



Available at  
[www.ElsevierComputerScience.com](http://www.ElsevierComputerScience.com)

POWERED BY SCIENCE @ DIRECT®

Int. J. Human-Computer Studies 61 (2004) 259–297

International Journal of  
Human-Computer  
Studies

[www.elsevier.com/locate/ijhcs](http://www.elsevier.com/locate/ijhcs)

# Towards a novel interface design framework: function–behavior–state paradigm

Y. Lin\*, W.J. Zhang<sup>1</sup>

*Advanced Engineering Design Laboratory, Department of Mechanical Engineering,  
University of Saskatchewan, 57 Campus Dr., Saskatoon SK, Canada S7N 5A9*

Received 17 September 2003; accepted 20 November 2003

---

## Abstract

In designing a human–computer interface (interface for short) for a complex work domain, the first question to be answered is what information should be presented on the interface display. The simplest answer may be: it depends on tasks to be performed by the human operator. In the past two decades, several studies towards a satisfactory answer to this question have been reported in literature, among which a study called ecological interface design framework is most sound. Motivated by a discussion with a nuclear power plant builder (in Canada) five years ago, we have conducted a study on the interface framework and obtained very interesting results. One of the salient findings is that the current implementation of the notion of the abstract function in the ecological interface design framework is worthy of further exploration. More fundamentally, one of its basic methods, called the five-level abstraction hierarchy used for work domain analysis, can be more commented on its architecture. Our findings are based on a critical analysis of published articles on the ecological interface design framework. We further postulated an alternative framework called function–behavior–state (FBS). We have conducted an experiment to compare these two frameworks, which positively supported our findings. The present article reports the critical analysis of the ecological interface design framework and describes the FBS framework. The experimental study has been reported separately in this journal.

© 2004 Elsevier Ltd. All rights reserved.

---

\*Corresponding author. Tel.: +306-966-5478; fax: +306-966-5427.

E-mail addresses: [yil063@mail.usask.ca](mailto:yil063@mail.usask.ca) (Y. Lin), [wjz485@enrg.usask.ca](mailto:wjz485@enrg.usask.ca) (W.J. Zhang).

<sup>1</sup> Also for correspondence.

## 1. Introduction

*Interface* is defined as the computer-based means by which workers obtain information about, and control the state of, a sociotechnical system and it is composed of displays and controls (Vicente, 1999). *Controls* are the part of an interface that is used by workers to act on automation or on the work domain (Vicente, 1999). *Displays* are the part of an interface that is used by workers to obtain information about a sociotechnical system (Vicente, 1999). The design life cycle of interface undergoes: (1) determine information content, (2) layout information content within windows and (3) design the presentation of information content. Wickens (1992) suggested that the interface design be a part of the plant system design. However, in practice the suggestion has yet to be followed; the interface design is still conducted as a post-plant design activity.

An interface can be viewed as an artefact. As any artefact is designed, principles or guidelines are needed to ensure a designed interface with rationality and systemacity. A framework is a set of such principles or guidelines. If this set addresses all the life-cycle issues, the corresponding framework may be called the interface life-cycle design framework; otherwise a framework may only address one or two interface life-cycle issues. The interface framework (framework for short) in this article only addresses the first life-cycle issue.

In the past two decades, several studies towards a satisfactory solution to this issue have been reported in literature, among which a study called ecological interface design framework is most sound. Motivated by a discussion with a nuclear power plant builder (in Canada) 5 years ago, we have conducted a study on the framework and obtained very interesting results. One of the salient findings is that the current implementation of the notion of the abstract function in the ecological interface design framework is worthy of further exploration. More fundamentally, one of its basic methods, called the five-level abstraction hierarchy used for work domain analysis, can be more commented on its architecture. Our findings are based on a critical analysis of published articles on the ecological interface design framework. We have further postulated an alternative framework called function–behavior–state (FBS). We have conducted an experiment to compare these two frameworks, which positively supported our findings. The present article reports the critical analysis of the ecological interface design framework and described the FBS framework. The experimental study has been reported separately in this journal.

The organization of this article is as follows. Section 2 gives an example, dual reservoir system simulation (DURESS), for facilitating the following discussion of the paper. This example was first described by Vicente (1991). Section 3 introduces the theory of the ecological interface design, which includes (1) the history of this theory, (2) the application of this theory to interface design for process plants and (3) the summary of experimental studies towards the validation of this theory. Section 4 gives a critical analysis of the related experimental studies that aimed to validate this theory. The critical analysis is important because it helps formulate an industrial standard to guide the development of an interface for operation safety in the application area such as the nuclear power plant. The discussion in this section leads

to the finding that the most controversial issue with the current application (or implementation) of this theory to interface design for process plants with special reference to the example used in this article is how the abstract function is displayed, what role the abstract function is with the proposed display. Section 5 presents a theoretical analysis of the roles of the current method to display the abstract function under the normal control operation, the fault detection operation and the fault diagnosis operation tasks, respectively. These three tasks are generalized to all operation tasks in process plants. Sections 6–9 propose an alternative framework based on a well-known paradigm in many science and engineering disciplines, i.e., the FBS paradigm. Section 10 illustrates how the FBS framework works using the DURESS plant as an example. Section 11 discusses general strategies for the display of information versus the operational tasks within the FBS framework. Section 12 compares the two frameworks theoretically. Section 13 gives a summary with discussion. It is generally remarked that an experimental study which has provided a verification of the theoretical discussion presented in this article should refer to [Lin et al. \(2003\)](#).

## 2. An example: the DURESS plant

In order to make the following discussion more concrete, a thermal–hydraulic process plant system called the DURESS is taken as an example. DURESS was initially prototyped by [Vicente \(1991\)](#) for illustrating and validating the ecological interface design framework. The structure of the DURESS plant system can be simplified and represented by a diagram shown in [Fig. 1](#). In [Fig. 1](#), VA and VB stand for input valve control, PA and PB for pump control, R for reservoir, VO for output valve control, H for heater control, T for temperature indicator, FWS for the feedwater stream and OWS for output water stream. The process of DURESS can be described as follows: The FWS is fed into the plant, split into two streams, emerging out of Reservoir R1 and Reservoir R2, respectively, forming into two output water streams (OWS1, OWS2). The goal of the process control is as follows.

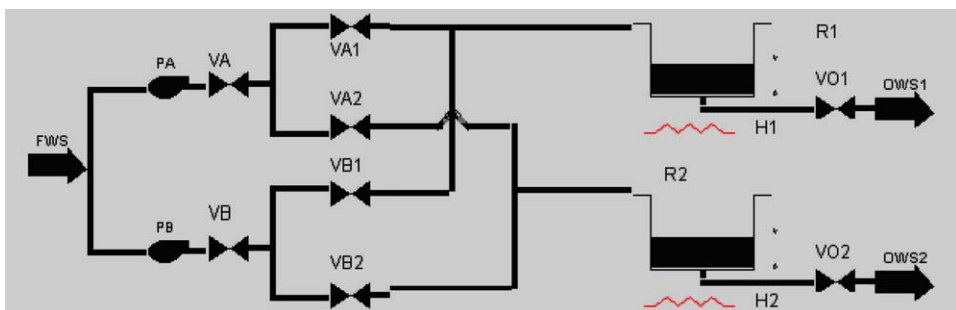


Fig. 1. DURESS plant system (the state network diagram).

*Goal 1:* the temperature of OWS1.

*Goal 2:* the temperature of OWS2.

*Goal 3:* the mass flow rate of OWS1.

*Goal 4:* the mass flow rate of OWS2.

In particular, goal 1 ( $DT_1$  for short) = 40°C; goal 2 ( $DT_2$  for short) = 20°C; goal 3 ( $DV_1$  for short) = 8 kg/s; goal 4 ( $DV_2$  for short) = 2 kg/s.

There are several operational constraints and assumptions (adapted from Vicente (1991)). Constraint 1: The water in the reservoirs can never overflow. Constraint 2: When the valve is closed the pump cannot be on. For example, if VA is off, PA cannot be on, and when VA1 and VA2 are off, PA cannot be on either. The following assumption was made: At PA (PB), VA (VB), VA1 (VA2), VB1 (VB2), R1 (R2), H1 (H2) and VO1 (VO2), sensors were in place to measure the actual physical behavior, i.e. (respectively), pressure (related to the pump), flow rate (related to the valve), level of water (related to the reservoir) and temperature (related to the heater).

### 3. Ecological interface design framework

#### 3.1. Ecological approach (Gibson, 1979)

In philosophy, there are several models proposed regarding how humans perceive things. One of these models is the ecological model (Gibson, 1979). As opposed to the stimuli response model (Zhang, 1994a), the ecological model includes the proposition that the environment that exists is the most important component in the formation of and the driving of a perception by humans. A comprehensive elaboration of various features of environments or surroundings around humans was given by Gibson (1979). However, the ecological model stayed at a level that is generic but did not provide a path towards specific application fields in manufacturing and service businesses. That is to say (further), Gibson's ecological approach to visual perception provided no explicit guideline for determining what information should be put on an interface display.

The ecological interface design framework, which was established by Rasmussen and Vicente (1989), was not simply Gibson's ecological approach, but was rooted in important work done by Rasmussen (1983) regarding the human problem-solving behavior. Rasmussen's (1983) idea was coincidentally parallel to or complementary to Gibson's ecological approach to visual perception. The following sub-sections present a detailed description of the ecological interface design.

#### 3.2. Rasmussen's two frameworks

Rasmussen (1983) proposed the 'skill, rule, knowledge' (SRK) taxonomy of levels of cognitive control as a unified framework. He further thought that during the process of human cognition of the environment (notice that at this point Rasmussen came close to Gibson), the mode was shifting among these three levels. Perhaps in

response to the urgent call from many governments to seek solutions to the problems identified from the Three Mile Island (Johnson, 2003), Rasmussen focused on fault diagnosis in an unanticipated situation in a human–machine system. Based on the observation of verbal protocols from the computer maintenance and process plant control applications, he devised a five-level of abstraction hierarchy as a general tool for work domain analysis.

These five levels of abstraction hierarchy were defined as follows (Rasmussen, 1983).

- The *functional purpose* (the purposes for which the system was designed).
- The *abstract function* (the causal structure of the process in terms of mass, energy, information, or value flows).
- The *generalized function* (the basic functions that the plant is designed to achieve).
- The *physical function* (the characteristics of the components and connections between them).
- The *physical form* (the appearance and spatial location of those components).

Rasmussen (1983) further elaborated the nature of this abstraction hierarchy as follows: “At the lower levels, elements in the description match the component configuration of the physical implementation. When moving from one level of abstraction to the next higher level, the change in system properties represented not merely removal of details of information on physical or material properties. More fundamentally, information is added on higher level of principles governing the cofunction of the various elements at the lower level”. There were two further important discussions on the theoretical foundations on Rasmussen’s five levels of abstraction hierarchy (Rasmussen and Vicente, 1989; Vicente and Rasmussen, 1992). The main argument given in these two papers was that the ecological interface design framework had a sound psychological foundation, and was in line with Gibson’s ecological approach of visual perception yet with an extension to a more complex work domain (i.e., the process plant).

### 3.3. The original ecological interface design for the process plant

Vicente et al. (1995) presented a more practical version of ecological interface design framework. The five levels of abstraction hierarchy and how they were captured in a visual form, together with the DURESS plant, are presented in Fig. 2.

Fig. 2 is an interface resulting from the application of the ecological interface design framework (i.e., largely, five levels of abstraction hierarchy; note that there appears to be little evidence that the derivation of the ecological interface described below necessarily calls for the notion of the SRK cognition control). The functional purpose is captured and displayed by the state variables, i.e., the desired flow rate DV1, DV2 and temperature DT1, DT2 setpoints. The generalized function is captured by the flow rate and the heat transfer rate, which are displayed beside the valves and the heaters, respectively. The abstract function is captured as the representation of the mass balance and energy conservation principles. The

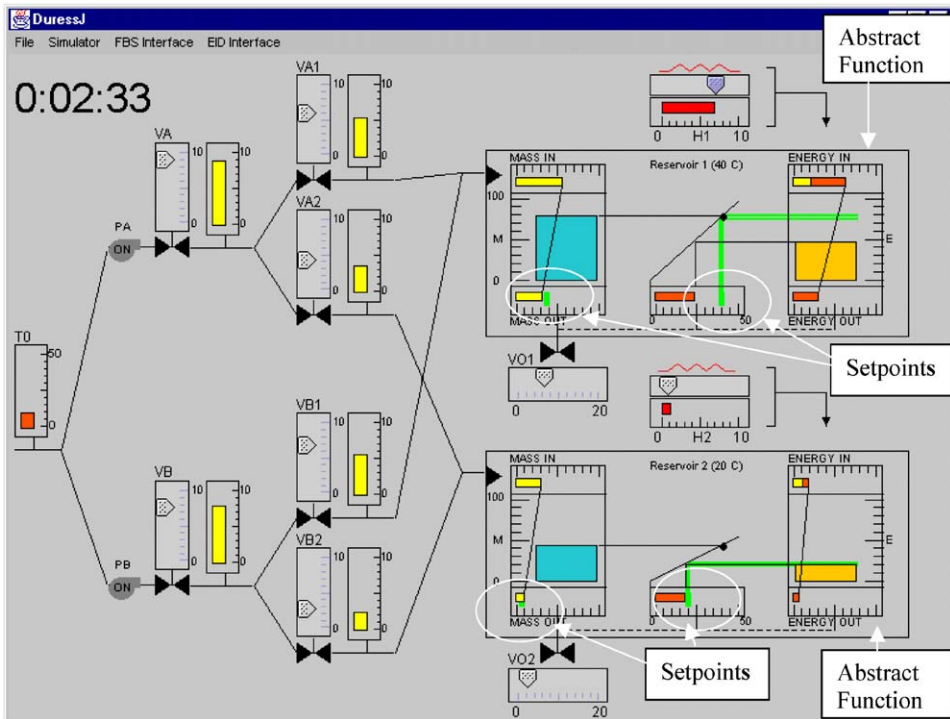


Fig. 2. EID–DURESS interface.

abstraction function for this particular plant is shown on the right part of Fig. 2. In particular, inputs are shown at the top ('mass in' and 'energy in'), and outputs are shown at the bottom ('mass out' and 'energy out'). The graph between the mass balance and the energy balance illustrates the relationship between volume, energy and temperature. The physical function is captured as the state of the pumps, valves and heaters. These states are displayed by the small triangular pointers on the respective scales. The physical form is represented by the topographic layout of the plant system (i.e., the location of the components and the connections between them). It is noted that the displays of the physical function and the physical form overlap.

On a general note, in this version of the ecological interface, no mention was given as to how the forms are designed, though the definition of the interface framework given by Vicente et al. (1995) appeared to have included the design of the visual form as a part of the issues of the framework. However, our present article is based on the definition of framework as mentioned previously, i.e., neither the layout nor the form of information content is included as a part of interface framework. Another general note is that all the quantities on the interface display seem to be sensor-measured data (which further implies the unlimited availability of sensors in the plant), though none of the ecological interface design literature explicitly mentioned

this point. In the following discussion, the unlimited availability of sensors is assumed.

### *3.4. Some variations of the ecological interface design framework/interface*

There are a few variations of the ecological interface reported in the literature. Burns (2000a) considered the display strategy in terms of timing of display (temporal factor or time condition) and location of display (spatial factor or spatial condition). Particularly, the spatial factor has two levels: (1) all five levels of information are put in one window (denoted as ‘high’) and (2) each level of information is put in a separate window (denoted as ‘low’). The temporal factor has further two levels: (1) simultaneous displaying of all four levels of information (excluding the physical form information) (denoted as ‘high’) and (2) one-by-one displaying of these levels of information (denoted as ‘low’). The following notation was used by Burns (2000a, b): high-spatial and low-temporal integration (HL), low-spatial and high-temporal integration (LH), and high-spatial and high-temporal integration (HH). It was implied in Burns (2000a, b) that the temporal factor was controlled by the operator in the sense that he or she ultimately controls what level of information is displayed.

### *3.5. Critical summary of the experimental verification of the ecological interface*

Experimental verifications of the usefulness of an ecological interface have been extensively carried out and reported in the literature. A summary of the selected reports is given below.

Vicente et al. (1995) used (1) the operator’s verbal protocol method and (2) the memory recall technique to validate the superiority of the ecological interface over the interface that contains only the physical form and physical function (the former called the P+F interface and the latter called the P interface). They showed in Experiment 1 of their study that the P+F interface is better than the P interface in terms of fault diagnosis accuracy. They argued that the improved performance with the P+F interface was not because of the added information content in the P+F interface with respect to the P interface but the highly structural information added. In Experiment 2 of their study, they studied the operator’s cognitive process in fault diagnosis using the verbal protocol and the so-called ‘canned scenario’ where the operator did not actually control the plant. They concluded that the participants who exhibited effective diagnosis performance using the P+F interface tended to initiate their routes from a higher level down to more detailed levels, as predicted. Although they did not mention specifically what was the highest level, it seems reasonable to assume that the highest level was the abstract function.

Christoffersen et al. (1996) performed the experiment to verify the effectiveness of the ecological interface for a longer duration of trials by participants. They found that the P+F interface produced more consistent performance than the P interface over the trial period, but initially the P+F interface may have been more difficult to use. They also found that the visual complexity did not degrade the performance with the P+F interface. The classes of tasks considered in their experiments are (1)

the start-up normal control operation, (2) the upgrading normal control operation and (3) the shutdown. The primary measure of performance was trial completion time.

Janzen and Vicente (1998) studied attention allocation within the abstraction hierarchy. Their approaches are summarized as follows. (1) Levels of information in the abstraction hierarchy were displayed in separate windows and the operator controlled the display of these windows. Note that this is what was described by Burns (2000a, b) as ‘low-spatial and low-temporal’ (LL for short). (2) They used two data acquisition techniques: one was to record the operator’s screen activity which was further considered by them as operator attention allocation, while the other was called the verbal protocol with which the participants spoke aloud out of their thinking processes while they were operating the plant through the interfaces. (3) The classes of tasks in their experiment were the normal control operation, fault diagnosis/compensation and fault detection tasks.

Their results showed (1) the more often participants visited the abstract function, the faster their performances on normal control operation task, (2) the more often participants visited the generalized functions and the abstract function, the more accurate their diagnosis, (3) dwelling briefly on the abstract function was related to shorter fault compensation completion time and (4) there is no such a trade-off between the diagnosis accuracy and the fault compensation completion time that participants who visited the abstract function frequently and exhibited good diagnosis accuracy (high score) may have had to dwell longer on the abstract function in order to thoroughly assess the event, thereby exhibiting longer fault compensation completion times.

Based on the above results, they further concluded: (1) The visual complexity due to the emerging features of the abstract function display may affect the rate for fault compensation completion. (2) The best performance in fault diagnosis accuracy is attributed to frequently consulting the generalized function. (3) The best performance in terms of normal control trial time (instead of fault diagnosis) is attributed to frequent visits to the abstract function. (4) The best performance in fault compensation completion time is attributed to the lesser dwell in the abstract function. (5) There is no correlation between more frequent visits to the abstract function and a reduction of fault detection time.

It is interesting to see that the *first* conclusion in Janzen and Vicente (1998), contradicted the conclusion made by Christoffersen et al. (1996); see the preceding discussion in Christoffersen et al. (1996). The *third* conclusion is not in line with the premise of the ecological interface, i.e., the ecological interface was targeted on problem-solving in an unanticipated environment (owing to the abstract function with its emerging features). The *fifth* conclusion appeared to contradict the result obtained by Vicente et al. (1995).

It is further noted that in the experiment conducted by Janzen and Vicente (1998), a technique used for the measurement of the number of visits and the dwell time was based on a simulation log file which recorded the control actions of the participants. This technique can be challenged by the fact that control actions do not completely reflect attention allocation, as attention is more in a sense of cognition—a kind of

mental thinking process. Control actions are somewhat consequences of a cognition with attention. Discussion of the relationship between control actions and attentions can be found in [Lin and Zhang \(2003\)](#). Further, using the eyegaze tracking system for attention tracking for evaluating different interfaces refers to [Lin et al. \(2003\)](#).

It is further speculated that when their data acquisition techniques for attention allocation tracking are improved (e.g., using the eyegaze facility), the result would be significantly changed, which may eliminate a ‘seemingly’ contradictory result based on the procedure from their experiment; that is, the shorter dwell time on the abstract function information led to the shorter fault compensation time; while the higher number of visits on the abstract function led to the better accuracy in fault diagnosis. Finally, their design of the interface into four windows, which were completely blocked out, was somewhat unrealistic in industry. Though this design simplified the experimental manipulation to answer their research questions, the result obtained from such an experimental setting may not be generalized to a more realistic interface setting in industry.

[Ham and Yoon \(2001a\)](#) argued that there was a need to experimentally study the role of the abstract function (AF for short) and generalized function (GF for short) by eliminating the possible effect of graphic forms of the information content with special reference to AF. They also recognized that the effectiveness of the display may be related to categories of tasks. With these considerations, they designed the following types of interfaces: (1) the P interface, (2) the P + GF (PG) interface, (3) the P + AF (PA) interface and (4) the P + GF + AF (PGA) interface. The displays of these interfaces were ‘plain’ in the sense that the metaphor to physical objects and semantics was totally eliminated (only the elementary bar graph was used). They considered fault diagnosis (which included fault detection). They used a set of performance measures. They did not take the verbal protocol measure because they were afraid of the unreliability of this measure in the sense that the participants’ mental workloads could be affected if they had to speak out their thoughts while they were operating the plant. Their results showed that the level of abstract function was positively correlated with the improvement of diagnosis accuracy only when the task was complex and demanding; however, no qualitative analysis was given of the complexity of tasks.

[Burns \(2000a, b\)](#) changed the direction of some of the research on the ecological interface. She considered the effect of the temporal and spatial display of five levels of information. In particular, she defined high-spatial and high-temporal (HH), high-spatial and low-temporal (HL) and low-spatial and high temporal (LH) modes of information display (see also the discussion in Section 3.4). Interestingly, she thought that the low-spatial and low-temporal (LL) mode was not practical, which might jeopardize the legitimacy of the study conducted by [Janzen and Vicente \(1998\)](#) where all their data and result were based on the low-spatial and low-temporal (LL) mode (see also the previous discussion on [Janzen and Vicente \(1998\)](#)). [Burns \(2000a, b\)](#) used the video tape recording of the operator’s screen activity, which is similar to the log file recording of the operator’s screen activity in [Janzen and Vicente \(1998\)](#) for exploring the cognitive process of the participants and the performance measures (fault detection time, fault diagnosis accuracy and fault diagnosis time). [Burns](#)

(2000a, b) found that (1) the fault detection was the fastest in the HL mode, (2) the fault diagnosis was the fastest in the HH mode and (3) the best performance of fault diagnosis accuracy occurred in the HH mode.

The validation of the ecological interface design framework has been attempted in some relatively large industrial applications. Jamieson and Vicente (2001) presented a comprehensive discussion of the application of the ecological interface design framework to a petrochemical plant. However, implementation appeared to be not experimentally done. Though, the application was indeed a large problem, they have not provided a compelling case to show explicitly what modification was needed on the ecological interface design. Another recent effort in applying the ecological interface to nuclear reactor system was made by Yamaguchi and Tanable (2000) on a simulated reactor system. The verification of this implementation was not accomplished. They concluded upon empirical evaluation (not reported in detail) that higher level information such as energy balance should be represented together with lower level information such as raw measuring data with a single graphical display format in a way to explicitly show how relevant raw measuring data are integrated to the higher level information item. It may be further noticed that the current interface design guidelines for the nuclear power plant control room (NUREG-0711, 1994; Penington, 1995; IEEE std 1289, 1998; CNSC G-276, 2003) have not explicitly included the notion of the abstract function.

#### **4. Critical analysis of the existing studies of the ecological interface design**

The preceding discussion has shown that there are considerable variations in the experimental studies toward the validation of the effectiveness of the ecological interface design framework. This section aims to explore possible reasons for these variations.

##### *4.1. Controversial issues*

It is no doubt that a display of physical function information alone is not enough, regardless of the visual forms used. There has been consistent verification of the need to display the generalized function in the ecological interface design framework. In practice, similar kinds of information with the generalized function information were used in the nuclear power plant control room before the ecological interface was proposed. For example, in the Three Mile Island accident, the water level was observable by the operator, and the water level is a kind of generalized function information. The most controversial issue with the ecological interface is the display of the abstract function information, and the question of particular interest is how and when to display it. This when-issue concerns whether, for all categories of tasks, there is a need to display the abstract function given the result of the existing study since (1) the visual complexity due to the display of the abstract function may affect the performance, and (2) the abstract function as displayed now, is not positively correlated with the improvement of performance for all the categories of tasks.

Closely related to this issue is the (negative) effect of visual complexity brought in when the abstract function information is displayed.

It is by intuition that in a task such as monitoring (or fault detection), the symptom was more easily captured, and this seems to imply that the abstract function in such a task situation may be less useful. Ham and Yoon (2001a) commented "... In the experiment, the PG showed a notably better performance than the PA display... . Another, and simpler, explanation is also possible. The diagnosis may have required less use of AF-level information than GF-level information in experiment ..." (p. 208). This comment is in strong agreement with the finding, i.e., conclusions obtained by Janzen and Vicente (1998).

#### *4.2. The effect of task*

Variability in many experimental results related to the ecological interface is closely related to the categories of tasks and also to the difficulty levels of tasks in each of these categories. For instance, Ham and Yoon (2001a) showed that the best performance by operators was achieved by displaying the generalized function (note that the physical function and the functional purpose are always displayed) for a certain kind of fault diagnosis task, which is related to component failure. Further, they showed how the increase of task difficulty levels affects the usefulness of the abstract function; i.e., they considered that the difficulty level of tasks in their Experiment 2 was much higher than that of tasks in their Experiment 1. However, careful examination of tasks for Experiment 1 and for Experiment 2, respectively, failed to provide sufficient information about the quantitative measure of the task difficulty level for the tasks in the two sets. Ham and Yoon (2001a) did mention or imply that in Experiment 2, the task scenario was a mix of normal control operation and fault diagnosis, which in the words of the authors is more 'demanding'; this demanding effort was responsible for the increased level of usefulness of the abstract function. However, it could be argued that task demand and task difficulty are different categories of concepts; they are not necessarily additive to one another.

#### *4.3. The effect of variation in defining an interface framework*

By definition, the interface framework is responsible for determining what information content needs to be displayed and when to display them. There is one more issue in interface design; that is, how to display information content. This issue includes the information content layout and the form of information content. Interface design theory and methodology refer to all these issues. It has been found that the results produced from existing studies may not be 'unanimously' based on the same definition of interface framework. As a result, some variation in the reports of studies may occur.

In Ham and Yoon (2001a), the framework appeared to be more concerned with answering the what-question, while many other studies are more or less concerned about the effects of factors other than the information content (what-to-display issue). For example, Burns (2000a) considered the layout of information (not exactly

the same as that discussed in Wickens and Carswell (1995)) and the when-to-display (the control of the display was at the disposition of the operator). The results generated by Ham and Yoon (2001a) and by Burns (2000a) are therefore not easily comparable. Janzen and Vicente (1998) explained their ‘contradictory’ result regarding fault diagnosis accuracy and fault compensation time as the cause of the ‘graphic emergent features of the Principles display’ (p. 539). This has implied that the factor of display can significantly influence the result.

There is one other concern in experimental verification of the ecological interface or any interface: the reliability and the practicalness of experimental results. *Reliability* refers, in particular here, to whether the result/conclusion produced for one particular purpose (e.g., to examine information content of an interface) underlying the experiment is actually ‘contaminated’ by some other factors that are uncontrollable (e.g., the visual form of the information content of an interface). *Practicalness* refers, in particular here, to whether the experiment setting was the same as or similar to the setting used in the industrial practice. There is a dilemma presented in the reliability and the practicalness when experiments are designed. Ham and Yoon’s (2001a) work may produce higher reliable results as they eliminated the effect of visual forms of information contents, but their results and conclusions may not be generalized to the more practical situation, i.e., the display forms (which are more sophisticated) of the ecological interface as shown in Fig. 2.

#### 4.4. *The effect of data collection techniques in ecological interface experiments*

One of the ways that data was collected in many ecological interface design framework studies was by a method called the verbal protocol real time technique (Vicente et al., 1995) for tracking operators’ attention. The purpose of this technique is to explore the cognitive process of the operator in various task situations, which is complementary to many other measures towards the end result the operator performs (perhaps called the product measure). This technique is, however, relatively coarse; the participants’ attention to report their thinking may be compromised by their attention to complete the tasks they are performing. A more robust technique of such a kind, e.g., the eye movement measurement technique (Lin, 2000), would improve this situation. Indeed, the experimental result based on the eye movement parameter for a nearly similar setting as Vicente et al. (1995) and Janzen and Vicente (1998) has shown considerable difference; see Lin et al. (2003) for details. The problem with those old eyegaze facilities (Lin, 2000; Burns et al., 2002; Lin et al., 2003) is its intrusiveness in nature, which would interfere with operators’ performance as well. To solve this problem, the experiment has to be designed for a relatively short duration of trial. The latest eyegaze tracking equipment, such as the product by Seeing Machine Inc. (2003), has solved this problem; that is to say, the current eyegaze measurement technique could be nonintrusive. The other data acquisition technique used, see Janzen and Vicente (1998) and Burns et al. (2002), is the recording of the operator’s screen activity; in these cases, the operator’s screen activity is in fact regarded as the operator’s cognitive process (which is certainly not the case). This technique may be responsible for inaccurate raw data acquisition

regarding the visits and the dwells in Janzen and Vicente (1998), which may further be responsible for the likely questionable result: the longer dwells on the abstract function, the longer fault compensation completion time. Since usually longer dwell corresponds to more attention, their result may then be interpreted as the more attention to the abstract function, the longer completion time needed for fault compensation (i.e., the poorer performance in fault compensation).

## 5. Theoretical analysis of the role of the abstract function

The nature of the abstract function is the constraint equation among a set of relevant state variables. For example, in the case of the DURESS plant system, referring to tank 1 in particular, such a set of variables includes: the flow rate entering the tank (further corresponding to the two valves, VA1 and VB1), the flow rate out of the tank (further corresponding to the valve VO1), the energy entering the tanks (further corresponding to the heater) and the energy out of the tank. The constraint equation is derived from physical laws, chemistry laws and other axioms from various science and technology domains (e.g., the design axioms for design of general systems; see Suh, 1990). In the ecological interface design framework (see Vicente, 1999), the constraint equation is represented by a differential equation or some general algebraic expression as a solution to the differential equation (the latter point might just be speculation) (p. 170). From a point of view of the mathematical representation of physical and/or chemical behaviors of an underlying system, it is clear that neither the differential equation nor the general algebraic equation (in an implicit form) accurately represents the behavior of the system (per se) because the system boundary and initial conditions are not considered.

Assume that there is the presence of the generalized function and the functional purpose in the terminology of the ecological interface design framework. In the following, a theoretical analysis of the role of the abstract function in three categories of tasks is presented.

### 5.1. Role of the abstract function to the fault diagnosis operation

The abstract function currently displayed was based on a group of state variables, states of which were sensor-measured. In particular, to the DURESS plant system, four differential equations, two for mass balance and two for energy balance, for the reservoirs or tanks (R1, R2), were the constraint equations. The display of these equations actually implied that the change rates of mass (water) and of the energy for two reservoirs (respectively) were zero, which corresponds to the stable situation of the dynamic process plant. The role of such a display in terms of the fault diagnosis may be as this: when there is a fault presented in the plant directly related to the reservoir, the abstract function display may give an indication showing that the constraint is broken, but what variable or variables cause this break are generally out of the power of the constraint equation itself (note that the constraint equation aggregates multiple variables). The lack of problem diagnosis capability with the

current method of displaying the abstract function in the ecological interface is, in the view of the authors, simply because there is no display or display procedure for a generic problem-solving strategy, e.g., the analytical redundancy technique (see Frank, 1990). It is interesting to note that the above view has been supported by the experiments conducted by Janzen and Vicente (1998) and Ham and Yoon (2001a, b); both studies showed a ‘weak’ link between the accuracy of fault diagnosis and the display of the abstract function. It should be noted that domain knowledge, if well represented, forms ontology of knowledge for generating solutions to any problem with a corresponding plant in that domain. With this point in mind, despite the claims of the abstract function display proponents for problem-solving in an unanticipated situation, it was found that all the fault examples published in the literature to test the ecological interface were falling into the repository of knowledge of the domain (i.e., the process plant).

### *5.2. Role of the abstract function to the fault detection operation*

Fault means that there is at least one broken constraint event. The constraint can be in the form of the generalized function or the abstract function. In the following, three cases of faults and the role of the abstract function in detecting these faults are explained.

*Case 1.* When a fault occurs on a component that is not directly related to the storage component (the tank in Fig. 2), e.g., valve VA, its generalized function will be violated and the sensor should record an abnormal state. In this case, the fault is identified by the abnormal symptoms, not necessarily by the mass and energy imbalance that may also be shown in the sophisticated graphic representation of the abstract function (Fig. 2). It should be noted that there are two contexts before the fault occurs: (1) the plant system is in a stable situation (i.e., all the targets have been reached), and (2) the operator performs a normal (updating) operation (i.e., trying to make the plant reach its targets). Clearly, in context (2), though the fault at VA will perturb the abstract function information on the display, no clue is given for whether the imbalanced mass and energy display is because of that perturbation coming from the faulted VA, or because the plant is yet to be manipulated to achieve its targets. In context (1), the role of the abstract function information on the display is redundant.

*Case 2.* A fault exists directly related to the tank (Fig. 2), e.g., on valve VA1. First of all, the symptom of this case is the same as that of Case 1. It is true that in this case, the abstract function (mass conservation part) may cause the rate of volume change in the tank to be greater or smaller than ‘normal’ depending on the nature of the problem in valve VA1. The graphical representation of the mass conservation in Fig. 2 may show that mass-in is greater or smaller than mass-out. It is clear, however, that if an operator detects this fault not by looking at the abnormal symptom (or at the generalized function) but instead by reasoning about the graphical representation of the abstract function per se, not only extra attention (which increases the mental workload) needs to be paid, but also a wrong fault detection result could be produced (as the cause of the imbalance of the mass conservation may also be the problem of valve VO1, for example).

*Case 3.* When a fault exists on any situation other than on the state variables mentioned above (e.g., when unknown extra water pours into the tanks), the symptom can still be captured by the generalized function (i.e., the rate of volume change in the tank will be either greater or smaller than normal). In this case the abstract function constraint will not provide an indicator for the fault because the flow rate at the inlet and the flow rate at the outlet of the tank are measured by the corresponding sensors (respectively), and they are not affected. A further note explains the rationale for this conclusion. The physics behind the mass balance in the tank is: mass-in–mass-out =  $A(dH/dt)$ , where  $A$  is the horizontal cross-section area of the tank, and  $H$  is the level of the water in the tank. Therefore, when mass-in–mass-out = 0 (i.e.,  $dH/dt = 0$ ), the bar connecting mass-in and mass-out will be in the vertical position, which provides a cue to the operator about the status of the plant system around the tank. From the discussion of the physics of the constraint represented by the current display of the abstract function, we can see that the variables with their equation involved for governing the display of the abstract function do not capture the event of extra water pouring into the tank.

### 5.3. Role of the abstract function to the normal control operation

In the normal control operation, there are basically two cognitive and manipulative issues: (1) a goal of action and (2) a plan of action. The functional purpose in the ecological interface gives the information for issue (1). With regard to issue (2), one needs to examine whether the abstract function, as it is defined and displayed, could provide a clue for operators to develop an effective plan in their mental worlds with which they can effectively manipulate the controls to make the plant system reach its goal.

As discussed earlier, the abstract function in the ecological interface is expressed by a kind of differential equation (Vicente, 1999). The constraint created by the differential equation may be valid to more general situations beyond a specific situation that aligns with a specific boundary condition of a plant. This means that the general constraint could not provide an exclusive clue to a specific problem prevailing in applications. This point can be best explained by the example of the analysis of a robot mechanism. Fig. 3 shows a five-bar mechanism (Appendix A provides some more detailed discussion of its kinematic governing equation, the branch concept and the configuration concept). The two actuators are represented as the controls, and they are manipulated by operators. The control goal is to make the end-effect point  $Q$  reach a new setpoint  $\tilde{Q}$ . There are two branches with this mechanism (see Fig. 3a,  $Q$  and  $\tilde{Q}$ ). The kinematic loop constraint equation for this mechanism is  $\vec{l}_1 + \vec{l}_3 = \vec{l}_5 + \vec{l}_2 + \vec{l}_4$  (Shigley and Uicker, 1995). From kinematics, it is known that both branches ( $Q, Q'$ ) (Fig. 3a) satisfy this loop constraint equation; In other words, this loop constraint equation could not distinguish branches (see Appendix A for details).

The question now is: Is there any unique role with the abstract function in the aspect of producing an effective action plan in the operators' mental world? The

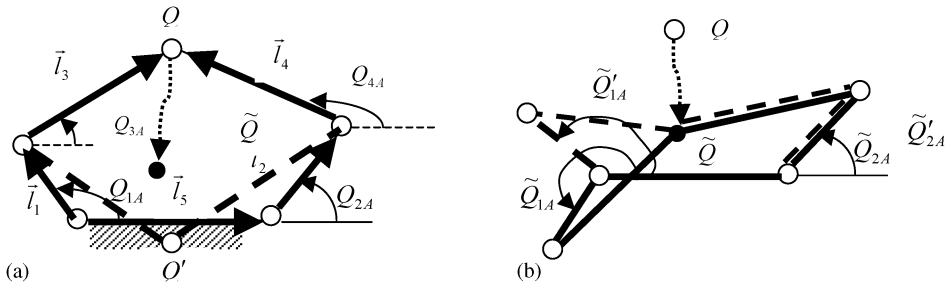


Fig. 3. A five-bar robot mechanism. (a) The schematic kinematic diagram (the dashed line represents a different branch other than that represented by the solid line; specifically the solid line refers to branch *A*, while the dashed line refers to branch *B*.  $Q_{1A}$ ,  $Q_{2A}$  are active state variables and  $Q_{3A}$ ,  $Q_{4A}$  are passive state variables. (b) Two configurations (the solid line: branch *A*; the dashed line: branch *B*) that both satisfy the loop constraint equation. Furthermore,  $\tilde{Q}_{1A}$ ,  $\tilde{Q}_{2A}$ : the active state variables when the target is reached from branch *A*;  $\tilde{Q}'_{1A}$ ,  $\tilde{Q}'_{2A}$ : the active state variables when the target is reached from branch *B*.

abstract function may play a role in creating information to operators about how ‘far’ the current state of the plant is away from the target state which is defined as a new setpoint ( $\tilde{Q}$  in Fig. 3) and as DT, DV in the DURESS plant. This role is fulfilled by showing, for example, mass-in greater or smaller than mass-out in the case of the DURESS plant, or by showing that the loop constraint equation is violated in the case of the robot mechanism. However, a direct examination of the difference between the flow rate of the water at the outlet of valve VO1 and the desired flow rate DV1 should be far more effective and efficient in fulfilling that role. For the robot mechanism (Fig. 3), a direct examination of the difference between the coordinates of  $Q$  and the coordinates of  $\tilde{Q}$  is more effective and efficient in fulfilling that role.

As to the question of whether the abstract function could give operators a type of effective path to reach the target goal (for example, in the case of the robot mechanism driven by two actuators ( $Q_{1A}$ ,  $Q_{2A}$ ) whether  $Q_{1A}$ ,  $Q_{2A}$  should be increased or decreased, for  $Q$  to be adjusted to  $\tilde{Q}$ ). The loop constraint equation, corresponding to the abstract function, does not provide any means preventing operators from changing the branch while they are manipulating the controls (i.e.,  $Q_{1A}$  and  $Q_{2A}$ ) to achieve the target (i.e.,  $\tilde{Q}$ ). Particularly, let us assume that the mechanism starts with the solid line schematic diagram in Fig. 3a, which is known as branch *A*. Two configurations, one corresponding to branch *A* (which is expected), and the other corresponding to branch *B* (which is not expected), can reach the target  $\tilde{Q}$ . It is entirely possible that the operator manipulates the controls and ends up with the configuration of branch *B*.

In conclusion, the abstract function, as displayed in the literature, plays a redundant role with respect to the functional purpose in issue (1) in the normal control task, and is possibly insensitive to the incorrect action plan, i.e., issue (2), which may be executed by operators. This latter point is in fact aligned with the second principle for interface design elaborated by Vicente and Rasmussen (1992), which requires that an interface should be designed as providing a consistent

one-to-one mapping between the work domain constraint and the cues or signs provided by the interface. Nevertheless, one may argue that the one-to-one mapping is associated with an interface which is not directed to solving problems in an unanticipated situation; the ecological interface sets a higher goal for problem-solving in an unanticipated situation, so the display of the abstract function comes into use. The counter-argument is simple: the task situation here is not so much in the sense of ‘unanticipated’. There is another view that may support the above counter-argument. Rasmussen (1983) associated the unanticipated situation with the knowledge-based problem-solving strategy in which humans need to establish a hypothesis and then to test it with the process continuing until the problem is solved. It is easy to see that in the normal task operation, the need of such a process (i.e., establish a hypothesis and then test it) is not justified.

## 6. Further discussions of ecological interface design

Indeed, the ecological interface falls into the category of the ecological approach (Gibson, 1979) and goes beyond direct perception in the sense that the ecological interface takes the inference as a significant process for fault diagnosis for complex systems. To build up this capability, the ecological interface design framework is based on two seminal concepts: (1) the SRK problem-solving strategy and (2) the five levels of abstraction hierarchy. Subsequently, three principles for interface design were proposed (Vicente and Rasmussen, 1992). The first one corresponds to the skill-based principle, the second one to the rule-based principle, and the third one to the knowledge-based principle. In order to support the knowledge-based principle, five levels of abstraction hierarchy, which represent the constraints of a work domain at various levels, are followed for designing an interface. The ecological interface design framework was aimed at designing an interface with which the operator can cope with faults in an unanticipated environment.

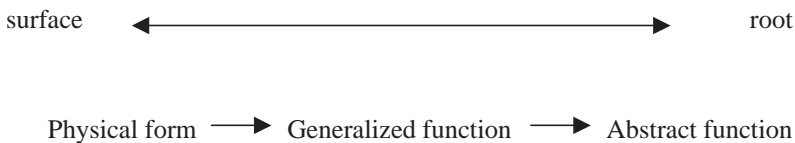
The ecological interface design framework was considered as a powerful tool for work domain analysis with its five levels of abstraction hierarchy. While the notion of abstraction hierarchy is very general, it is not an optimal solution to a work domain that has a clear structure (e.g., process plants). Theoretically, the abstract function should be applied to system, sub-system and component. In practice, the abstract function was only seen applied to a so-called storage component. There is a great deal of confusion regarding the applicability of the abstract function. For example, Janzen and Vicente (1998) considered the abstract function applicable to the system and sub-system, while Burns (2000b) considered the abstract function applicable to components as well. The mention or implication that the abstract function was only applied to those so-called ‘storage’ components is only seen from Vicente (1999).

One needs to notice that the ecological interface design framework, as originally published in Rasmussen (1983), Rasmussen and Vicente (1989), and Vicente and Rasmussen (1992), can be different from the ecological interface (Vicente et al., 1995); the latter is actually an implementation of the principle of the ecological

interface design framework. The current display of the abstract function in the ecological interface is of little use to the fault diagnosis task, yet it is of some redundant use to the fault detection and normal control operation tasks. This is because the current method for such a display is very much like the constraint satisfaction paradigm where the constraints are represented for domain structure or knowledge. Violating the predefined constraints implies that an error occurs (i.e., the fault detection). It is known that there would be multiple solutions satisfying the constraint. The current ecological interface leaves the screening of the multiple solutions into one 'correct' solution (which corresponds to the fault diagnosis) to the operator. In conclusion, it is worthwhile to explore an alternative interface framework which may improve the ecological interface design framework. In the subsequent sections, we introduce a new interface framework.

## 7. The rationale behind an alternative interface design framework

The current implementation of the ecological interface design framework and its experimental verification were elaborated above. This elaboration has shown that the current implementation of the ecological interface design framework is by no means the most suitable one for the process plant operation application, and remains controversial and warranting further rigorous studies. The ecological interface design framework is rooted in problem-solving philosophy (especially together with the SRK framework). As such, the model for interface design is more adaptable to the problem-solving strategy that exhaustively explores the root of causes leading to faults, namely the cognitive process below.



There is a need to develop an interface design framework that starts from the plant design model and facilitates a model-based plant operation/maintenance strategy (i.e., the problem-solving is built upon the plant design model). Plants are designed, made and assembled by human designers and manufacturers. The operators perform tasks (cognitive and/or physical) on the plants through the interface to make the plant achieve the desired goals that are set by the designer and agreed upon by various manufacturers involved in the development of the plants. There is another functional unit in the operation of the plants, i.e., the maintenance personnel. The maintenance personnel are closely related to the operation personnel (i.e., operators) and their error-free communications for safe and effective plant operation are extremely important. For example, one of the errors in the Three Mile Island accident was the gap in the understanding of the states of block valves that blocked the flow of water from the emergency feedwater pumps between the operation

personnel and the maintenance personnel; specifically, the latter closed the valves for a maintenance purpose, but the former thought that the valves were open. That said, this problem can only be resolved by (1) having a common model of the plant available all the time along the plant life cycle, (2) letting plant personnel access this common model at any time and at any place through the interface (if deemed necessary) and (3) ensuring the information to be accurate and fresh in the plant model.

Another point of evidence arising from the Three Mile Island accident in supporting of the above argument is the critical gap in the understanding of the state of a pressure relief valve that opened as pressure built up within the primary system due to the rising of heat. In particular, the designer of the plant devised the indicator for this valve corresponding to what the operator commanded, while in the operator's mind the indicator represented its actual state (i.e., the state is sensor-measured). The rationale for the designer to make such a design decision on this valve might have been such that (1) this valve is used for the situation where the pressure in the primary system is in excess, (2) the pressure would reduce to a certain level when the valve is open and (3) the mechanical design of this valve system ensures that this valve would close right after its relief (i.e., open). The designer knew that the actuation of this relief valve to close after it was opened was not reliable, but the designer could not foretell that the false indicator of this valve (i.e., open in reality while close according to the indicator) could lead to the subsequent dangerous situation. The operator, worried that the relief valve might go full (or 'solid') when the water rose to fill it, subsequently decided to draw off the water in the primary system when to the contrary the system actually desperately needed the water. If the operator had been given the opportunity to know the design rationale, the operator would likely have checked the actual pressure in the primary system and examined whether this actual pressure should keep the relief valve open after it was opened. It was then possible that the series of wrong actions (i.e., drawing the water out of the primary system) could have been avoided. Operators' knowledge of the design rationale can only be made possible when the operator can gain access to the plant design model through an interface, which further requires that the interface must be designed to have this capability. The problem regarding the relief valve could also have been avoided if the information about the state of the plant appearing on the interface display had been sensor-measured in the normal plant operation, and the operator had been trained to know this.

The discussion above implies that the interface design framework should start from the plant design model, which is a key to computer aided plant system life cycle management (Zhang, 1998). There have been various such models proposed in the design research community in the past decade (Zhang, 1994b). The most general model is the one first described by Umenda et al. (1990) called the FBS model. The next section will give full details of this model.

An added benefit gained from the above idea is that the design of an interface can be integrated with the design and manufacture of a plant system. By considering hiring, training and deploying human operators, including the determination of operator cognitive level, the operator physical level, the number of operators and the

task scopes of operators (or human resource management for short) as a part of the design of the human–plant system, the three activities of design (i.e., design of a plant, design of an interface and human resource management) could possibly be carried out concurrently. This would then avoid unnecessary design iterations and to at best could eliminate the communication gap between the operation personnel and other personnel involved in the life cycle of a plant.

## 8. The FBS framework

There are various models proposed for the design of a general system (including a plant system) such as the Axiomatic Design Theory (Suh, 1990), the Systematic Design Methodology (Pahl and Beitz, 1995), the FBS model (Umenda et al., 1990), and the Integrated Bond-Graph and Function-Means Methodology (Bracewell and Sharp, 1994). The FBS model was adopted with some modification in this study as the basis for the interface framework. In fact, the proposed framework is parallel to the FBS design model; that is to say, the core concepts of the FBS design model are the core concepts of the FBS framework. In the following, a set of core concepts of the FBS framework, largely reshaped from those of the FBS model, is described.

### 8.1. Structure, state and state variable

A *system* has a structure that is a set of entities connected in a meaningful way. Entities are perceived by a set of properties, and these properties are called the *states*. The properties and the states are given a name. The name of the state is the *state variable*. The state variable can be further divided into the active state variable and the passive state variable. The *active state variable* corresponds to the actuation entity in a physical system that can be directly manipulated by the operator through an interface. The active state variable can also be called the *control*. The *passive state variable* depends on the active state variable. Such dependency is governed by the ‘principle’ (see later discussions). In a human–machine interface for plants, the operator manipulates the active state variable, not the passive state variable. To a particular plant system, its state variables and their dependencies are determined when the plant system is systematically designed by following a particular design theory and methodology, such as the FBS model.

### 8.2. Behavior

The *behavior* is the causal relationship or structure among a set of related state variables/states. Such kinds of relationships have two parts: intensional and extensional. The intensional part of behavior is represented by algebraic or differential equations. The extensional part of behavior is an exhibition of states of the related state variables. The extensional part of behavior is either *actual* in the sense that the passive state is known based on the measurement by the sensor in the structure, or *expected* in the sense that the passive state is calculated based on

the constraint equations. There are two situations in which the states are related: for example, (a) if State 1 changes from 0.3 to 0.4, State 2 will change from 5 to 6; (b) at time  $t$ , State 1 is 0.3, and after  $\Delta t$ , State 1 will become 0.5. In machine systems design, situation (a) is governed by kinematics principles, while situation (b) is governed by kinetics or dynamics principles. The present version of the FBS framework assumes that the sensor is reliable and perfect (free of noise).

### 8.3. Principle

The *principle* governs or accounts for the behavior in the way that the causal relationship or structure is derived from the principle; for example, Newton's laws are used to derive the dynamic model of a robotic system. The principle can be viewed as the constraint that restricts the actual exhibition of a particular behavior. The constraint is further expressed in a form of mathematical equation that contains variables (i.e., the state variable mentioned). There are two types of constraint equations for the engineering problems (e.g., the process plant): (1) implicit form and (2) explicit form. The implicit form of constraint equation may have multiple solutions, while the explicit form of constraint equation is one of the multiple solutions of the implicit form, which is the one 'closest' to the initial configuration. Because a plant system works in a particular configuration, the explicit form of the constraint equation should be the most directed governor to the plant.

### 8.4. Function

The *function* is defined as a purpose in the mind of humans and can be realized by the system (structure) through the provision of certain behaviors by the structure. The function has two essences: (1) the intensional and (2) the extensional. The intensional essence of the function is the 'qualified' behavior of a system. The extensional part of the function is the exploiting of the intensional part of the function, purposely governed by 'effect' (physical or chemical effect for a physical system; social effect for a social system; combined effect for a sociotechnical system). The semantics of the function are given by an assertion which has the following syntactic form: Function := verb | noun | [proposition] | [value 1] | [proposition] | [value 2], where the notation '[' ]' means optional, and the notation ':=' means the assertion. For example, a system has the function to generate the flow rate of water from 4 to 8 kg/s.

### 8.5. FBS architecture

The above concepts are related in a definite way, as shown in Fig. 4. It can be seen from Fig. 4 that the structure concept is located at the bottom, followed by the state, the behavior and the function concepts. The principle concept is situated between the state concept and the behavior concept in order to give rationale for constraint equations such that given a set of values of the active state variables, the passive states are found through the evaluation of the constraint equations.

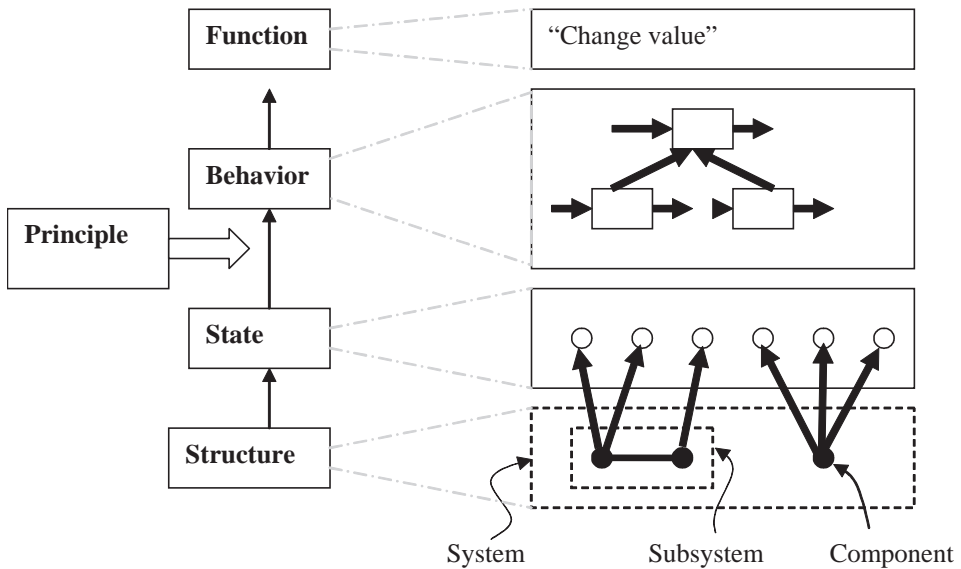


Fig. 4. The FBS framework.

### 8.6. System decomposition

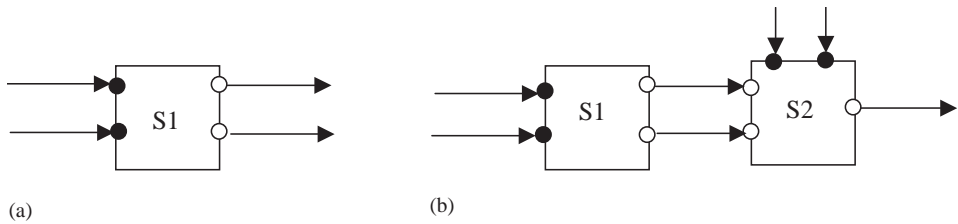
It is further noted that the behavior, the principle and the function concepts follow system decomposition. This means that it makes sense to speak of the behavior (function, principle) of a component (or sub-system, system). This note has been implied in the section to the right in Fig. 4.

## 9. The FBS interface design

Designing interfaces should result in a display of a set of widgets. Widgets are windows through which humans and machines can communicate. Widgets carry semantics of information (data and/or knowledge). Widgets are displayed in certain ways by following certain principles (Wickens, 1992). In the following, the display of the core elements of the FBS framework (not including issues such as shapes and colors of widgets) is discussed. The DURESS plant is used as an example for facilitating the discussion.

### 9.1. Display of the state/structure

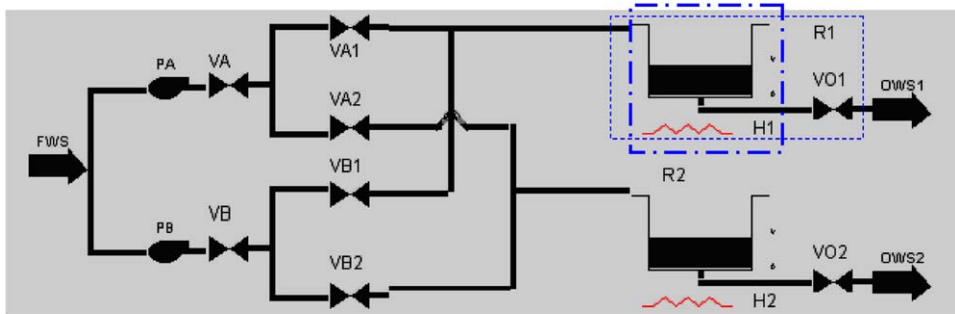
Each widget represents a state variable. It is noted that the structure (component, sub-system and system) is composed of one or more active state variables and passive state variables. Fig. 5 shows a general display of the state/structure. Fig. 5a shows that the structure has two active state variables and two passive state



● Active state variable    ○ Passive state variable

S1, S2: The identity of structure

Fig. 5. Display of the state/structure.



Component structures: PA, VA, VA1, VA2, PB, VB, VB1, VB2, VO1, VO2;

Subsystem structures: R1+H1, R2+H2;

□ : subsystem (HRT1, R1, and VO1);

□ : subsystem (HRT1 and R1).

Fig. 6. Display of the DURESS state/structure.

variables. Fig. 5b shows how two structures are connected, in which structure S2 by itself has two active state variables. It is noted that here the structure can be component, sub-system or system.

Applying this display method to the DURESS plant system, the display of the state/structure is obtained (see Fig. 6). In Fig. 6, widgets are designed as far as possible to imply the semantics of their physical entities. It is noted that in Fig. 6, the reservoir and the heater are combined into one sub-system; the widget that represents the reservoir and heater sub-system has one active state variable, and three passive state variables (i.e., one corresponding to the sum of VB1 and VA1, one to the flow rate of water out of the reservoir and one to the temperature of water out of the reservoir). It is also possible to consider R1, H1 and VO1 as one sub-system.

### 9.2. Display of the behavior

The (actual) behavior is displayed by simultaneously highlighting a set of related active states and passive states. Take the behavior of valve VA as an example (see Fig. 7). The active state of VA is the valve opening, and the passive state of VA is the flow rate; sliding the widget of the active state (i.e., the valve opening) will immediately cause a display of the passive state (i.e., the flow rate). It is noted that here all the state variables are sensor measured.

### 9.3. Display of the function

The function follows the behavior. The function can be displayed in various ways. The most compact way is to have a widget on the passive state variable to imply a target (see Fig. 8). The function here means: *to adjust the corresponding passive state to the target value.*

### 9.4. Display of the principle

As discussed before, the principle implies the constraint equation. The constraint equation is not supposed to display to the operator, albeit it is certainly an element related to the plant design model. However, the extensional component of the constraint equation (the explicit form, or one particular configuration) is displayed. That is to say, for a valve (as an example), the following information is displayed

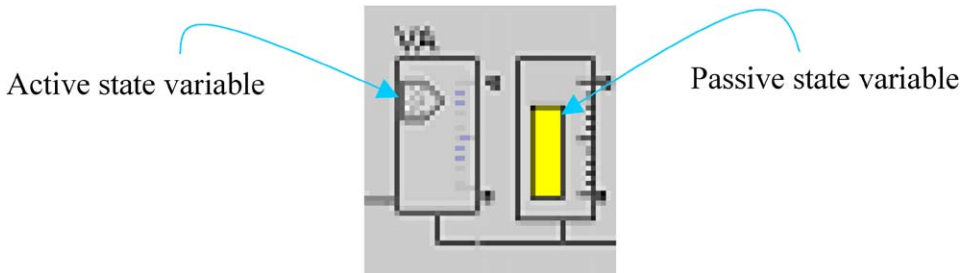


Fig. 7. Display of the behavior.

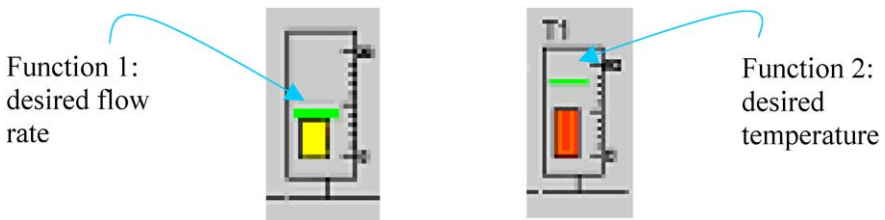


Fig. 8. Display of the function.

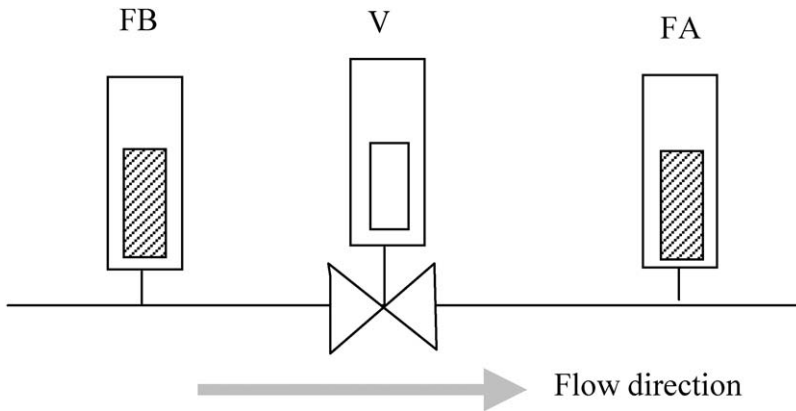


Fig. 9. Display of the principle (V: valve; FB: flow rate before the valve; FA: flow rate after the valve).

(see Fig. 9): (1) the sensor measured FB, (2) the sensor measured valve opening and (3) the computed FA based on the constraint equation for the valve. It is important to note that here FA is a passive state variable, FB is an active state variable with respect to the valve (but it is certainly not a control) and  $V$  is an active state variable (it is a control). It can be concluded that for a component/sub-system/system, the display of the principle is by means of the simultaneous display of the actual (or sensor-measured) active state variable and the computed passive state variable. The display of the computed passive state variable should be in proximity to the actual (or sensor-measured) passive state variable. In the next section, there is a discussion of how the display of the principle information assists in fault diagnosis.

### 9.5. Display of the system decomposition

The display of system decomposition is made in the following way. On the interface where the active state variable network is displayed (see Fig. 6), a kind of display control widget is provided for initiating the display of system decomposition. More generally, a menu called display control is designed on the display of the active state variables network (see Fig. 12 later). One of the items in this menu corresponds to the display of system decomposition. The system decomposition is a hierarchical structure following the whole-part relationship.

Fig. 10 shows a system decomposition of the DURESS plant. This decomposition is the same as one used in designing and constructing a plant. Note that this demonstrates a benefit of the FBS approach, i.e., a seamless integration of design of plants and design of their interfaces. When any node in the system decomposition diagram is triggered, the boundary of a corresponding sub-system is highlighted with a uniform form (e.g., one same color). For example, if a sub-system called ‘Water Temperature Adjusting’ (in Fig. 10) is triggered, a way of highlighting all the components of this sub-system is applied, specifically, a box enclosing the

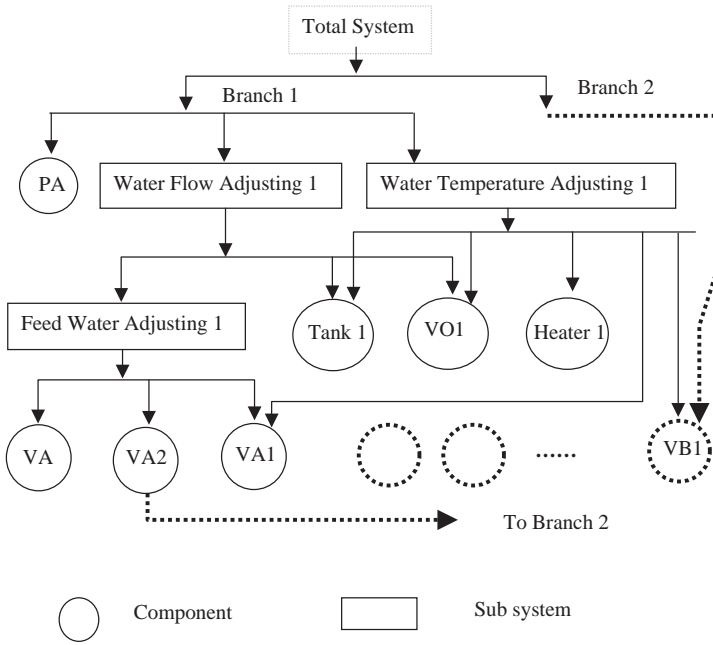


Fig. 10. Partial system decomposition of the DURESS plant.

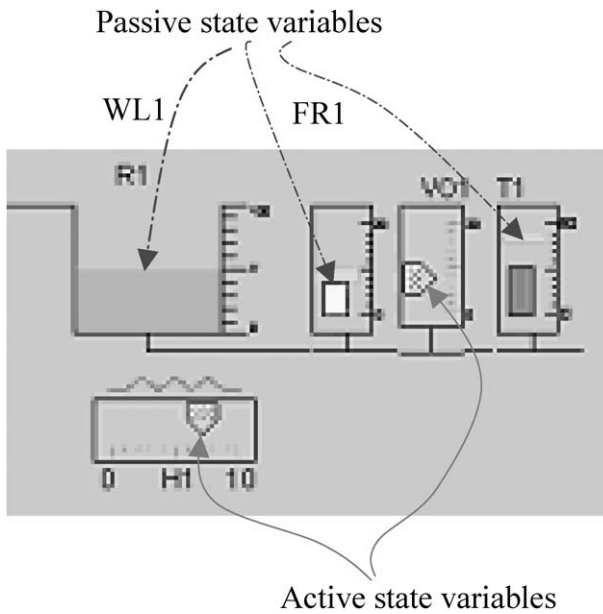


Fig. 11. Design of system decomposition. (the example sub-system consists of the components: R1, H1 and VO1).

components (R1, H1, VO1) should be displayed on the active state variable network diagram (see Fig. 6); correspondingly, on the display of behavior, the corresponding widgets for the active state variables (H1, VO1) and the passive state variables (T1, FR1, WL1) are highlighted (see Fig. 11), where FR1 is the flow rate at the exit of valve VO1, and WL1 is the water level in the tank R1. The display approach as described should follow the Proximity compatibility principle (PCP) proposed by Wickens and Carswell (1995), which is, however, not the main concern of this article. In the next section, there is a discussion of how the display of the principle information together with the display of system decomposition assists in fault diagnosis.

## 10. Example: the DURESS plant interface design

The procedure of generating an interface for the DURESS plant is demonstrated. The *first* step is simply to analyse the design model of a plant, which should be available from the plant design/manufacturing phase. The assembly model, as shown in Fig. 10, is usually a part of the design model; the other part of model is the plant connectivity model (Zhang, 1994b). While there may be different definitions of assemblies (Fig. 10), nevertheless the definitions must come from the design.

The *second* step in generating the interface is to collect a set of active state variables (in particular controls) and a set of passive state variables. Obviously, these sets also come from the design. The following is a list of active state variables and passive state variables.

Active state variables (controls) are as follows.

SA01: the switch for power of pump PA (on/off).

SA02: the switch for power of pump PB (on/off).

SA03: the opening of valve VA.

SA04: the opening of valve VB.

SA05: the opening of valve VA1.

SA06: the opening of valve VB1.

SA07: the opening of valve VA2.

SA08: the opening of valve VB2.

SA09: the power adjusting of the heater H1.

SA10: the power adjusting of the heater H2.

SA11 : the opening of valve VO1.

SA12: the opening of valve VO2.

Passive state variables (a partial list) are as follows.

SA13: the flow rate at the outlet of valve VA.

SA14: the flow rate at the outlet of valve VB.

SA15: the flow rate at the outlet of valve VA1.

SA16: the flow rate at the outlet of valve VA2.

SA17: the flow rate at the outlet of valve VB1.

SA18: the flow rate at the outlet of valve VB2.

- SA17: the flow rate at the outlet of valve VO1.
- SA18: the flow rate at the outlet of valve VO2.
- SA19: the temperature of water at the outlet of VO1.
- SA20: the temperature of water at the outlet of VO2.
- SA21: the level of the water in the tank R1.
- SA22: the level of the water in the tank R2.
- SA23: the temperature of water at the inlet of the plant system.

The *third* step in generating the interface is to assign widgets to state variables, and to design the forms and the spatial structure of widgets. Although the control of displays along the time dimension is the responsibility of human operators, the design of interfaces should make this available to the human operator for the plant control operation (see the control area widget in Fig. 12).

The *last* step in generating the interface is the layout of widgets on the display screen, a task that is primarily computer programming work. The discussion of the details of this step is omitted. Fig. 12 is the result of prototyping of the interface for DURESS plant. It is worth mentioning that the central area of widgets is for temporal control of (1) display of information, (2) system decomposition and (3) principles.

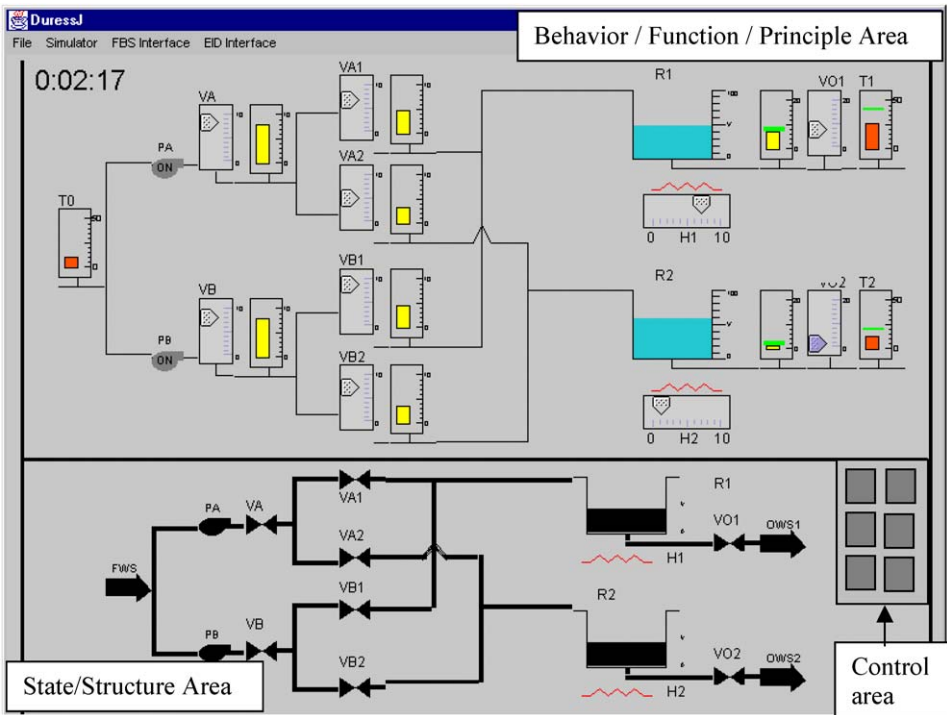


Fig. 12. DURESS interfaces based on the FBS framework.

## 11. Information contents versus operation tasks

The usefulness of the core information contents described above depends on operational tasks of a human–machine system. For the process plants application, there are three basic categories of tasks. The first category of task is called the *normal control operation*, where operators are expected to manipulate a plant such that it can reach its target (or make a plant fulfill its function). The second class of tasks is called the *fault detection*, where operators are expected to detect whether a plant is running normally, or whether there is any error with the plant operation. The third class of task is called the *fault diagnosis/compensation*, where causes for an identified fault are tracked down. It can be elaborated that for the normal control operation and fault detection classes of tasks, operators do not necessarily need access to the principles information; however, principles information will be needed for the fault diagnosis/compensation class of task.

In the normal control operation task, the operational goal is to make respective passive states reach their corresponding target states through the manipulation of the active state variables. The effects of manipulating any active state variable can be immediately seen on the target state variable when all sensors work normally. If there is any fault, the symptom will be, first of all, appearing at those target state variables. The symptom down to individual components/sub-systems can usually be captured by the sensors and then indicated by the passive state variable display.

In the case of fault diagnosis, the procedure needs to compare the active behavior with the expected behavior for components/sub-systems. Suppose that the cause of a fault is the damage of a valve. The comparison of the actual flow rate (measured by a sensor) out of the valve given a particular setting of the valve opening (an active state variable) and the expected flow, computed with the constraint equation, should determine the cause. In the following a more complicated fault is demonstrated.

Suppose that extra water dropped into the tank (R1) from an unknown source. The symptom is evident; i.e., the water level in R1 rises (assume that the system targets have already been reached). The fault diagnosis starts after the fault is detected based on the symptom. The normative procedure for finding causes is as follows.

*Step 1.* Ask for display of the tank (R1). This will lead to the display of all state variables (sensor measured) related to the tank (see Fig. 13a).

*Step 2.* Ask for display of the principle surrounding to the tank (R1). This will lead to the display of the computed values of all the passive state variables (see Fig. 13b). From Fig. 13b, it is clear that in all the passive state variables except for the water level passive state variable, the actual behaviors agree with the computed (expected) behaviors. Additionally, this step of the display makes sense of the display of principles information in the FBS framework.

*Step 3.* Ask for the display of all the controls linking to the tank (R1); see Fig. 13c.

*Inference:* Because the actual water level does not agree with the expected water level while all other behaviors are ‘correct’, the only cause for the rising of the water level must be the extra water from an unknown source.

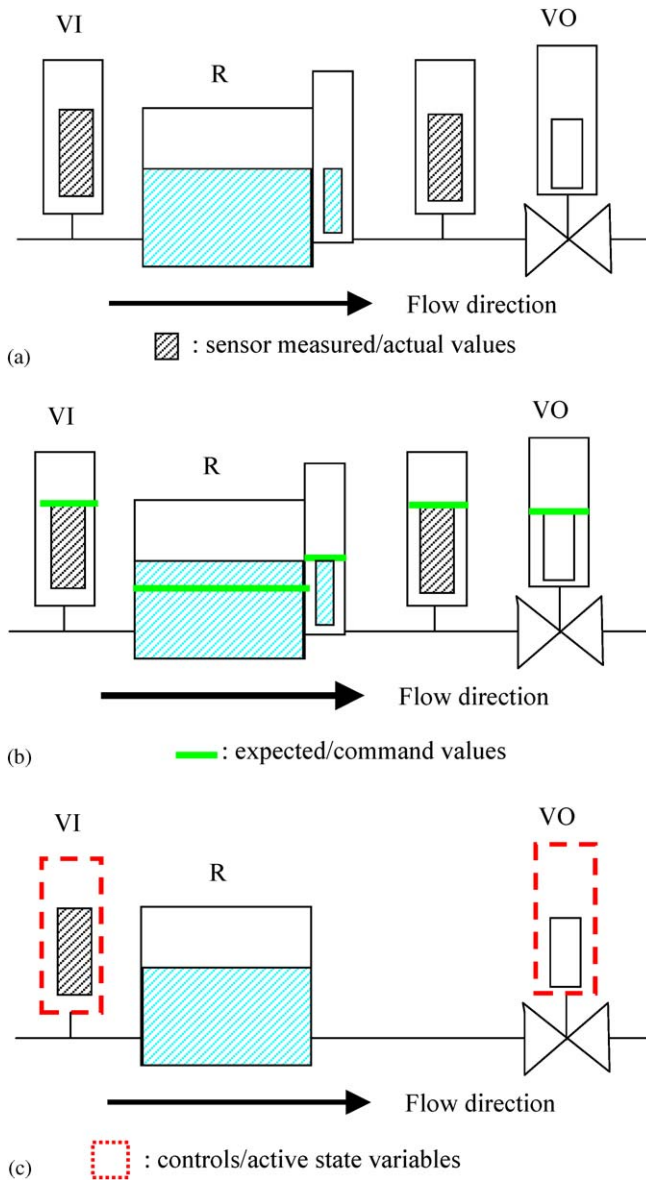


Fig. 13. Fault diagnosis for extra water in the tank (V: valve; VI: input valve; VO: output valve; R: reservoir). (a) Display of the behavior; (b) display of the principle; (c) display of the control.

Suppose that the cause for the rising of the water level is the accidental shutting down of the output valve VO1. Following the procedure described above, one should infer the cause of the fault from Step 3 that VO1 is closed. Thus the cause can be tracked down.

Suppose that the cause for the rising of the water is a clog in the outlet of the tank (R1). Following the procedure described above, one should infer the cause of the fault from Step 2 that the flow rate (sensor measured) disagrees with the computed (or expected) flow rate at the outlet of the tank (R1).

## 12. Comparison of two interface frameworks

It is worthwhile to explore the differences and similarities between the ecological interface and the function–behavior–state interface in order to finally develop an industrial standard for the process plant application.

It is clear that the two approaches stem from different perspectives of looking at the interface design problem. The ecological interface is founded from (1) the knowledge-based problem-solving of complex domain problems and (2) the representation of domain knowledge. The FBS interface sets its base on (1) the cognitive compatibility principle, (2) the representation of domain knowledge, (3) the analytical redundancy fault diagnosis technique and (4) the adaptive property of information contents and their structures with respect to tasks. Regarding knowledge representation, [Table 1](#) shows a comparison.

Note the following remarks regarding [Table 1](#).

**Remark 1.** In the ecological interface, the generalized function concept is only applied to the component level. While not explicitly stated, in practice the generalized function captures the single input and single output ‘flows’ (in the context of process control) (see also [Janzen and Vicente \(1998\)](#) about their use of the term ‘flow’). In the function–behavior–state framework, however, the behavior can be applied to the component, the sub-system, and the system. This is a very useful point in fault diagnosis because it allows for a top-down strategy to track down the fault, a process similar to the general cognitive process methodology used by a medical doctor in tracking down the cause of disease in the patient. This concept comes to the FBS framework as a result of applying the cognitive compatibility principle ([Wickens and Carswell, 1995](#)); in particular, the strategy taken by the FBS framework here is to choose concepts which can most ‘match’ the ‘average’ person.

Table 1  
The ecological interface design versus the FBS interface

The ecological interface design	The FBS framework
Physical form	Structure
Physical function	State
Generalized function	Behavior
Abstract function	Principle/constraint equations
Functional purpose	Function

**Remark 2.** The notion that behavior follows the system decomposition in the FBS framework is equally applied to the function; whereas the functional purpose level in the ecological interface design framework, though not explicitly stated, is only applied to the system level.

**Remark 3.** There is some confusion regarding the system decomposition notion in the ecological interface design framework. For example, Janzen and Vicente (1998) put the reservoir (or tank) in the DURESS plant as a sub-system, while others (e.g., Ham and Yoon, 2001a) tended to consider the tank to be a component. Plausible causes for this confusion may result from the fact that the abstract function was actually applied to both the tank and the output valve in the study reported by Janzen and Vicente (1998). Certainly, it is cumbersome to put together both the tank and the output valve as a component. This point is closely related to the next remark.

**Remark 4.** In the ecological interface design literature, the abstract function is applied only to those components that have a ‘storage’ function (e.g., the tank) for the process plant application, though this is not explicitly stated in the ecological interface design theory (Rasmussen, 1983; Rasmussen and Vicente, 1989; Vicente and Rasmussen, 1992). Note that there is, however, only one mention of the storage component relating to the abstract function (i.e., Vicente, 1999). This restricted use of the abstract function looks ‘odd’; yet a careful examination of what actually happens with the abstract function on the display may resolve this ‘odd’ appearance. The use of the abstract function (the mass balance for example) is such that the change rate of the mass in the tank is zero, corresponding to the stability of the plant (based on the first-order dynamic model).

**Remark 5.** When it comes to the display of these concepts (Table 1), there are large differences presented between the two interfaces. *First*, the ecological interface design framework displays all four levels (usually excluding the physical form level) at once, and these are ‘permanently’ on the interface display; the FBS framework, however, does not display the principle/constraint equation for the normal control operation and fault detection tasks. *Second*, the FBS strategy for display of the principle information is such that the principle/constraint equation tells the deepest knowledge about domain, so this is the interface best used for explaining the causes for faults functionally. This strategy is very similar to what is currently happening in medicine; there has been a growing interest in finding disease markers at the protein level, which is at a much deeper level of knowledge governing the biological process than that of previous research.

Moving to a morphological aspect of the ecological interface design and FBS frameworks, Rasmussen and Vicente (1989) explained that in the ecological interface design framework the higher level of concept is of both the emerging and the governing properties with respect to its immediate lower level of concept. When this point was put into use for interface design, they appeared to only take the emerging properties, evidenced by the fact they put all the five levels along the same dimension

(which may be called the emerging dimension). In the function–behavior–state framework, the five levels of concept are not on the same dimension, and there are two dimensions in the FBS framework (i.e., the governing dimension and the emerging dimension, which are orthogonal to each other). In particular, the structure, state, behavior and function are on the emerging dimension, while the principle is on the governing dimension. In fact, there is one more notion called the *effect* (in the FBS framework) that is related to the behavior and the function (see also the pervious discussion of the FBS framework on the function definition). The notion of effect tries to answer the question regarding why particular behaviors are useful, meaning functional (i.e., having particular functions). As such, the effect may be defined as a technological and sociological phenomenon observed by humans and used for certain purposes, depending on how far the human’s imagination and innovation can reach. This notion has yet to be explored in the interface design. Overall, Fig. 14 shows the morphological structure of the FBS framework. In this figure, the dashed line with double arrows means that the state of the actual state variable (as a kind of input) is fed into the Principle, while the state of the passive state variable (as a kind of output) results from the evaluation of the Principle.

Regarding the information display, the ecological interface design framework displayed all the five levels of information on the emerging dimension at one time and at one place regardless of what the operation tasks and task environments were. In the FBS framework, the levels of information were displayed depending on the operation tasks. In particular, for the start-up operation, only the structure and the state levels of information may be displayed. For the normal control or fault detection task, the state, the behavior and the function level of information may be displayed. For the fault diagnosis task, the principle information is displayed in such

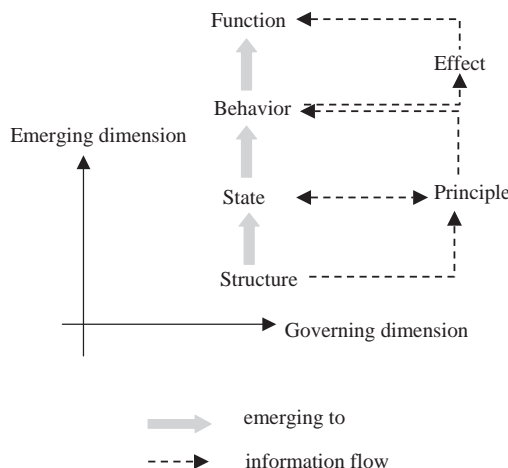


Fig. 14. Morphological structure of FBS framework.

a manner that the analytical redundancy technique is incorporated into the displaying process.

On a general note, the ecological interface has not provided a normative procedure for fault diagnosis because its abstract function is only displayed for its emerging property, albeit in Vicente (1999) mention was given regarding the general strategy or approach for fault diagnosis. Both the ecological interface and FBS interface are ecological in certain ways, since both emphasize the representation of domain structural knowledge. Care must be taken with regard to the scope of the environment that underlies both frameworks. While in the ecological interface design framework the environment is meant for the domain knowledge, the FBS framework considers the environment as comprised of (plant) domain knowledge, task and task situation. The task situation is a broad notion that is meant for anything other than the domain knowledge and task. Consequently, the FBS framework is also task-sensitive.

### 13. Summary and discussion

The FBS framework was primarily aimed at process (discrete event and/or continuous event) control applications. The novelty of this framework is such that it sets out as its basis on design/manufacturing of a machine or plant system; in particular, it argues that the model of plant systems used by design (including manufacturing and maintenance) engineers should also be the underlying model, as a framework, to derive the information needed for human operators. This is, therefore, a junction point to separate the FBS framework from the ecological interface design framework. The ecological interface design framework is rooted in a knowledge-based problem-solving paradigm, based on which a five-layer general problem-solving architecture was proposed. It is argued that the problem-solving paradigm is not optimal to the process control operations, as the problem-solving model can create an intermediate step that is unnecessary (see Fig. 15a), whereas the FBS framework removes this unnecessary step (see Fig. 15b).

The FBS framework is rooted in the FBS model for design of general products. The FBS model was first developed by researchers in the artificial intelligence field, and then was used by design researchers. The FBS model has the following concepts: structure, state, behavior, function, principle (between the state and the behavior) and effect (between the behavior and the function). The concept of effect was not explored in the FBS framework. The FBS framework can derive the information needed and the display of the information on the time dimension with respect to operational tasks.

One of the important advantages of the FBS framework is that it provides a straightforward opportunity to incorporate interface design into the product life cycle design. A comparative study on the FBS framework and the ecological interface design framework based on the experiment has been presented in Lin et al. (2003).

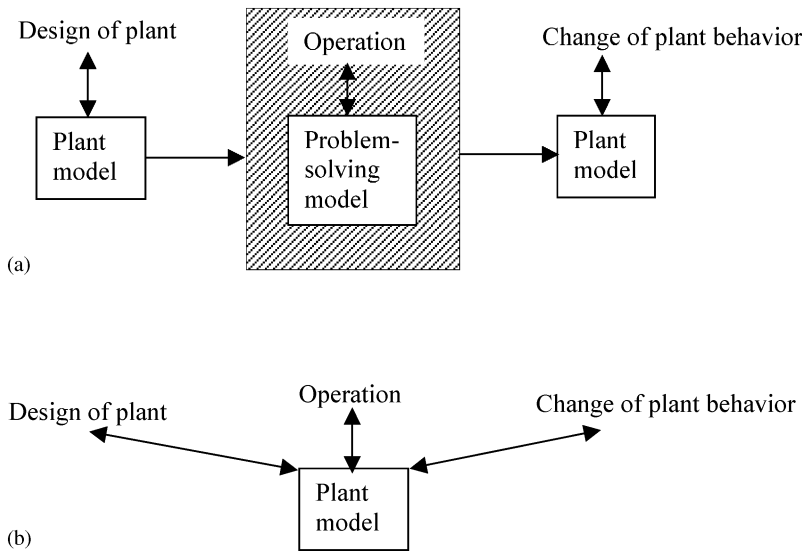


Fig. 15. Incorporation of interface design into product life-cycle design. (a) The problem solving strategy for interface design; (b) the design model showing strategy for interface design.

### Acknowledgements

The generous financial support from the Natural Sciences and Engineering Research Council (NSERC) and the Atomic Energy of Canada Limited (AECL) is acknowledged. The authors would like to thank Kim J. Vicente for his kind supply of the DURESS software system. We would also want to take this opportunity to express our appreciation to the anonymous reviewers of this article in bringing their constructive suggestions. And finally, a very special thank you is due to Brian Gaines, *IJHCS* editor-in-chief, for his efficient editorial management.

### Appendix A. Kinematics of a closed loop linkage: loop equation, branch, configuration

Fig. 16 is a five-bar linkage, where  $Q_{1A}$  and  $Q_{2A}$  are two active state variables, or generalized coordinates in terms of Lagrange's mechanics, while  $Q_{3A}$  and  $Q_{4A}$  are passive state variables. The kinematic analysis is to develop the governing equation, through which, given  $Q_{1A}$  and  $Q_{2A}$ , one can find  $Q_{3A}$  and  $Q_{4A}$ . The principle for developing such a governing equation is to conceptualize each link as a vector (see Fig. 17), and then according to the vector algebra, we have

$$\vec{l}_1 + \vec{l}_3 = \vec{l}_5 + \vec{l}_2 + \vec{l}_4, \tag{A.1}$$

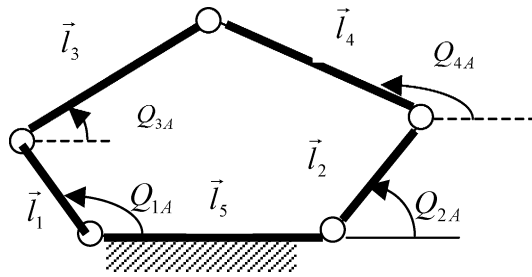


Fig. 16. A five-bar linkage.

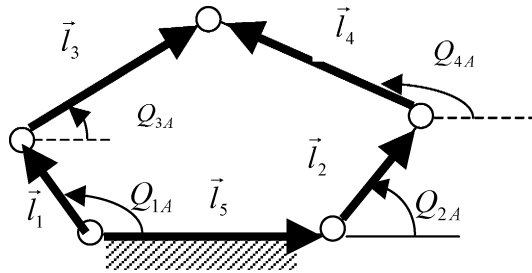


Fig. 17. Vector model of the five-bar linkage.

with

$$\vec{l}_1 = l_1 e^{iQ_{1A}}, \tag{A.2}$$

$$\vec{l}_2 = l_2 e^{iQ_{2A}}, \tag{A.3}$$

$$\vec{l}_3 = l_3 e^{iQ_{3A}}, \tag{A.4}$$

$$\vec{l}_4 = l_4 e^{iQ_{4A}}. \tag{A.5}$$

Eqs. (A.2)–(A.5) are complex number expressions. Substitution of Eqs. (A.2)–(A.5) into Eq. (A.1) should lead to one vector equation containing two unknown variables:  $Q_{3A}$  and  $Q_{4A}$ . Two scalar equations can be written from one vector equation, therefore we will have two scalar equation containing  $Q_{3A}$  and  $Q_{4A}$ , i.e.,

$$Q_{3A} = \tan^{-1} \left[ \frac{\pm \sqrt{A^2 + B^2 - C^2}}{C} \right] + \tan^{-1} \left[ \frac{B}{A} \right], \tag{A.6}$$

$$Q_{4A} = \tan^{-1} \left[ \frac{\mu + l_4 \sin Q_{3A}}{\lambda - l_4 \cos Q_{3A}} \right] + \tan^{-1} \left[ \frac{B}{A} \right], \tag{A.7}$$

where

$$\lambda = l_4 \cos Q_{2A} - l_1 \cos Q_{1A}, \tag{A.8}$$

$$\mu = l_4 \sin Q_{4A} - l_1 \sin Q_{1A}, \tag{A.9}$$

$$A = 2l_3\lambda, \tag{A.10}$$

$$B = 2l_3\mu, \tag{A.11}$$

$$C = l_2^2 - l_3^2 - \lambda^2 - \mu^2. \tag{A.12}$$

The detailed derivation of Eqs. (A.6) and (A.7) can be found in [Ouyang et al. \(2003\)](#).

There will be two sets of solution for  $Q_{3A}$  and  $Q_{4A}$ , corresponding to the sign “+” and “-”, respectively, given a set of  $Q_{1A}$  and  $Q_{2A}$ ; geometrically these two sets correspond to one solid schematic diagram and one dashed line schematic diagram (in [Fig. 18](#)).

These different sets of solutions are called different branches in the mechanisms literature. There is a simple method to identify these two branches graphically ([Zhang, 1990](#)). We pick up three points  $A, B, Q$  (or  $Q'$ ). The order of the triangle formed by  $A, B, Q$  (or  $Q'$ ) will be either clockwise or counter-clockwise, which identifies two different branches. In [Fig. 18](#), we obtain: the triangle  $AB\tilde{Q}$ : counter-clockwise; the triangle  $AB\hat{Q}$ : clockwise. Therefore, they belong to different branches; specifically we name  $AB\hat{Q}$  as branch  $A$  and  $AB\tilde{Q}$  as branch  $B$ .

From a viewpoint of kinematic governings Eq. (A.1) is called general governing equation, while Eqs. (A.6) and (A.7) are called specific governing equations. If we use the specific governing equations, e.g., the specific governing equations corresponding to the solid line in [Fig. 18](#), when we change  $Q_{1A}$  and  $Q_{2A}$  sequentially, we will get a series of results of  $Q_{3A}$  and  $Q_{4A}$  which all correspond to the solid branch.

Configuration differs from the branch in that a specific configuration is determined by a set of active state variables ( $Q_{1A}$  and  $Q_{2A}$ ). Therefore, [Fig. 18](#) shows one configuration of linkage; yet there are two branches corresponding to that configuration. It is possible that more than one configuration could achieve a target. [Fig. 19](#) shows two configurations achieving the target  $\tilde{Q}$ . These two configurations are of different branches. The configuration drawn by the solid line is of branch  $A$ , while the configuration drawn by the dashed line is of branch  $B$ .

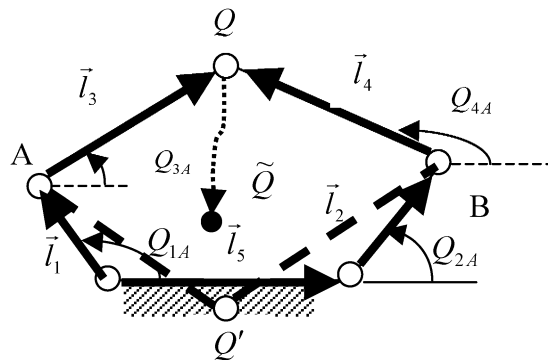


Fig. 18. Branches and configurations (solid line: branch  $A$ ; dashed line: branch  $B$ ).

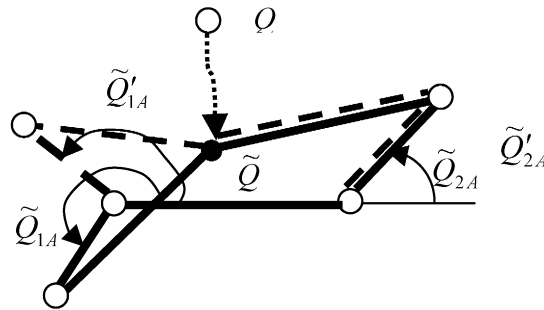


Fig. 19. Reaching a target with two configurations of different branches (solid line: configuration 1 of branch  $A$ ; dashed line: configuration 2 of branch  $B$ ).

## References

- Bracewell, R., Sharp, J., 1994. A computer aided methodology for the development of conceptual schemes for mixed energy-transforming and real-time information systems. In: Sharp, J., Ho, V. (Eds.), *Computer Aided Conceptual Design*. LooseLeaf Co., UK, pp. 79–94.
- Burns, C.M., 2000a. Navigational strategies with ecological displays. *International Journal of Human-Computer Studies* 52, 111–129.
- Burns, C.M., 2000b. Putting it all together: improving display integration in ecological displays. *Human Factors* 42 (2), 226–241.
- Burns, C.M., Thompson, L.K., Rodriguez, A., 2002. Mental workload and the display of abstraction hierarchy information. *Proceedings of the 46th Annual Meeting of the Human Factors and Ergonomics Society*, Baltimore, MD, September 28–October 4.
- Christoffersen, K., Hunter, C.N., Vicente, K.J., 1996. A longitudinal study of the effects of ecological interface design on skill acquisition. *Human Factors* 38 (3), 523–541.
- CNSC G-276 (Regulatory Guide), 2003. *Human factors engineering program plans*. Canadian Nuclear Safety Commission, Canada.
- Frank, M.P., 1990. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy—a survey and some new results. *Automatica* 26, 459–474.
- Gibson, J.J., 1979. *The Ecological Approach to Visual Perception*. Houghton Mifflin Company, Boston, ISBN: 0-395-27049-9.
- Ham, D.H., Yoon, W.C., 2001a. Design of information content and layout for process control based on goal-means domain analysis. *Cognition, Technology and Work* 3, 205–223.
- Ham, D.H., Yoon, W.C., 2001b. The effects of presenting functionally abstract information in fault diagnosis tasks. *Reliability Engineering and System Safety* 73 (2), 103–119.
- IEEE Std 1289, 1998. *IEEE guide for the application of human factors engineering in the design of computer-based monitoring and control displays for nuclear power generating stations*. The Institute of Electrical and Electronics Engineers, Inc., New York, NY 10017-2394, ISBN 0-7381-0182-6.
- Janzen, M.E., Vicente, K.J., 1998. Attention allocation within the abstract function. *International Journal of Human-Computer Studies* 48, 521–545.
- Jamieson, G.A., Vicente, K.J., 2001. Ecological interface design for petrochemical applications: supporting operator adaptation, continuous learning, and distributed, collaborative work. *Computers and Chemical Engineering* 25, 1055–1074.
- Johnson, S., 2003. Inside TMI: minute by minute. (website: <http://kd4dcy.net/tmi/>).
- Lin, Y., 2000. Experimental study based on eyegaze measurement for computer interface. AEDL Technical Report, Advanced Engineering Design Laboratory, Department of Mechanical Engineering, University of Saskatchewan, Canada, 105pp.

- Lin, Y., Zhang, W.J., 2003. Combining eye movement and hand movement measures for evaluating human-machine interfaces. 2003 IEEE International Conference on Systems, Man & Cybernetics, Washington, DC, USA, October 5–8, pp. 1089–1096.
- Lin, Y., Zhang, W.J., Watson, L.G., 2003. Using eye movement parameters for evaluating human-machine interface frameworks under normal control operation and fault detection situation. *International Journal of Human-Computer Studies* 59, 837–873.
- NUREG-0711, 1994. Human factors engineering program review model. US Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, Washington, DC, 20555-0001.
- Ouyang, P., Li, Q., Zhang, W.J., 2003. An integrated approach for force balancing and trajectory tracking for controllable mechanism. *Journal of Mechatronics* 13, 887–905.
- Pahl, G., Beitz, W., 1995. *Engineering Design: A Systematic Approach*, 2nd Edition. Springer, London.
- Penington, J., 1995. Human factors guides—report prepared for the atomic energy control boards. AECB Project No. 2.280.2.
- Rasmussen, J., 1983. Skills, rules, and knowledge: signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man and Cybernetics* SMC-13, 257–266.
- Rasmussen, J., Vicente, K.J., 1989. Coping with human errors through system design: implications for ecological interface design. *International Journal of Man-Machine Studies* 31, 517–534.
- Seeing Machine, 2003. [www.seeingmachiens.com](http://www.seeingmachiens.com).
- Shigley, J.E., Uicker, J.J. Jr., 1995. *Theory of Machines and Mechanisms*. McGraw-Hill, New York.
- Suh, N.P., 1990. *The Principles of Design*. Oxford University Press, Oxford.
- Umenda, Y., Takeda, H., Tomiyama, T., Yoshikawa, H., 1990. Function, behavior and structure. In: Gero, J.S. (Ed.), *Applications of Artificial Intelligence in Engineering the Fifth International Conference*, Vol. 1. Springer, Berlin, pp. 177–194.
- Vicente, K.J., 1991. Supporting knowledge-based behavior through ecological interface design. Doctoral Dissertation, University of Illinois, Urbana-Champaign, Unpublished.
- Vicente, K.J., 1999. *Cognitive Work Analysis, Toward Safe, Productive, and Healthy Computer-based Work*. Lawrence Erlbaum Associates, Hove.
- Vicente, K.J., Rasmussen, J., 1992. Ecological interface design: theoretical foundations. *IEEE Transactions on Systems, Man, and Cybernetics* 22 (4), 589–606.
- Vicente, K.J., Christoffersen, K., Perekhita, A., 1995. Supporting operator problem-solving through ecological interface design. *IEEE Transactions on Systems, Man, and Cybernetics* 25 (4), 529–545.
- Wickens, C.D., 1992. *Engineering Psychology and Human Performance*. Harper Collins, New York, ISBN 0673461610.
- Wickens, C.D., Carswell, C.M., 1995. The proximity compatibility principle: its psychological foundation and relevance to display design. *Human Factors* 37 (3), 473–494.
- Yamaguchi, Y., Tanable, F., 2000. Creation and evaluation of an ecological interface system for operation of nuclear reactor system. *Proceedings of the XIVth Triennial Congress of the International Ergonomics Association and 44th Annual Meeting of the Human Factors and Ergonomics Society* Vol. 3, San Diego, CA, pp. 571–574.
- Zhang, H., 1994a. Stimulus-stimulus and stimulus-response compatibility: an interactive activation model. Doctoral Dissertation, University of Michigan, Ann Arbor, MI, Unpublished.
- Zhang, W.J., 1990. Determination of the assembling position of planar linkages. *Journal of China Textile University (English edition)* 7 (2), 68–72.
- Zhang, W.J., 1994b. An integrated environment for CAD/CAM of mechanical system. Ph.D. Thesis, Delft University of Technology, The Netherlands, 63pp, ISBN 90-370-0113-0.
- Zhang, W.J., 1998. A knowledge-based life cycle engineering approach to nuclear power plants. The proposal submitted to NSERC and AECL for a CRD Grant (successful), 30pp.