Behav Hum Decis Processes 38:162–180

Ditlevsen O (2003) Decision modeling and acceptance criteria. Structu Saf 25:165–191

Jacko JA, Salvendy G (1996) Hierarchical menu design: breadth, depth and task complexity. Percept Mot Skills 82:1187–1201

Jiang JJ, Klein G (2000) Side effects of decision guidance in decision support systems. Interact Comput 12:469–481

Johannsen G, Levis AH, Stassen HG (1994) Theoretical problems in man-machine systems and their experimental validation. Automatica 30:217–231

Johnson EJ, Payne JW (1985) Effort and accuracy in choice. Manage Sci 31:395–414

Jonassen DH (2000) Toward a design theory of problem solving. Educat Technol Res Develop 48(4):63–85

Kieras D, Polson PG (1985) An approach to the formal analysis of user complexity. Int J Man-Machine Stud 22:365–394

Leitch R, Gallanti M (1992) Task classification for knowledge-based systems in industrial automation. IEEE Trans Syst Man Cybern 22:142–152

Leplat J (1998) Task complexity in work situations. In: Goodstein LP, Anderson HB, Olsen SE (eds) Tasks, Errors and Mental Models, Taylor and Francis, London, pp.105–115

Li K, Wieringa PA (2000) Understanding perceived complexity in human supervisory control. Cognit Technol Work 2:75–88

Liao C, Palvia PC (2000) The impact of data models and task complexity on end-user performance: an experimental investigation. Int J Hum-Comput Stud 52:831–845

Maynard DC, Hakel MD (1997) Effects of objective and subjective task complexity on performance. Hum Perform 10(4):303–330

Morris NM, Rouse WB (1985) Review and evaluation of empirical research in troubleshooting. Hum Factors 27(5):503–530

Park J, Jung W, Jung K (2008) The effect of two complexity factors on the performance of emer-

gency tasks - an experimental verification. Reliabil Eng Syst Saf 93:350–362
Rouse WB (1978) Human problem solving performance in a fault diagnosis task. IEEE Trans Syst Man Cybern 8(4):258–271
Rouse WB, Rouse SH (1979) Measures of complexity of fault diagnosis task. IEEE Trans Syst Man Cybern 9(11):720–727
Schmuck P Gundlach W (1989) Reduction of mental effort in tasks of different complexity. In: Klix F, Streitz NA, Waern Y, Wandke H (eds) Man-computer Interaction Research. Elsevier, Amsterdam
Shugan SM (1980) The cost of thinking. J Consumer Res 7(2):99–111
Sintchenko V, Coiera E (2002) Which clinical decision benefit from automation? A task complexity approach. In: Surjan G, Engelbrecht R, McNair P (eds) Proceedings of MIE2002, IOS Press, Amsterdam, pp.639–648
Spangler WE and Peters JM (2001) A model of distributed knowledge and action in complex systems. Decis Support Syst 31:103–125
Stassen HG, Johannsen G, Moray N (1990) Internal representation, internal model, human performance model and mental workload. Automatica 26(4):811–820
Sundstrom GA (1993) Towards models of tasks and task complexity in supervisory control applications. Ergonomics 36:1413–1423
Svensson E, Angelbrog-Thandrez M, Sjoberg L, Olsson S (1997) Information complexity – mental workload and performance in combat aircraft. Ergonomics 40:362–380
Thelwell PJ (1994) What defines complexity? In: Robertson SA (ed) Contemporary Ergonomics: Ergonomics for All, Taylor and Francis, London, pp.89–94
Turk Z (2001) Multimedia: providing students with real world experiences. Automat Construction 10(2):247–255
Van Eekhout JM, Rouse WB (1981) Human errors in detection, diagnosis and compensation for failures in the engine control room of a super tanker. IEEE Trans Syst Man Cybern 11(12):813–816
Wright P, Pocock S, Fields B (1998) The prescription and practice of work on the flight deck. In: Green TRG, Bannon L, Warren CP, Buckley J (eds) 9th European Conference on Cognitive Ergonomics, Limerick University Press, Limerick, Ireland, pp.37–42
Wood RE (1986) Task complexity: definition of the construct. Organizat Behav Hum Decis Processes 37:60–82
Wood RE, Locke EA (1990) Goal setting and strategy effects on complex tasks. Res Organizat Behav 12:73–109