Human Factors, vol.47, no.1, pp.12, 2005

Designing Effective Human-Automation-Plant Interfaces: A Control-Theoretic Perspective

Greg A. Jamieson and **Kim J. Vicente,** University of Toronto, Toronto, Canada

In this article, we propose the application of a control-theoretic framework to human-automation interaction. The framework consists of a set of conceptual distinctions that should be respected in automation research and design. We demonstrate how existing automation interface designs in some nuclear plants fail to recognize these distinctions. We further show the value of the approach by applying it to modes of automation. The design guidelines that have been proposed in the automation literature are evaluated from the perspective of the framework. This comparison shows that the framework reveals insights that are frequently overlooked in this literature. A new set of design guidelines is introduced that builds upon the contributions of previous research and draws complementary insights from the control-theoretic framework. The result is a coherent and systematic approach to the design of human-automation-plant interfaces that will yield more concrete design criteria and a broader set of design tools. Applications of this research include improving the effectiveness of human-automation interaction design and the relevance of human-automation interaction research.

INTRODUCTION

The use of automation in complex sociotechnical systems has proved to be a double-edged sword. It is a technology that, perhaps more so than any other, speaks with a forked tongue to system designers. On the one hand, it promises unprecedented reliability, reduced workload, improved economy, and fewer errors. On the other hand, it whispers of less tangible, but no less real, costs to operators in terms of skill degradation, mental isolation, and monitoring burdens (Hirschhorn, 1984; Norman, 1990; Parasuraman, Molloy, Mouloua, & Hilburn, 1996).

The human factors response to automation has been hesitant, perhaps on account of the discomforting discrepancy between the advantages and disadvantages just noted. Automation technology seems to possess a momentum that cannot be deterred, not to mention controlled. The warnings of human factors professionals examining automation effects seem to go unheeded by control engineers swept up in the wave of microprocessor and software innovations that sometimes overwhelm concerns for human limitations. Control systems automation is the epitome of a technology-driven enterprise. To be sure, it has been very effective in achieving some of its goals. For this reason, it is difficult to question its application. However, a number of authors have pointed out that automation introduces problems of its own (e.g., Wiener & Curry, 1980). How can engineers be convinced that automation technology is a tool and not a panacea for human-machine interaction problems (Billings, 1997)? The ironic solution that we adopt in this article is to use automation designers' own language – control theory – to point out the role of human factors considerations in the design of automation.

Reviews of the evolution of automation in complex systems and of the empirical research on human interaction with automation are already available in the literature (Billings, 1997; Moray, 1986; Mouloua & Parasuraman, 1994; Parasuraman & Mouloua, 1996; Parasuraman & Riley, 1997; Sheridan, 1987, 2002; Wickens, Mavor, Parasuraman, & McGee, 1998; Wiener,

Address correspondence to Greg A. Jamieson, Department of Mechanical and Industrial Engineering, University of Toronto, 5 King's College Rd., Toronto, ON, Canada M5S 3G8; jamieson@mie.utoronto.ca. **HUMAN FACTORS,** Vol. 47, No. 1, Spring 2005, pp. 12–34.

1985). The novel contribution of this article is a domain-independent, control-theoretic framework to investigate human factors issues in automation. We demonstrate that the framework provides critical insight into evaluating the design of existing automation interfaces in nuclear power plants. We also show the value of this generic framework by using it to critically review the literature dealing with mode proliferation, mode transitions, mode awareness, and mode error. Finally, the control-theoretic framework is used to propose a set of design principles that augment existing guidelines to more effectively contribute to the design process.

A CONTROL-THEORETIC FRAMEWORK FOR STUDYING AUTOMATION

Definitions

The following definitions will be employed in this article. In a departure from the journal's usual editorial style, these terms, and a few others, are capitalized throughout the manuscript to ensure that their use is not confused with other uses of the same or similar terms with which the reader may be familiar.

System: Plant + Controllers + Instrumentation + Interface.
Plant: Final Control Components + Process.
Final Control Components: controllable equipment that is used to influence a Process.
Process: the entity being controlled.
Controller: the automated means by which action is exerted on the Final Control Components.
Instrumentation: the means by which data about the Plant and Controllers are gathered.
Interface: Displays + Controls.
Displays: the devices through which Operators obtain information about the System.
Controls: the devices through which Operators take Action on the System.
Human Operator: the human actor who interacts with the System to achieve goals.

Note that the Human Operator has not been included within our definition of System, not because the Operator is outside our scope of analysis but because there is value in distinguishing the Human Operator from the System for the purposes of this article. Although the boundaries defined here can be difficult to establish in practice, they are conceptually useful because they facilitate an understanding of the nature of different types of Human-System interactions. The relationships among the defined terms, and the problems to which they are applicable, should become clearer as we place them in the context of a negative feedback loop.

Negative Feedback Loop

Figure 1 presents a simple but conceptually powerful model of an automated System based on a standard negative feedback control loop (adapted from Wade, 1994). In this figure, boxes represent elements in the feedback loop and arrows denote signals between elements. The elements in the loop were defined previously, whereas the signals are defined in Table 1. The lower portion of Figure 1 is a traditional regula-

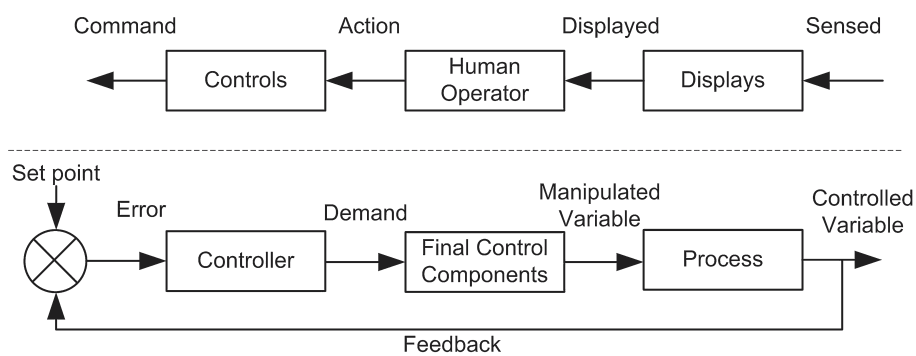

Figure 1. A negative feedback control loop.

TABLE 1: Definitions of Signals in the Feedback Control Loop

Signal	Source	Destination
Set Point	Design decision: Human Operator and/or Controls and/or Process	Comparator
Error	Comparator	Controller
Demand	Controller	Final Control Components
Manipulated Variable	Final Control Components	Process
Controlled Variable	Process	Design decision: Displays and/or Human Operator
Feedback	Process	Comparator
Sensed	Design decision: Process and/or Final Control Components and/or Controller and/or Controls	Displays
Displayed	Displays	Human Operator
Action	Human Operator	Controls
Command	Controls	Design decision: comparator and/or Final Control Components and/or Displays

tory feedback loop composed of three distinct elements (the upper portion of the loop will be described later). If all of these elements are working as planned, then the behavior of the feedback loop can be described in a relatively straightforward fashion. Tracing through the diagram from left to right, the comparator subtracts the Feedback signal from the Set Point (i.e., the goal) to create an Error signal. This Error signal is sent to an automatic Controller that generates (via programmed logic) a Demand signal that is sent to the Final Control Components. The Final Control Components (e.g., a valve) map this Demand signal onto a change in the state of the Manipulated Variable (e.g., valve position), which then influences the Process itself, causing the Controlled Variable (e.g., flow rate) to change as well. The Instrumentation (e.g., a flow meter) gathers information about the Controlled Variable and feeds it back to the comparator.

The upper portion of Figure 1 serves as a placeholder for the Interface and the Human Operator, two elements that play a key role in feedback control loops that are not fully automated. Moving from right to left, Sensed data signals are fed into the Displays element and Displayed to the Human Operator. The Operator takes Action by manipulating the Controls in the Interface, which generates a Command signal. The connections between the upper portion of the figure and the rest of the feedback control loop (i.e., which signals serve as inputs and which as outputs) cannot be described generically because there is no process that is agreed upon by automation designers for identifying which connections are required. Connecting the upper and lower portions of Figure 1 involves making a set of design decisions (see Table 1). One of the contributions of this article is to propose how these decisions might be made.

Thus far, our discussion of the negative feedback loop has been abstract because control theory is a generic language for describing the behavior of dynamic systems. In fact, this is one of its primary advantages. It provides a well-defined set of concepts that are applicable to diverse application domains. In process control, aviation, and elsewhere, both qualitative and quantitative applications of control theory have been a feature of human factors for decades (e.g., Rouse, 1980; Sheridan & Ferrell, 1974). However, the framework may be of limited value in the treatment of asynchronous control problems, such as those encountered in tactical decision making, which may not be well characterized by a standard negative feedback control loop. Also, the mathematical formalisms of

control theory are not yet sophisticated enough to comprehensively account for human cognitive processes (Bainbridge, 1981). Nevertheless, as we will show in the remainder of this article, the qualitative application of control-theoretic concepts can provide a valuable and unique set of insights that have not yet been fully drawn in the literature.

Implications

Although control engineers will find our feedback model oversimplified, it serves to make five important points. First, four different elements must be distinguished conceptually: the automation, which is the Controller, the Final Control Components that the automation acts on, the Process itself, and the Interface that (nominally) allows the Human Operator to monitor and manipulate the rest of the elements. Frequently, researchers do not discriminate clearly and consistently among these elements. A common approach is to use the catch-all label, *system.* In some cases, *system* seems to mean automation, whereas in other cases it seems to include the Final Control Components and the Process together. Furthermore, in some cases, it seems that authors inadvertently switch between these two uses of the term *system* in the same publication. We say that this *seems* to be the case because we can make only inferences based on the surrounding text; a definition of *system* is rarely given.

The second implication is that a failure can occur in different System elements. For example, it is possible for malfunctions to occur in the Controller, the Final Control Components, or the Process. These are qualitatively different events that may require different types of Operator Actions (e.g., turning off the automation vs. isolating the failed component vs. shutting down the Process). If only the catch-all term *system* is used, then it will be possible to say merely that there is a problem "somewhere." It will not be possible to localize the problem to a particular element and act accordingly.

Third, if it is important for the Operators to discriminate among the different types of failures just described, then all of the signals in the lower portion of Figure 1 should be available in the Interface (although not necessarily always or simultaneously). For example, to determine if there is a problem with the Controller, Operators should know what the current Error signal is, know what the current Demand signal is, and then use their knowledge of how the Controller is supposed to work to determine if the Demand is what it should be, given the current input to the Controller. If any one of these elements is not displayed or known, then the problem is formally underdetermined and, thus, cannot be solved, whether it be by a human or machine actor. The use of a normative model (in this case, of the Controller) to check if the System element is behaving as expected is known as *analytical redundancy,* a well-accepted technique in control theory (Frank, 1990). The same general approach can be applied to other System elements. For example, if the Demand signal being fed to the Final Control Components by the Controller and the actual value of the Manipulated Variable are known, then knowledge of how the Final Control Components are supposed to work can be used to see if these elements are responding as they should. This type of analytical redundancy comparison can also be made for the Process by comparing the current Controlled Variable value with the expected value, given the current Manipulated Variable acting as a signal input to the Process.

If any one of these signals is not available, then it will not be possible to determine precisely where the problem is. Note that this is not an empirical issue. There is no need to conduct experiments because the problem is not an incompatibility with human capabilities and limitations; rather, the problem is that human or machine actors would be faced with an underdetermined task (Degani & Heymann, 2002). For example, if the Controlled Variable does not match the goal Set Point, one cannot know if it is because (a) the Controller is sending appropriate Demand signals but the Final Control Components are not responding as they should (e.g., an actuator failure); (b) the Final Control Components are functioning properly and are merely reacting to inappropriate Demand signals from a faulty Controller (e.g., faulty control logic); or (c) both the Controller and Final Control Components are behaving properly, but there is a fault in the Process (e.g., a leaky tank). Later in this article, we show that some automation interfaces suffer from this problem.

Thus the control-theoretic framework in Figure 1 has important implications for design.

Fourth, the same Plant can be regulated by one of many different Controllers. Thus, the design of the Controller need not be taken as a given but can, instead, be seen as an element that should be specified by taking into account, among other things, human factors considerations (Leveson & Palmer, 1997; Riley, 1996, 2001; Vicente & Rasmussen, 1990). This perspective differs from the traditional view that the design of automatic Controllers is strictly a technical issue that must be resolved before human factors professionals can start their job of Interface design (see Vicente, Christoffersen, & Hunter, 1996).

Fifth, and finally, the internal structure of the Controller is different from the internal structure of the Final Control Components or of the Process. Thus when researchers recommend that Interfaces be designed so as to make the "system" transparent, it is not clear what this really means. For example, a visualization of the Controller will look very different from a visualization of the Process. After all, the former is the element doing the controlling, whereas the latter is the element being controlled.

In summary, the control-theoretic framework in Figure 1, although simple, leads to several points that have significant implications for both research and design. These points are summarized in the left column of Table 2 (the right column will be discussed later). As with any generic systems framework, some points may be more relevant to one application domain than to another.

AUTOMATION INTERFACE DESIGN: THE CASE OF NUCLEAR POWER

In this section, we put the control-theoretic framework into action by using it to account for field study findings pertaining to the design of automation interfaces in commercially operated nuclear power plants.

Background

The observations described here were conducted as part of a research program focusing on how Operators monitor nuclear power plants under normal operating conditions (Mumaw, Roth, Vicente, & Burns, 2000; Vicente, Mumaw, & Roth, 2004; Vicente, Roth, & Mumaw, 2001). The Plants that were observed are highly automated and thereby provide a relevant case study for the purposes of this article. The research was sponsored by a government regulatory body that was concerned about several minor incidents that had been reported in which Plants had slowly drifted from their normal state without anyone noticing for quite some time, in some cases not until the following 12-hr shift.

TABLE 2: Human-Automation Implications of Feedback Control Loops: Standard and Mode Induced

Implied by Figure 1	Implied by Figure 3
The Controller, the Final Control Components, the Process, and the Interface must be distinguished as elements.	The various modes within the Controller, the Final Control Components, the Process, and the Interface must be distinguished.
Failures can occur in any System element.	Failures can occur in any mode within any System element.
All signals in the feedback loop should be accessible in the Interface if Operators are to discriminate between failures in different System elements.	The number of potential signal paths has been multiplied by modes. These paths should all be accessible in the Interface if the Operator is expected to distinguish between failures in modes and elements.
The choice of Controller is a design decision with human factors implications.	The choice of mode availability within the Controller is a design decision with human factors implications.
The internal structure of each element is distinct from the others in the System.	The internal structure of each mode within each element is distinct from the others in the element and elsewhere in the System.

The sponsors were thus interested in better understanding the demands imposed by monitoring the automated Plant under normal operations and the types of strategies that Operators had adopted to satisfy these demands. By doing so, the sponsors expected to understand why lapses in monitoring had occurred and, perhaps, how they could be prevented through changes in systems design, training, or operation.

Putting the Control-Theoretic Framework to Work

The control-theoretic framework in Figure 1 was used to investigate the issues just described. As summarized in Table 2, this framework suggests that one should distinguish among four System elements because failures can occur in any element. In addition, the framework suggests that all signals in the feedback loop need to be displayed to Operators if they are going to be able to discriminate between different types of failures efficiently and reliably. How well did the observed control rooms stand up to these recommendations?

There were two types of Displays to reveal the state of the automated control loops in the plants that were observed: (a) analog hard-wired displays consisting of one meter for each control loop and (b) digital CRT-based displays consisting of one or more pages of alphanumeric data for each control loop. Although the form in which the information was presented in each of these display types differed, the content provided by each was essentially the same. Thus, for the purposes of this article, we will focus solely on the analog Displays.

Figure 2 illustrates the design of these Displays. An analog scale serves as the frame of reference for the meter. Against that backdrop, the value of the goal Set Point for the automatic Controller is shown as a wide green band, and the current value of the Controlled Variable is shown on the same scale as a thin red bar. The distance between the two thereby represents the Error signal. When the goal is satisfied, the red bar is in the green bar, as in Figure 2a. When the goal is not satisfied, the red bar falls outside of the green bar, as in Figure 2b. Below the analog meter, there is a very small scale (not shown in Figure 2) indicating the Demand signal that is sent to the Final Control Components. This scale, although providing relevant information, is too small to be legible from where the Operators normally stand or sit. Note that the state of the Manipulated Variable is not displayed at all.

Thus the design of these Displays falls short of the criteria stipulated by our framework

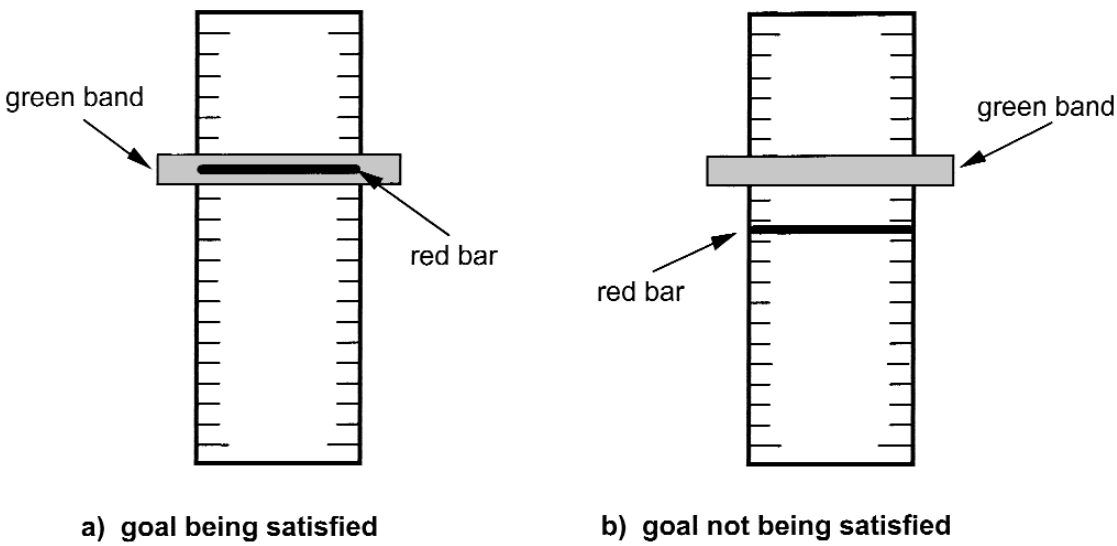

Figure 2. Display for analog automation in a nuclear power plant control room; in (a) the control goal is satisfied; in (b) the control goal is not satisfied.

because not all of the signals in the feedback loop are shown to the Operators. We would thereby expect that they would have problems in monitoring the status of the automation and detecting or diagnosing failures directly when they occur. More specifically, with the Demand signal illegible and the Manipulated Variable missing, the connection between the Controller and the Process is masked. Detecting a problem that the Controller cannot handle would be very difficult. Also, because the state of the Manipulated Variable is an output from the Final Control Components and an input to the Process, following the logic of analytical redundancy we would predict that the absence of this signal means that Operators cannot reliably and directly distinguish between problems that take place in the Final Control Components and problems with the Process.

Two hypothetical examples can illustrate the rationale behind this prediction. Take the case of a vessel with an automated level Controller. If there is a leak in the vessel (the Process, in this case), the Controller will compensate by adding more water to keep the level (the Controlled Variable, in this case) at the desired Set Point. As long as the Controller is able to compensate for the leak, the Display will be in the state shown in Figure 2a because the goal will be satisfied. Based on these indications, it is not possible to detect that there is a failure in the Process (the leak). Everything looks normal because the thin red bar is in the thick green band. It is also not easy to tell that the Controller may be working very hard to compensate for the failure because the Demand signal (which will increase in magnitude to compensate for the leak) is not legible from the Operators' usual location. In the absence of the proper feedback, automation can mask Process failures until the automated Controller is no longer able to compensate, and this has been noted in other industries as well (Norman, 1990). Thus, because the Demand signal is not very salient and the Manipulated Variable is not displayed at all, fault detection problems for the Operators are created under this scenario.

Consider the same level controller in the case in which the goal is not satisfied (i.e., the state of the Controlled Variable does not match the Set Point value). In this case, the Display will be in the state shown in Figure 2b. If the Display remains in this state, then the Operator can detect that there is a problem. However, because the Manipulated Variable is not shown, it is not possible for the Operator to determine from this Display exactly what the problem is. For instance, it could be that there is a large leak in the vessel (the Process) and that the Controller is functioning normally but is simply not capable of compensating for this leak. Alternatively, it is possible that there is no problem at all with the Process but that the Final Control Component (e.g., an input valve) has failed, causing the water level to remain higher or lower than the current Set Point value. The Display does not show the state of the Final Control Component, so it does not support Operators in discriminating between these two potential root causes. Note, moreover, that it is very important to make this discrimination because the countermeasures that need to be taken are radically different in the two cases (i.e., fix the leak in the case of the Process failure vs. repair or replace the input valve in the case of the Final Control Component failure). Thus, under this second scenario, the fact that the Manipulated Variable is not displayed also creates significant detection problems for Operators.

The logic of analytical redundancy built into the control-theoretic model helps explain some of the findings in this nuclear power plant control room. Operators reported that they frequently relied on "beetle" alarms *instead* of the automation Displays (see Figure 2) to detect or diagnose problems. Beetles are water-sensitive detectors that are placed in certain locations in the Plant. When a beetle alarm goes off, it is an indication that water is in a place where it should not be (e.g., because of a leak). Interestingly, however, the beetle alarms do not provide enough information for Operators to diagnose directly what is going on. Instead, they provide a coarse indication that there is an abnormality. The Operators then try to reason backward to see if they can figure out where the water is coming from and why. The root cause can actually be very distant from the location of the beetle that signals the alarm. Note also that the beetle alarms provide a delayed indication of a problem: An alarm is generated only when water reaches the location of a beetle. Thus the Operators may

already be well behind the pace of an event by the time they receive a beetle alarm.

The control-theoretic framework explains the rationale behind the operators' diagnostic strategy. The Displays that are available to monitor the status of the automated Systems do not provide all the information that is needed to detect and diagnose problems directly. As a result, the Operators rely more on other, secondary sources of information (the beetle alarms) that provide a more reliable indication of problems. The disadvantage of this approach, however, is that these secondary sources of information are both spatially and temporally removed from the locus of the root problem. Consequently, Operators engage in elaborate reasoning strategies to try to trace back to the origin of the abnormality. This reasoning process is prone to errors and takes time. Moreover, fault detection is already delayed by the fact that the beetle alarms may go off well after the problem has begun. As a result, Operators are frequently forced to respond in a reactive, rather than an anticipatory, manner. It should not be surprising, therefore, that incidents occurred in which the Plant gradually drifted from where it should have been without anyone noticing for an extended period. For all of these reasons, the displays that are currently available in these nuclear power plants to monitor automation are far from ideal.

Conclusions

The framework in Figure 1 provides an effective conceptual basis for interpreting some of the results obtained by Mumaw et al. (2000) and Vicente et al. (2001, 2004). The framework provides a referent against which existing displays can be evaluated. It also provides a basis for predicting the types of problems that are likely to be encountered if the displays are found to be deficient, as they were in this case. Finally, a control-theoretic perspective also provides a basis for recommending design changes that should lead to enhanced performance. In this case, making the Demand signal more salient, presenting the state of the Manipulated Variable, and showing the relationships among all of the signals in Figure 1 should provide Operators with a more robust basis for dealing with abnormalities, whether they occur in the Controller, Final Control Components, or Process.

MODES IN AVIATION AUTOMATION

In this section, we show the value of the control-theoretic framework for the problems of mode transition, mode awareness, and mode error. In doing so, we illustrate how the framework is relevant to different design problems (automation interface design vs. mode design) and to different application domains (nuclear power vs. aviation).

Why Study Modes?

Mode-related issues are perhaps the most notorious of automation's side effects, having received a wealth of attention from human factors professionals, particularly in the aviation domain. However, mode proliferation and its associated problems are not restricted to aviation. Evidence for this phenomenon has been provided in the maritime (Lee & Sanquist, 1996) and medical (Cook, Potter, Woods, & McDonald, 1991) domains as well (however, see Jamieson & Guerlain, 2000). It is anticipated that solutions that apply to the mode problem in aviation will apply to many other complex systems. More important, we anticipate that the concepts put forth in this article will be applicable to other human-automation interaction problems across domains with similar control characteristics. For all of these reasons, modes provide a challenging and potentially quite useful test bed for evaluating the power of our control-theoretic framework.

What Is a Mode?

Given the prevalence of labels incorporating the word *mode,* one might expect modes to be well defined and understood. However, some discussions of mode error and mode awareness take place without characterizing modes themselves (see Degani, 1996, and Leveson & Palmer, 1997, for exceptions). In fact, the most frequent reference to the term *mode* (at least in a human factors context) fails to clearly distinguish modes from mode errors (see Norman, 1988, p. 179). We attempt to remedy this trend by providing an explanation of a set of automation-related phenomena: modes, mode transitions, mode awareness, and mode error.

A mode is a manner of device behavior (Degani, Shafto, & Kirlik, 1999). We interpret "manner of device behavior" to mean that an element

in the feedback loop can deliver a signal output that varies as a function of its mode state for a given signal input. A device may have multiple modes of operation, although only one mode may be active at any given moment. Operationally speaking, modes define sets of mutually exclusive behaviors (Leveson & Palmer, 1997). These multiple manners of behavior can serve at least two purposes. First, they can allow one device to accomplish multiple tasks. For example, a digital watch might have multiple modes to perform four tasks: tell time, set an alarm, display the date, or act as a stopwatch. Second, multiple modes can allow a device to accomplish a single task using any one of a number of strategies. For example, a digital watch might allow the user to tell time by choosing between civilian or military time formats. The first example demonstrates how modes can expand the number of tasks a device can perform (multitask mode), whereas the second demonstrates how modes can expand the number of strategies a device can employ to accomplish a single task (single-task/multiple-strategy mode). This important distinction between tasks and strategies is rarely, if ever, drawn in the literature on modes.

Modes can be present in the Plant, the Controller, or the Interface. For example, an automobile transmission (Plant) can be configured in three modes: neutral, forward (perhaps in more than one gear), or reverse. A flight management system (Controller) can be configured to adjust speed via pitch or thrust. A cockpit display for radar (Interface) can be set to provide rose (360°) or arc coverage. Note that in the flight management system (FMS), the mode of control does not change the mode of the aircraft itself (i.e., the Plant). The various Controller modes are imposed by designers who try to anticipate behaviors that the pilots will want the Plant to assume. Similarly, having a radar's Interface present data in an arc rather than a circle does not change the manner in which the radar itself (the Plant) operates. The various Display modes are imposed by designers who try to anticipate the structure of information that the flight crew will want to observe.

Some of these distinctions have been recognized in the literature. For example, Sarter and Woods (1992a) distinguished between system (which appears to correspond with Controller) and interface modes. Also, Degani and Kirlik (1995) separated interface and control modes. Note, however, that none of these authors explicitly distinguished the third category of modes: those of the Plant. This distinction is important because it leads to another principle that is frequently overlooked in the literature: Modes are not just a feature of automation. They are a potential feature of all three System elements independently and should be dealt with as such by designers.

Mode Transition

Because a Plant, Controller, or Interface can exhibit several modes of operation, designers must also provide for transitions between modes. In most contexts, this means that users are required to take actions to make such transitions. However, as the applications in which mode options are available become more complex, transitions also become more complex (Degani, 1996, 2004). Plants, Controllers, and Interfaces are capable of partially and fully autonomous mode transitions. The variety of automated subsystems, their interrelated mode settings, and the mixed autonomy of transitions can increase overall System complexity exponentially (Sarter & Woods, 1997).

There are two primary types of mode transitions in an automated device: commanded and uncommanded (Vakil, Hansman, Midkiff, & Vaneck, 1995) or, equivalently, manual and automatic (Degani et al., 1999). A commanded or manual mode transition is executed in response to Operator input. An uncommanded or automatic transition is engaged by the automation and can occur in response to some sort of envelope protection, loss of data, or the satisfying of a precondition. Furthermore, both Vakil et al. and Degani et al. distinguished a hybrid subtype of mode transition. An automatic/conditional mode transition (Vakil et al.) occurs when, after operating in a given mode up to some target state specified manually (e.g., an altitude), the automation switches modes. The Degani et al. description of automatic/manual mode transitions is similar but somewhat broader in that it allows for manual engagement of the mode transition without reaching the target state.

Although automatic and automatic/conditional mode transitions may occur without direct

Operator request, it is important to note that these "automatic" mode changes occur in response to designer requests, if not Operator requests. The state conditions under which the transitions take place are formally defined in the design of the device and are thus subject to redesign and/or description in a Display (Degani & Heyman, 2002). Experimental evidence indicates that Operators are more likely to lose track of modes when nondirect transitions take place (Sarter & Woods, 1992b).

Mode Awareness

With mode-induced flexibility comes an added responsibility: management (Sarter & Woods, 1994). Once given the opportunity to choose a mode, an Operator must make a decision, keep track of it, and act in a way that is consistent with it. This added responsibility is frequently referred to as *mode awareness:* "knowledge and understanding of the current and future status and behavior of automated...systems" (Sarter, 1995, p. 239).

Mode confusion is the failure to maintain mode awareness – a misidentification of machine behavior and transitions between behaviors (Degani et al., 1999). Sarter and Woods (1995b, 1997) have referred to the unanticipated System response associated with mode confusion as "automation surprises." Note that mode confusion does not necessarily correspond to mode error unless an erroneous action is taken.

Mode Error

A mode error occurs when an Operator takes an action that is appropriate in one mode but inappropriate in the present (active) mode (Degani et al., 1999; Norman, 1988) or fails to take an action that would be appropriate in the active mode. Mode error has been cited as a contributor to a number of fatal aviation incidents ("Automated Cockpits Special Report," 1995a, 1995b; Billings, 1997; Degani, 2004), has been shown to be a major factor in pilot-automation interaction difficulties (Sarter & Woods, 1994), and has been recognized as a problem with successive generations of cockpit automation (Sarter & Woods, 1995a, 1997). Note that mode errors can take the form of errors of omission or errors of commission. Operators can fail to intervene when automation takes an inappropriate action (omission), or they can engage a mode that is not appropriate for a given situation (commission; Sarter & Woods, 1992a).

Modes in the Feedback Control Process Model

Figure 3 demonstrates how the presence of modes can complicate the negative feedback model introduced earlier, in Figure 1. The relationships between input and output signals are now conditional upon the active mode. Figure 3 reflects this conditionality by segmenting System elements into modes using dotted lines and adding different signal paths between elements for each mode segment. Thus, depending on the active mode of Controller operation, the Demand signal may change for the very same Error signal (e.g., an FMS can send Demand signals to the engines or to the control surfaces, depending on the engaged vertical navigation mode). Similarly, depending on the Display mode, the same

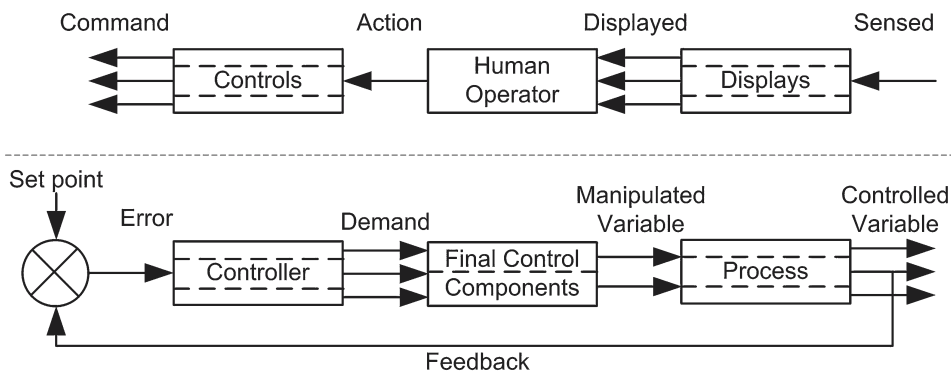

Figure 3. A negative feedback control loop for a System with modes.

Sensed signal might be Displayed differently (e.g., a signal from a radar unit is configured in either arc or rose format).

The control-theoretic perspective led to the five generic research and design implications listed in the left column of Table 2. These implications are expressed more specifically for the problem of modes in the right column of Table 2. First, the need to distinguish between System elements is increased. Because each element can have multiple modes, identifying those elements requires that their modes be distinguished as well. Using the catch-all label *system* becomes even more imprecise as it now refers not only to multiple elements but possibly to multiple modes within elements as well. Second, there is an additional potential new source of failure. Whereas before we noted that failure could occur at any System element, there is now the added potential for failures localized to modes within elements. Third, locating a fault in the System is potentially much more difficult because the number of signal paths at which failures can occur has been multiplied. Before, normative models of the System elements were required to achieve analytical redundancy. Now, normative models are needed for each mode configuration of each System element. Fourth, the number of degrees of freedom open to the automation designer has increased. Not only is the design of the Controller open to human factors considerations; so is the selection of mode configurations to be incorporated in that Controller. Fifth, if one wishes to pursue the idea of making the internal structures of the elements transparent, more details must be specified (i.e., the various mode-induced relationships between signals).

Summary

The control-theoretic approach provides a framework that highlights a number of issues for research and the design of human-automation interaction. By speaking in the language of control theory, the approach draws new connections between the often disparate perspectives of automation designers and human factors engineers. In the section on nuclear power plants, for example, we demonstrated how underspecified automation Displays can make monitoring and fault diagnosis in automated Systems difficult or impossible for Operators. In addition, our discussion of modes, their associated phenomena, and their impact on the feedback control process model demonstrated that evolutionary trends in automation technology pose increasingly complex challenges for both researchers and designers. Thus our application of the well-known control-theoretic framework provides novel insight into how existing applications of automation technology often do not work for people. Our ultimate goal, however, is to support designers in employing automation in ways that are more beneficial to Human Operators. In the following section, we will use the framework to identify principles for the design of human-automation interaction.

AUTOMATION DESIGN PRINCIPLES: BEYOND MENTAL MODELS AND FEEDBACK

There are two predominating design principles for improving human-automation interaction (e.g., Sarter & Woods, 1995b, 1997). Mode awareness problems and mode error can be reduced if (a) the operator's faulty mental model of the "system" is corrected and (b) more effective feedback about what the automation is doing is provided. Although these principles are useful, they are limited. Additional design principles that are made explicit by the control-theoretic framework can also be used to improve performance. In this section, we discuss how the framework can expand and clarify the existing principles. We also show how the framework leads to four new design principles with which human factors engineers can engage automation designers in the pursuit of improved human-automation-plant interaction.

- Principle 1: The Human Operator must have accurate mental models of both the Plant and the Controller.

Research has shown that mode confusion leads to the following questions about automation: "What is it doing now?," "Why is it doing that?," "What will it do next?," and "How in the world did we get into this mode?" (Sarter & Woods, 1995b). From these questions, it seems that Operators are confused about how the observed behavior of the Controller relates to System goals. That pilots would be confused about

which components the automated actors use to take action seems unlikely because Demands are executed using the same Final Control Components as manual control actions (Norman, 1990). What is the source of the confusion, then? How should automation designers address these questions? Consider the following:

> If pilots were provided with an overall mental representation of the functional structure of the FMS, they would be better able to manage and utilize the automated systems in unusual or novel situations. Given that their role has shifted toward the detection of deviations from the expected and toward troubleshooting and managing such situations, this capability seems to be very important for pilots in highly automated aircraft. (Sarter & Woods, 1992b, p. 320)

The meaning of the phrase "overall mental representation of the functional structure" is clarified by the following: "The monitoring of an automated system requires an adequate mental model of the structure of the system" (Sarter & Woods, 1995a, p. 182). This is a statement with which we agree; however, the "system" to which the authors are referring seems to be the Controller (i.e., the FMS). Although it is certainly necessary to make explicit to the Operator the way in which the automation functions, the control-theoretic perspective reveals that this design guideline should be extended. Operator confusion can arise because the source of the behavior of an element of the System cannot be easily or accurately determined. Therefore, the Human Operator must understand the attributes and transfer functions that define both the Plant (i.e., the aircraft being controlled) *and* the Controller.

Taken individually, calls for accurate mental models of the Plant or Controller are not new. However, the literature on human-automation interaction in the aviation domain places a great deal of emphasis on mental models of automation and neglects models of the Plant (e.g., Sarter & Woods, 1994; Vakil & Hansman, 2002). This tendency may reflect an assumption that pilot mental models of aircraft are veridical. In fact, we (and all of the aviation automation experts we consulted) have not identified any experimental studies of the adequacy of pilots' mental models of the aircraft (engines, hydraulics, electrical, fuel, etc.). Accident investigations indicate that pilots sometimes do not have accurate mental models of the Plant (Besnard, Greathead, & Baxter, 2004). As Palmer and Abbot (1994) pointed out, aircrew "failure to understand the implications of certain system failures on the capability of other aircraft systems has been cited as a contributing factor in several accident and incident cases" (p. 2). Thus we would be remiss to exclude the operator's mental models of the Plant in developing a systematic design approach for automated Systems.

Alternatively, the neglect of Operator mental models of the Plant may reflect a belief that human-automation interaction difficulties are best addressed by improving mental models of automation. Although automated Systems indeed necessitate an accurate mental model of the Controller, the model of the Plant is no less important. In fact, as the next design principle demonstrates, mental models of Plants are possibly more important for automated Systems.

- Principle 2: All stakeholders must agree on a shared model of the Plant.

Vakil et al. (1995) pointed out that aviation automation has evolved progressively, adding new features in generational increments without surrendering older features (see also Billings, 1997). "The entropic growth has created a system [i.e., Controller] that appears to lack a simple, consistent, global model. This lack requires pilots to create their own ad-hoc models" (p. 244). Vakil et al. also stated that there is a "lack of underlying structure to the automation, making it difficult for pilots to develop consistent mental models" (p. 243).

From the perspective of the control-theoretic framework, these two statements prompt a few clarifications. First, some sort of structure must exist in the automation. Otherwise it would not be capable of reducing the degrees of freedom in the Plant to that required for action. Disorder can be reduced only by regularity (i.e., structure). From the viewpoint of a Human Operator, that structure may look more like a plate of spaghetti than a dedicated support system, but it is nevertheless still structure. Second, if the issue is about modeling complex systems, according to the law of requisite variety a simple model is not likely to be of much use: Complex systems require complex controllers (Ashby,

1956). Therefore, overly simplified models may actually be misleading to Operators and, thus, detrimental to performance. Third, if one infers that by "consistent" and "global" Vakil et al. (1995) meant internally coherent and accurate, this presents another problem. It seems that the spaghetti-like structure of existing aviation automation is a byproduct of an evolutionary design process. This sort of "legacy" structure can be faithfully modeled, but it will still appear as ad hoc as the process that produced it. In other words, a good model cannot compensate for the structural complexity of the modeled entity (i.e., in this case, the FMS).

A complementary approach would be to ensure that the automation has an adequate and explicit "mental" model of the Plant itself (see Besnard et al., 2004; Conant & Ashby, 1970; Leveson & Palmer, 1997). Under this view, automation designers would have to cooperate with Plant designers to ensure that the Controllers and the Plant are driven by integrated functions and compatible intentions (Rasmussen & Goodstein, 1987). In such a design approach, the interface designer's responsibility would be to ensure that the model of the Plant that served as the basis for designing the Controller – including the designers' intentions – is available to the Operator. In this way, the structure (and emerging behavior) of the Controller can be related to the functionality of the Plant (which would also be modeled in the Interface).

In other words, the interface designer should externalize both a model of the Controller (e.g., Vakil & Hansman, 2002) and a model of the Plant (Vicente & Rasmussen, 1992). Doing so would allow Operators to visualize how their own Actions and the Commands issued by the automation serve the higher level purposes of the Plant and meet the shared goals of the Operator and Controller (thereby supporting Design Principle 1). Ideally, a single model of the Plant would be (a) used to design the Controller, (b) represented in the Interface, and (c) used by Operators to understand the Plant. Such an approach should ensure that Plant, Controller, and Operator have a common, integrated design basis that provides for effective coordination among all three elements (Rasmussen, 1986; Rasmussen, Pejtersen, & Goodstein, 1994; Vicente, 1999; see also Holder & Hutchins, 2001).

- Principle 3: Create analytical redundancy in the Interface.

Accounts of mode confusion or mode error often lead to statements about improving the feedback provided by the Interface. For example, "To support the increased need for coordination, systems would need to be more transparent. Feedback design would need to be improved to support the human operator in keeping track of and in anticipating the status and behavior of his machine counterpart" (Sarter & Woods, 1995a, p. 181). This recommendation has merit. If Operators are not in touch with what is taking place in the System, then an appropriate response is to provide more or better data. However, in most aviation incidents in which automation was implicated, the data required to detect problems were available (Norman, 1990; Sarter & Woods, 1995b). Calling for improved feedback begs the question, "Feed back what?"

The quick answer is, "Feed back the structure and state of the automation." In fact, Sarter and her colleagues have shown that providing tactile (Sklar & Sarter, 1999) or peripheral visual (Nikolic & Sarter, 2001) feedback of uncommanded mode changes (i.e., state information) can improve the detection rates and reaction times of pilots who are not experienced with advanced automation. Earlier, however, we likened the legacy structure of modern aviation automation to a plate of spaghetti. Making such an automation structure transparent is likely to result in "visible spaghetti" and may be as likely to contribute to Operator confusion as it is to resolve it. We stated before that good modeling cannot make up for structural complexity in the entity being modeled – neither can good Interface design. If the automation is structurally complex, then making the automation more transparent will only reveal that complexity.

There is a second limitation to the "improve feedback" design guideline. "What is needed is continual feedback about the state of the system, in a normal natural way, much in the manner that human participants in a joint problem-solving activity will discuss the issues among themselves" (Norman, 1990, p. 591). This statement raises a new question: How should feedback be presented to the Operator? The challenge to Norman's (1990) recommendation is that Human

Operators seem to be far more effective at communicating with each other than with automation. How can this be replicated? "The task of presenting feedback in an appropriate way is not easy to do. Indeed, we do not yet know how to do it" (Norman, 1990, p. 591). The issue, as Norman (1990) pointed out, is that automation is not sensitive enough to provide information to the Operator that is *relevant to the context.* Feedback needs to be meaningful, not just available. The problem is not that the Operators do not have enough data but that they do not have enough information. The effectiveness of data depends on the resources required to transform them into information (Sarter & Woods, 1997; Woods, 1996): "Feed back? Yes, but how?"

Both of the questions emanating from the preceding discussion can be couched in terms of a control-theoretic perspective. Specifically, which signals in Figure 1 or 3 need to be fed into the Display element (feed back what?), and in what form should they be Displayed (feed back how?). In terms of feedback content, the analytical redundancy approach provides concrete design direction about what must be provided. In Systems that require Operators to shut down the Plant when there is an automation failure (as in some nuclear power plants), then the operator need merely be informed that the error signal is outside the acceptable range and provided with a Control to command disengagement of the automation. However, in Systems where Operators must diagnose the root cause of the fault to compensate effectively, then the Interface must support analytical redundancy for every element in the System by making all of the signals in the feedback control loop accessible to the Operator. The output signal from each element must be compared with the expected value, given the input signal and a normative model of the element. Moreover, in mode-capable Systems, normative models of each mode are required as well. This technique will allow the Operator to localize sources of disturbance in the System in cases where such localization is required.

It should be noted that support of analytical redundancy would necessitate a substantial expansion of the information presentation requirements for an Interface. This challenge must be met head on because to omit information is to create an underspecified situation in which operators would be "blind" to what is really going on (see the earlier nuclear example). Emergent feature displays (Bennett & Flach, 1992) and context-sensitive information presentation (Woods, 1991; Woods, Patterson, Roth, & Christoffersen, 1999) are potential means for presenting all of the necessary information in a manageable form. The control-theoretic framework does not offer a selection or prioritizing framework for managing this expansion. It can, however, assist the human factors engineer in communicating to automation designers the information cost of increasingly complex Controller design (see Principle 4).

The manner in which the proposed framework answers the "how" question was discussed in the Design Principle 2 section. The appropriate means of communicating among designers, automation, and Operators is through a shared model of the Plant (Rasmussen & Goodstein, 1987). To the extent that all three of these actors can "converse," having a common understanding of the functions of the Plant, they will be communicating information as opposed to mere data.

- Principle 4: Reduce the structural complexity of the Controller.

Making automation visible and providing Operators with proper mental models may lead to only modest gains. Larger improvements in performance may be achieved by making the underlying structure of the Controller simpler to begin with. As Leveson, Pinnel, Sandays, Koga, and Reese (1997) pointed out,

> If it is true that mode-related problems are caused by clumsy or poorly designed automation, then changing the human interface, training, or operational procedures is not the obvious, or at least the only solution. Instead, if we can identify automation design characteristics that lead to mode awareness errors or that increase cognitive demands, then we may be able to *redesign the automation* without reducing system capabilities. (p. 134, italics added)

The control-theoretic perspective makes it clear that several different Controllers can be developed for a given Plant, thereby opening the door to redesigning the structure of the automation. Thus, rather than being an unalterable

given, the functionality of the Controller should be treated as something to be designed to make Operators' jobs easier. The perspective that we have taken reveals two ways in which this can be accomplished.

The first, and simplest, suggestion for limiting mode confusion is to reduce the number of modes. Figure 3 demonstrates that such an approach would reduce the number of possible signal paths through the System. Although this suggestion is not new, no criteria have been established for determining how many modes are too many. The control-theoretic perspective suggests that having significantly more Controller or Interface modes than Plant modes may be indicative of excessive flexibility. Automated Controllers and Interfaces that offer a range of modes that substantially exceed those inherent in the Plant give the user flexibility at the cost of increased management. If designers can match Controller modes to Plant modes, then there will be natural mappings between System elements. The ultimate design goal, then, may be to reduce the number of Controller and Interface modes to the number of Plant modes.

Second, automation designers must recognize the trade-off between flexibility and management responsibility in mode-capable Systems; an increase in the former invariably incurs more of the latter. One key to this trade-off is the distinction between multiple task modes and single-task/multiple-strategy modes. In the case of single-task/multiple-strategy modes, designers must be able to justify inclusion of such flexibility. Providing "another way of doing the same task" cannot be considered sufficient justification for burdening Operators with additional monitoring duties. Therefore, single-task/multiple-strategy modes should be used judiciously.

Although it remains to be demonstrated that these two tactics would improve human-automation interaction in aviation, some field observations suggest that their application in the petrochemical processing industry mitigates mode confusion problems. Jamieson and Guerlain (2000) presented a field study of operator interaction with model-based predictive controllers, a form of advanced automation widely used in the process industries. Although they observed many of the same monitoring and interaction difficulties attributed to pilots of highly automated aircraft, mode confusion was not a serious problem for the refinery operators. An evaluation of the design characteristics of the automation showed that the mode structure is consistent with the preceding recommendations, even though this was not the explicit intention of the automation designers.

The approach that we are advocating requires that human factors engineers have a say in what have traditionally been viewed as strictly "technical" issues (see Vicente et al., 1996). Although it would certainly be difficult to convince manufacturers to redesign their FMSs, there are some in the aviation industry who are advocating exactly that (e.g., Riley, 1996, 2001; Vakil & Hansman, 2002). Like them, we believe that much could be gained by redesigning the functionality of the automation to make it structurally less complex, in addition to making it more observable. Then, providing feedback will probably have synergistically bigger payoffs because the mental models describing Controller functioning will be psychologically more comprehensible and manageable.

An Example

We have introduced four design principles derived from the control-theoretic framework that are more specific than many of the human-automation design guidelines currently available in the literature. In the remainder of this section, we present an example to demonstrate these principles in action.

The Process. The problem in our example is to design a Controller and Interface for the simple chemical Process shown in Figure 4. The Process consists of a chemical reactor, two reactant flows into the reactor, one product flow, and two control valves. Reactants A and B flow into the reactor, where they react, causing an increase in pressure. The flow of Reactant B controls the rate of reaction. Following the reaction, the product either flows on to another part of the System for continued processing or through a flare valve that safely vents to the atmosphere.

Plant modes. The Plant can be operated in one of four modes: normal, start-up, shutdown, and maintenance. We will consider the first two of these modes here. During normal operation mode, a high flow rate of Reactant A (Controlled Variable 1) contributes to Process stability.

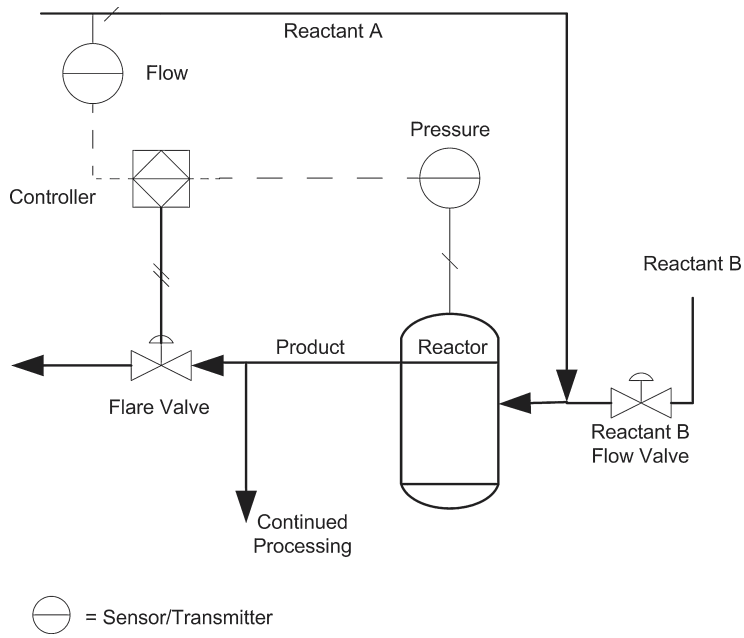

Figure 4. A simple chemical Plant and Controller.

High reaction rates make the Plant profitable; however, they also increase the pressure (Controlled Variable 2) in the reactor, which can become dangerously unstable at extreme pressures. Pressure in the reactor can be reduced by opening the flare valve (Final Control Component 1).

During start-up, the flow rate of Reactant A changes rapidly, thereby destabilizing the Process. During these transients, it is difficult to throttle the reaction using the Reactant B flow control valve (Final Control Component 2). A poorly throttled reaction yields low-quality product, which is discarded by opening the flare valve.

Application of Design Principle 1. The challenge is to develop a Controller that will lead to a productive and safe Process. Design Principle 1 states that meeting these objectives requires supporting Operator understanding of both the Controller and the Plant. In effect, the principle broadens the scope of the design problem from that typically faced by automation designers, who are usually concerned only with the design of the Controller. In the present case, the designers have not been given control over the design of the Plant and must focus on supporting effective Operator control through the design of the Controller and Interface. The optimal design for these two elements will not introduce complexity beyond that already existing in the Plant. The closer they come to reaching that goal, the easier it will be for the Operator to learn, control, and troubleshoot the System.

Consider now the introduction of a multiple-input/single-output Controller for the flare valve (see Figure 5 for the corresponding feedback control loop). The comparator compares two inputs against separate Set Points and sends two Error signals to the Controller. The first input, the Reactant A flow rate (Controlled Variable 1), is compared against an established low-flow limit (Set Point 1). The second input, the reactor pressure (Controlled Variable 2) reading, is compared against an established high-pressure limit (Set Point 2). The Controller calculates two positions (Demand) for the flare valve (Final Control Component), one for each input. The mode of the Controller determines which of those two Demand values will be sent to the valve. In normal operating mode, the Controller is designed to identify the greater of the two Demand values (flow or pressure) and issue a single Demand signal to the valve. The valve position (Manipulated Variable) will adjust to effect a change in the Process. Thus, if the flow of Reactant A is below the set limit, the Controller will signal the flare valve to open, thereby preventing low-quality

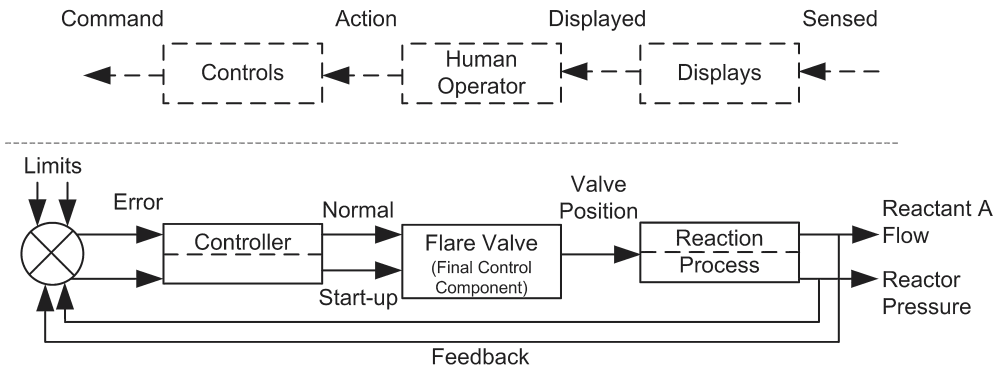

Figure 5. The negative feedback loop for the Process and Controller in Figure 4.

product from flowing downstream. Alternatively, if the reactor pressure is higher than the set limit, the flare valve will open to relieve that pressure safely. If both limits are exceeded, the greater of the two Demand values determines the valve position (thereby biasing safety over production).

In the transitory start-up mode, the designers program the Controller to ignore the Error signal for Reactant A flow. Thus the flare valve will not open if Reactant A flow is low. This allows the pressure in the reactor to climb and the flow to increase, establishing a stable state in the reactor. Should the pressure rise too far, the Controller will still open the flare valve.

Application of Design Principle 2. Design Principle 2 states that the stakeholders must agree on a shared model of the Plant that can serve as the basis for both the Controller and Interface designs. This requires a functional analysis of the Plant to identify the structure of the Process and Final Control Components. Included in this model is an identification of the inherent modes of the Plant, as discussed previously. With this shared model of the Plant, the Controller and Interface design can proceed independently under the knowledge that the two System elements will share compatible design intentions. Those intentions will be encapsulated in the design of both the Controller and the Interface. The same intentions might also be used to develop training programs and company or industry policies on automation use.

Application of Design Principle 3. Design Principle 3 states that the Interface must provide for analytical redundancy. This requires a consideration of the implications of analytical redundancy and also a specification of what type of interaction the Operators are expected to have with the System. The nature of that interaction will determine the extent of the information that must be made available in the Interface. For instance, if Operators are expected to monitor the performance of the Controller and be prepared to take over manual control of the System if the automation fails, then they will need all of the signals in the feedback control loop as well as normative models of the elements (see Figure 6, top panel). In contrast, if the Operator is expected only to monitor the System state and respond to an upset (i.e., low flow of Reactant A or high reactor pressure) or automation failure by manually opening the flare valve, then the Interface need provide only a Display indication of the two Controlled Variables and the Control mechanism to issue the Command to open the flare valve (see Figure 6, bottom panel).

The contrast between the information requirements for each of these expected responses is substantial. To accomplish the upset response task, the Operator need have only a very limited understanding of the state of the Plant and essentially no knowledge of the behavior of the Controller. A minimal amount of feedback for this task would include just three items. However, if the Operator is expected to engage in fault diagnosis and compensation, substantially more feedback is required. By comparing the Error signals and the Demand signal with the model of mode-dependent Controller behavior, the Operator can determine whether the Controller is performing as expected. Similarly, by comparing

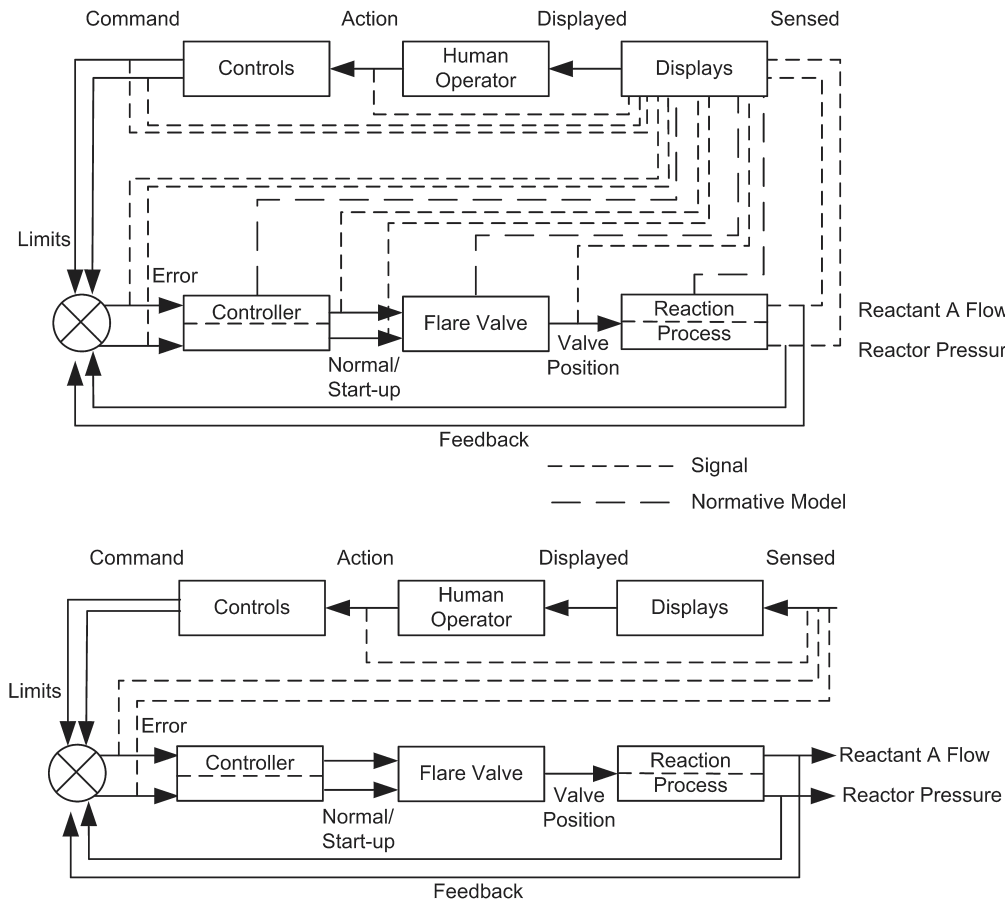

Figure 6. Comparison of Interface requirements for an Interface (top) capable of supporting troubleshooting and fault management through analytical redundancy and (bottom) capable of supporting only basic control responses.

the Demand signal, Manipulated Variable signal, and a model of the flare valve, the Operator can assess the health of this Final Control Component. Each of these is a critical step in localizing the cause of an upset that could be attributable to problems in the Plant, Final Control Component, or Controller. If any of the signals around these components is missing from the Interface, then the anticipated control task is underspecified and susceptible to failure. Moreover, if the Interface provides external, valid models of the Controller and the Plant, then it will have satisfied Design Principles 1 and 2 without resorting to extensive Operator training.

Application of Design Principle 4. The automation scheme could be made more complex by incorporating the Reactant B flow control valve into the Controller, as shown in Figure 7. The design would thus change from being a multiple-input/single-output Controller to a multiple-input/multiple-output Controller. Under this configuration, there are two possible Controller submodes for each of the Plant modes, as shown in Figure 8. For example, during start-up the Controller would allow for reactor pressure control via control of either the flare valve or the Reactant B flow control valve.

This automation configuration could be advocated on the grounds that (a) producing product during start-up would enhance efficiency, (b) flow control of Reactant B would reduce Operator workload in normal mode, and (c) the new submodes provide the Operators with greater flexibility of control. However, Design Principle 4 advises designers to consider whether the advantages of this added flexibility outweigh the

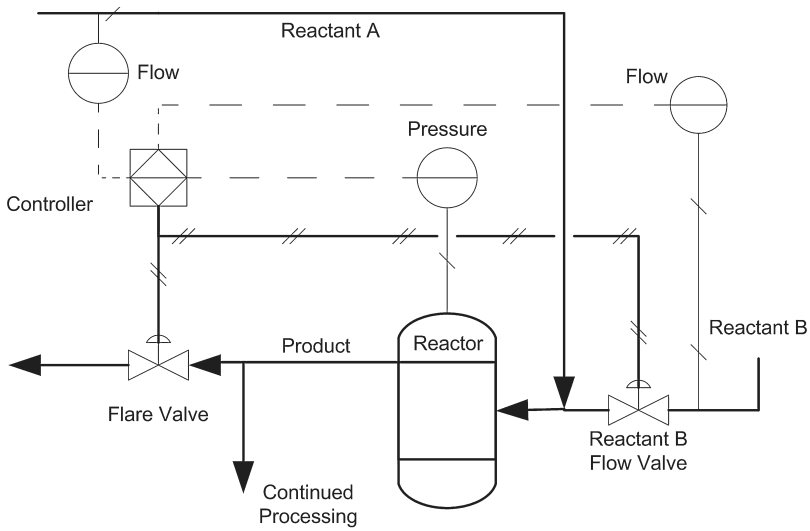

Figure 7. A more complex Controller for the chemical reaction Process in Figure 4.

expected costs. Expanding the number of Controller submodes (a) increases the complexity of the Controller, (b) creates stronger coupling within the Plant components, and (c) increases the mode management task.

Recall that the ultimate design goal derived from Design Principle 4 was to limit the number of Controller modes to the number of Plant modes. A choice could be made to eliminate the original normal mode of operation and adopt the new one, thus keeping a one-to-one mapping between Controller modes and Plant modes. This solution allows for some of the advantages of the more complex control scheme while limiting the disadvantages. Design Principle 4 is meant to support this process of trading off automation advantages and disadvantages.

Achieving Effective Human-Automation Design

The control-theoretic framework discussed in this article makes several contributions to the literature on human-automation-plant design. However, the preceding example illustrates that it is not a detailed, comprehensive methodology for the design of automated Systems. Continued development of the ideas put forth here is required to flesh out important details. Although it is beyond the scope of this article to elaborate, we can anticipate some of these developments.

We have explicitly called for the use of a Plant model that can be shared by designers, Controllers, and Human Operators. A strong candidate

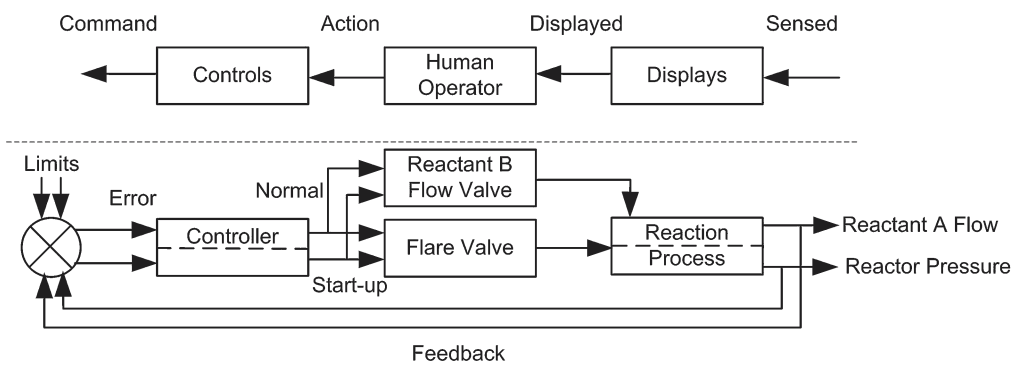

Figure 8. The negative feedback loop for the Process and Controller in Figure 7.

for such a modeling framework is the *abstraction hierarchy* (Rasmussen, 1985), which provides a psychologically relevant way of representing complex work domains. The abstraction hierarchy is a multilevel Plant model defined by means-ends relations between the levels that Operators can use for goal-directed reasoning. Because it exclusively models the Plant, the abstraction hierarchy can provide the common language through which these actors can communicate familiar functions and intentions. We have also stated that a model of the Controller is necessary. The *decision ladder* (Rasmussen, 1976) has been previously identified as a candidate framework for this job. It can be used as a template to identify the information-processing activities required for an actor to complete a task. The decision ladder demonstrates that these information-processing requirements are independent of the actor. For a given task, those requirements can be the responsibility of the designer, the Operator, or the automation (Rasmussen & Goodstein, 1987). This "actor independence" facilitates a common depiction of Controller operation and Operator behavior, a perspective we advocated earlier.

COMPLEMENTARY PERSPECTIVES

Most of the human-automation interaction literature has focused on identifying and characterizing interaction problems, classifying common symptoms and triggers across domains, and evaluating design manipulations in local contexts. In the late 1990s, several reviews of human-automation performance research appeared in the literature and outlined requirements for a "human-centered automation" philosophy (e.g., Billings, 1997; Parasuraman & Riley, 1997). In the past few years, several researchers have proposed systematic frameworks for addressing aspects of the human-automation interaction problem. In this section we consider two such efforts, compare their contributions with those of the control-theoretic framework, and note the human performance issues addressed by each.

Types and Levels of Automation

Parasuraman, Sheridan, and Wickens (2000) proposed a classification of automation into four types: *information acquisition, information analysis, decision selection,* and *action implementation* (see Lee & Sanquist, 1996, and Billings, 1997, for similar classifications). The classification reflects the rapidly expanding range of functions that can be fulfilled by machines, particularly in the emerging areas of information automation. By characterizing a System according to its level of automation (i.e., from low to high) in each type, Parasuraman et al. (2000) aimed to support designers in choosing what functions to automate. They proposed a series of steps by which this characterization can be used to inform the principled design of automation.

Although we have emphasized action implementation automation in this article, all four phases of the information-processing model map loosely onto elements of the feedback control model. Information analysis takes place in the Displays element as raw data are integrated and presented to the user in a form that (one hopes) aids the user in performing system identification. The analytical redundancy postulate speaks directly to how this type of automation should be designed. The Controller can be seen as the decision selection element, evaluating current Controlled Variable states against the Set Points and deriving Demand signals based on stored algorithms. Information acquisition takes place throughout the feedback loop as signals are collected for presentation in the Displays.

The mapping between the models is not a clean one, nor should it be expected to be. The four-phase model describes types of automation in terms of open-loop human information processing, whereas the feedback control model describes the relationships between elements and signals in the context of a closed-loop manual control problem. Despite this fundamental difference in perspective, the models address complementary human-automation interaction issues. For example, Parasuraman et al. (2000) demonstrated that the four-phase model can be used to integrate performance issues such as mental workload, complacency, and skill degradation. We have shown that the control-theoretic framework bears on monitoring and fault management. The models also differ in terms of the uses to which they are best suited. The four-phase model is promoted as a means of determining "which functions should be automated

and to what extent" (Parasuraman et al., 2000, p. 286). The control-theoretic framework focuses on promoting a more coherent and systematic approach to the design of Interfaces.

The four-phase model reflects a high-level approach to the problem of designing for effective human-automation interaction in automated Systems. It considers a broad range of human-automation interaction difficulties and provides an outline of how types and levels of automation can be used effectively in scoping a novel System. Presumably, as the framework evolves, more detailed design questions would be encountered and resolved. This approach contrasts with that taken by Degani (2004) and Degani and Heymann (2002), who have proposed a framework that was first used to address the specific problem of mode transitions (see next section). Ongoing work is expanding the implications of the framework to a broader range of Systems and issues.

Formal Verification of Mode Transitions

Degani and Heymann (2002) presented a formal approach for verifying human-machine systems. They employed finite-state machines to model the state transition behavior of "machines." The same framework can be used to describe a user model (i.e., the user's model of the machine) and the behavior of an Interface. An algorithm can then be applied to rigorously compare the behaviors of the user model, Interface, and "machine" for a given set of anticipated tasks. Discrepancies in the user model or Interface pose challenges to the designer that would result, if left unresolved, in an underspecified control problem.

Degani and Heymann (2002) demonstrated use of the state transition method to ensure that the Interface to an automated System sufficiently describes the mode transition behavior of the Controller. The approach can also be used to inform the development of verifiable training programs (i.e., the means of forming the user model) for Human Operators. Whether or not finite-state machines can account for human-automation performance problems remains to be demonstrated.

Although Degani and Heymann (2002) exclusively discussed Controllers as examples of "machines," their approach is equally amenable to Plant descriptions (see Degani, 2004, for examples). Given Plant, Controller, and Interface models for a single System, the state transition framework might generate insights that are consistent with the control-theoretic framework, although this remains to be shown. However, each framework is better suited for a different type of System. On the one hand, the control-theoretic framework was conceived to describe dynamic System behavior, whereas the state transition framework is less amenable to comprehensive modeling of Systems with continuous or time-dependent behaviors (see Degani & Heymann, 2002; however, see Oishi, Tomlin, & Degani, 2003). On the other hand, finite-state models afford an exhaustive treatment of mode transition behaviors, whereas the feedback control model highlighted only qualitative impacts of mode behavior. Thus the strengths and weaknesses of the state machine and control-theoretic perspectives are remarkably complementary.

Summary

Parasuraman et al. (2000) stated that the four-phase model is a "simple starting point with surprisingly far-reaching implications for automation design" (p. 288). The same can be said about the feedback control or finite-state machine models. Each of these models sheds light on some human-automation interaction issues, yet none offers a comprehensive solution for the challenge of designing human-centered automation. Although the present state of knowledge in the human factors discipline does not support an integrated framework for human-automation interaction, this is clearly a goal to which researchers should aspire.

CONCLUSIONS

In this article, we have presented a systematic theoretical framework that clarifies the elements and actors that must be coordinated for automated Systems to function effectively. The control-theoretic perspective serves researchers by offering some conceptual clarity and precision and by suggesting focal points for further investigation. In addition, it serves designers by clarifying existing design recommendations and by offering new implications that can lead to improved human-automation interaction. Because

it provides a broader and more systematic view, the framework reveals design possibilities (e.g., externalizing models of the Plant and the Controller; redesigning the Controller to make it less structurally complex) that have not been made explicit previously. Consequently, this control-theoretic framework provides a valuable, complementary contribution to the existing research on human-automation interaction.

More work remains to be done before a detailed, comprehensive design methodology for automated Systems can be achieved. A productive path toward this goal is to adopt automation designers' own language – control theory – to explain why human factors issues are so important and how such issues can be effectively integrated with traditional control engineering concerns. The end result should be more effective human-automation-plant interfaces.

ACKNOWLEDGMENTS

This research was sponsored in part by grants from the University of Toronto Connaught Foundation and the Natural Science and Engineering Research Council of Canada. Thanks to Asaf Degani, Stephanie Guerlain, John Lee, Deb Mitta, Vic Riley, and the reviewers for their comments on previous versions.

REFERENCES

Ashby, R. W. (1956). *An introduction to cybernetics.* London: Chapman and Hall.

Automated cockpits special report, Part I. (1995a). *Aviation Week and Space Technology, 142*(5), 52–65.

Automated cockpits special report, Part II. (1995b). *Aviation Week and Space Technology, 142*(6), 48–57.

Bainbridge, L. (1981). Mathematical equations or processing routines? In J. Rasmussen & W. B. Rouse (Eds.), *Human detection and diagnosis of system failures* (pp. 259–286). New York: Plenum.

Bennett, K. B., & Flach, J. M. (1992). Graphical displays: Implications for divided attention, focused attention, and problem solving. *Human Factors, 34,* 513–533.

Besnard, D., Greathead, D., & Baxter, G. (2004). When mental models go wrong: Co-occurrences in dynamic, critical systems. *International Journal of Human-Computer Studies, 60,* 117–128.

Billings, C. E. (1997). *Aviation automation: The search for a human-centered approach.* Mahwah, NJ: Erlbaum.

Conant, R. C., & Ashby, W. R. (1970). Every good regulator of a system must be a model of that system. *International Journal of Systems Science, 2,* 89–97.

Cook, R. I., Potter, S. S., Woods, D. D., & McDonald, J. S. (1991). Evaluating the human engineering of microprocessor-controlled operating room devices. *Journal of Clinical Monitoring, 7,* 217–226.

Degani, A. (1996). *Modeling human-machine systems: On modes, error, and patterns of interaction.* Unpublished doctoral dissertation, Georgia Institute of Technology, Atlanta.

Degani, A. (2004). *Taming HAL: Designing interfaces beyond 2001.* New York: Palgrave Macmillan.

Degani, A., & Heymann, M. (2002). Formal verification of human-automation interaction. *Human Factors, 44,* 28–43.

Degani, A., & Kirlik, A. (1995). Modes in human-automation interaction: Initial observations about a modeling approach. In *Proceedings of the 1995 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 3443–3450). Piscataway, NJ: Institute of Electrical & Electronics Engineers.

Degani, A., Shafto, M., & Kirlik, A. (1999). Modes in human-machine systems: Review, classification, and application. *International Journal of Aviation Psychology, 9,* 125–138.

Frank, P. M. (1990). Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy – A survey and some new results. *Automatica, 26,* 459–474.

Hirschhorn, L. (1984). *Beyond mechanization: Work and technology in a postindustrial age.* Cambridge, MA: MIT Press.

Holder, B., & Hutchins, E. (2001). What pilots learn about autoflight while flying on the line. In *Proceedings of the 11th International Symposium on Aviation Psychology* (n.p.). Columbus: Ohio State University.

Jamieson, G. A., & Guerlain, S. (2000). Operator interaction with model-based predictive controllers in petrochemical refining. In *Proceedings of the Human Performance, Situation Awareness and Automation Conference* (pp. 172–177). Marietta, GA: SA Technologies.

Lee, J. D., & Sanquist, T. F. (1996). Maritime automation. In R. Parasuraman & M. Mouloua (Eds.), *Automation and human performance: Theory and applications* (pp. 365–384). Mahwah, NJ: Erlbaum.

Leveson, N. G., & Palmer, E. (1997). Designing automation to reduce operator errors. In *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1144–1150). Piscataway, NJ: Institute of Electrical & Electronics Engineers.

Leveson, N. G., Pinnel, L. D., Sandays, S. D., Koga, S., & Reese, J. D. (1997). Analyzing software specifications for mode confusion potential. In *Proceedings of the Workshop on Human Error and Systems Development* (pp. 132–147). Glasgow, Scotland: University of Glasgow.

Moray, N. (1986). Monitoring behavior and supervisory control. In K. R. Boff, L. Kaufman, & J. P. Thomas (Eds.), *Handbook of perception and human performance* (Vol. 2, 40.1–40.51). New York: Wiley.

Mouloua, M., & Parasuraman, R. (Eds.). (1994). *Human performance in automated systems: Current research and trends.* Hillsdale, NJ: Erlbaum.

Mumaw, R. J., Roth, E. M., Vicente, K. J., & Burns, C. M. (2000). There is more to monitoring a nuclear power plant than meets the eye. *Human Factors, 42,* 36–55.

Nikolic, M. I., & Sarter, N. B. (2001). Peripheral visual feedback: A powerful means of supporting effective attention allocation in event-driven, data-rich environments. *Human Factors, 43,* 30–38.

Norman, D. A. (1988). *The psychology of everyday things.* New York: Basic.

Norman, D. A. (1990). The "problem" with automation: Inappropriate feedback and interaction, not "over-automation." *Philosophical Transactions of the Royal Society of London (B), 327,* 585–593.

Oishi, M., Tomlin, C., & Degani, A. (2003). *Discrete abstractions of hybrid systems: Verification of safety and application to user-interface design* (NASA/TM-2003-212803). Moffett Field, CA: National Aeronautics and Space Administration, Ames Research Center.

Palmer, M. T., & Abbot, K. H. (1994). *Effects of expected-value information and display format on recognition of aircraft subsystem abnormalities* (NASA Tech. Report No. 3395). Hampton, VA: National Aeronautics and Space Administration, Langley Research Center.

Parasuraman, R., Molloy, R., Mouloua, R., & Hilburn, B. (1996). Monitoring of automated systems. In R. Parasuraman & M. Mouloua (Eds.), *Automation and human performance: Theory and applications* (pp. 3–17). Mahwah, NJ: Erlbaum.

Parasuraman, R., & Mouloua, M. (Eds.). (1996). *Automation and human performance: Theory and applications.* Mahwah, NJ: Erlbaum.

Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, and abuse. *Human Factors, 39,* 230–253.
Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics, 30,* 286–297.
Rasmussen, J. (1976). Outlines of a hybrid model of the process plant operator. In T. B. Sheridan & G. Johannsen (Eds.), *Monitoring behavior and supervisory control* (pp. 371–383). New York: Plenum.
Rasmussen, J. (1985). The role of hierarchical knowledge representation in decision making and system management. *IEEE Transactions of Systems, Man, and Cybernetics, SMC-15,* 234–243.
Rasmussen, J. (1986). *Information processing and human-machine interaction: An approach to cognitive engineering.* Amsterdam: North-Holland.
Rasmussen, J., & Goodstein, L. P. (1987). Decision support in supervisory control of high-risk industrial systems. *Automatica, 23,* 663–671.
Rasmussen, J., Pejtersen, A. M., & Goodstein, L. P. (1994). *Cognitive systems engineering.* New York: Wiley.
Riley, V. (1996). What avionics engineers should know about pilots and automation. In *Proceedings of the 14th AIAA/IEEE Digital Avionics Systems Conference* (pp. 252–257). Piscataway, NJ: Institute of Electrical & Electronics Engineers.
Riley, V. (2001). A new language for pilot interfaces. *Ergonomics in Design, 9*(2), 21–26.
Rouse, W. B. (1980). *Systems engineering models of human-machine interaction.* New York: Elsevier.
Sarter, N. B. (1995). "Knowing when to look where": Attention allocation on advanced automated flight decks. In R. S. Jensen (Ed.), *Proceedings of the Eighth International Symposium on Aviation Psychology* (pp. 239–242). Columbus: Ohio State University.
Sarter, N. B., & Woods, D. D. (1992a). Mode error in supervisory control of automated systems. In *Proceedings of the Human Factors Society 36th Annual Meeting* (pp. 26–29). Santa Monica, CA: Human Factors and Ergonomics Society.
Sarter, N. B., & Woods, D. D. (1992b). Pilot interaction with cockpit automation: Operational experiences with the flight management system. *International Journal of Aviation Psychology, 2,* 303–321.
Sarter, N. B., & Woods, D. D. (1994). Pilot interaction with cockpit automation: II. An experimental study of pilots' model and awareness of the flight management system. *International Journal of Aviation Psychology, 4,* 1–28.
Sarter, N. B., & Woods, D. D. (1995a). Autonomy, authority, and observability: Properties of advanced automation and their impact on human-machine coordination. In *Proceedings of the 6th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Man-Machine Systems* (pp. 181–184). Cambridge, MA: International Federation of Automated Control.
Sarter, N. B., & Woods, D. D. (1995b). How in the world did we ever get into that mode? Mode error and awareness in supervisory control. *Human Factors, 37,* 5–19.
Sarter, N. B., & Woods, D. D. (1997). Team play with a powerful and independent agent: Operational experiences and automation surprises on the Airbus A-320. *Human Factors, 39,* 553–569.
Sheridan, T. B. (1987). Supervisory control. In G. Salvendy (Ed.), *Handbook of human factors* (pp. 1243–1268). New York: Wiley.
Sheridan, T. B. (2002). *Humans and automation: Systems design and research issues.* New York: Wiley.
Sheridan, T. B., & Ferrell, W. R. (1974). *Man-machine systems: Information, control, and decision models of human performance.* Cambridge, MA: MIT Press.
Sklar, A. E., & Sarter, N. B. (1999). Good vibrations: Tactile feedback in support of attention allocation and human-automation coordination in event-driven domains. *Human Factors, 41,* 543–552.
Vakil, S. S., & Hansman, R. J. (2002). Approaches to mitigating complexity-driven issues in commercial autoflight systems. *Reliability Engineering and System Safety, 75,* 133–145.
Vakil, S. S., Hansman, R. J., Midkiff, A. H., & Vaneck, T. (1995). Feedback mechanisms to improve mode awareness in advanced autoflight systems. In *Proceedings of the Eighth International Symposium on Aviation Psychology* (pp. 243–248). Columbus: Ohio State University.
Vicente, K. J. (1999). *Cognitive work analysis: Toward safe, productive, and healthy computer-based work.* Mahwah, NJ: Erlbaum.
Vicente, K. J., Christoffersen, K., & Hunter, C. N. (1996). Response to Maddox critique. *Human Factors, 38,* 546–549.
Vicente, K. J., Mumaw, R. J., & Roth, E. M. (2004). Operator monitoring in a complex dynamic work environment: A qualitative cognitive model based on field observations. *Theoretical Issues in Ergonomics Science, 5,* 359–384.
Vicente, K. J., & Rasmussen, J. (1990). The ecology of human-machine systems: II. Mediating "direct perception" in complex work domains. *Ecological Psychology, 2,* 207–250.
Vicente, K. J., & Rasmussen, J. (1992). Ecological interface design: Theoretical foundations. *IEEE Transactions on Systems, Man, and Cybernetics, 22,* 589–606.
Vicente, K. J., Roth, E. M., & Mumaw, R. J. (2001). How do operators monitor a complex, dynamic work domain? The impact of control room technology. *International Journal of Human-Computer Studies, 54,* 831–856.
Wade, H. L. (1994). *Regulatory and advanced regulatory control.* Research Triangle Park, NC: Instrument Society of America.
Wickens, C. D., Mavor, A. S., Parasuraman, R., & McGee, J. P. (Eds.). (1998). *The future of air traffic control: Human operators and automation.* Washington, DC: National Academy.
Wiener, E. L. (1985). Beyond the sterile cockpit. *Human Factors, 27,* 75-90.
Wiener, E. L., & Curry, R. E. (1980). Flight-deck automation: Promises and problems. *Ergonomics, 23,* 995–1011.
Woods, D. D. (1991). The cognitive engineering of problem representations. In J. Alty & G. Weir (Eds.), *Human-computer interaction in complex systems* (pp. 169–188). London: Academic.
Woods, D. D. (1996). Decomposing automation: Apparent simplicity, real complexity. In R. Parasuraman & M. Mouloua (Eds.), *Automation and human performance: Theory and applications* (pp. 3–17). Mahwah, NJ: Erlbaum.
Woods, D. D., Patterson, E. S., Roth, E. M., & Christoffersen, K. (1999). Can we ever escape from data overload? A cognitive systems diagnosis. In *Proceedings of the Human Factors and Ergonomics Society 43rd Annual Meeting* (pp. 174–178). Santa Monica, CA: Human Factors and Ergonomics Society.

Greg A. Jamieson is an assistant professor of mechanical and industrial engineering at the University of Toronto, where he is also codirector of the Cognitive Engineering Laboratory. He received a Ph.D. in mechanical and industrial engineering in 2003 from the University of Toronto.

Kim J. Vicente is a professor of mechanical and industrial engineering, biomaterials and biomedical engineering, computer science, and electrical and computer engineering and is also founding director of the Cognitive Engineering Laboratory at the University of Toronto. He received a Ph.D. in mechanical engineering in 1991 from the University of Illinois at Urbana-Champaign.

Date received: September 12, 2001
Date accepted: August 17, 2004