International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 1, Number 1 (2011), pp. 17-30

http://www. ripublication.com

Describing and Probing Complex System Behavior: A Graphical Approach

D.S. Rao

Professor, Chitkara University, HIMUDA Education Hub,
Plot No. 3 & 4, Atal Nagar, Barotiwala, Distt. Solan - 174 103, India.
E-mail: dr.dsrao@yahoo.in, subba.rao@chitkarauniversity.edu.in

Abstract

Hands-on training and operation is generally considered the primary means that a user of a complex system will use to build a mental model of how that system works. However, accidents abound where a major contributing factor was user disorientation/misorientation with respect to the automation behavior, even when the operator was a seasoned user. This paper presents a compact graphical method that can be used to describe system operation, where the system may be composed of interacting automation and/or human entities. The fundamental goal of the model is to capture and present critical interactive aspects of a complex system in an integrated, intuitive fashion. This graphical approach is applied to an actual military helicopter system, using the onboard hydraulic leak detection/isolation system as a testbed. The helicopter Flight Manual is used to construct the system model, whose components include: logical structure (waiting and checking states, transitional events and conditions), human/automation cross communication (messages, information sources), and automation action and associated action limits. Using this model, examples of the following types of mode confusion are identified in the military helicopter case study: 1) Unintended side effects, 2) Indirect mode transitions, 3) Inconsistent behavior, 4) Ambiguous interfaces, and 5) Lack of appropriate feedback. The model also facilitates analysis and revision of emergency procedures, which is demonstrated using an actual set of procedures.

Introduction

One of the hopes placed in automation during its early years was emancipation – automation's emancipation from the human. The last three decades are awash with [8]. An uneasy co-existence between software and wet-ware has reluctantly been

accepted by most commercial designers, but the usual practice is to assign as many aspects of system operation to software and fill the remaining gaps with a human. This "software-centered" design created a new breed of accidents characterized by breakdowns in the interaction between operator and machine. Swinging in the opposite direction to remedy this has been "human-centered" design, where emphasis can be placed on artificial constraints that might arise from a user's naïve mental model (i.e., fool-proofing) or from a designer's model of the "one best way" [11]. Another emerging perspective treats human variability as a source of stability within an adaptive system instead of as erroneous behavior. Flach et. al [4] has termed this approach "use-centered" design, where it is assumed the human will naturally adapt to the functional constraints if those constraints are visible.

A key goal of the Software Engineering Research Lab create a methodology that will support integrated design of the automation and human tasks in complex, safety-critical systems. Such a methodology will not only address unsafe and problematic system features, but will be able to do so early in the design process when changes can still be made relatively easily. The methodology will be based on formal modeling, simulation, and analysis techniques starting with a user model of the system and generating appropriate and safe software and task models. The modeling tools should assist engineers and human factors experts in enhancing situation awareness, minimizing human errors such as those related to mode confusion, enhancing learnability, and simplifying the training of humans to interact with the automation.

A first step in achieving these goals is to determine how to use modeling and analysis to detect or prevent automation features that can create mode confusion. Three types of models are used: a *user model*, an *operator task model*, and a detailed specification of the *blackbox automation* behavior [7]. In this paper we describe the user model, which has shown to be helpful in detecting system features that can lead to mode confusion. This model appears to hold promise for use-centered design both as an analysis tool and as an onboard display concept. A specific case study employing the user model on an actual hydraulic leak detection/isolation system is described. The goals of the case study were to show scalability and efficacy of the approach for complex systems.

Background

Identified six categories of system design features that can contribute to mode confusion errors: ambiguous interfaces, inconsistent system behavior, indirect mode transitions, lack of appropriate feedback, operator authority limits, and unintended side effects [9]. One result of a case study by Leveson and Palmer [10] was a recognition that mode confusion errors could only be identified if the software (automation) model was augmented by a simple model of the controller's view of the software's behavior (a user's model) - the formal software specification was not enough.

The work of Rodriguez et. al [12] investigated the utility of comparing user and pilot task models for detecting potential mode confusion in a MD-11 Flight

Management System (FMS) case study. Building on this work, Bachelder and Leveson [1] found that the analyst's "situational awareness" of human/machine interplay improved if key aspects of the operator model were incorporated in the user model, thus producing a hybrid of the two. In this way accuracy, speed and focus are enhanced – comparing individual elements of two complex, structurally dissimilar model tends to be difficult and distracting.

Degani [3] developed a task-modeling framework, known as OFAN, which is based on the Statecharts language. Our experience in using Statecharts on real systems found it to be inadequate for our goals. Therefore, we have designed a blackbox automation requirements specification and modeling language call SpecTRM, which includes specification of modes and which we have found scales to large and complex systems [7]. The SpecTRM toolset is based on a methodology that supports human problem solving and enhances the safety and quality of systems, such as those that integrate human decision-making and automated information gathering. The SpecTRM tool set uses an approach for describing system specifications known as the Intent Specification.

Intent specifications are based on fundamental ideas in system theory and cognitive engineering. An intent specification not only records information about the system, but also provides specifications that support human problem solving and the tasks that humans must perform in system and software development and evolution. There are seven levels in an intent specification, each level supporting a different type of reasoning about the system. The information at each level includes emergent information about the level below and represents a different model of the same system. Figure 1 shows the overall structure.

Javaux uses a finite state machine to describe a cognitive mental model, which he uses to identify potential instances of mode confusion [5, 6]. We do not try to model human cognition or human mental models. Instead we model the blackbox behavior of the automation that the user expects and depends upon to perform the required steps needed to complete a given task. Modeling the actions involved in an operator task potentially allows analysis of the operator interaction along with a formal model of the rest of the system.

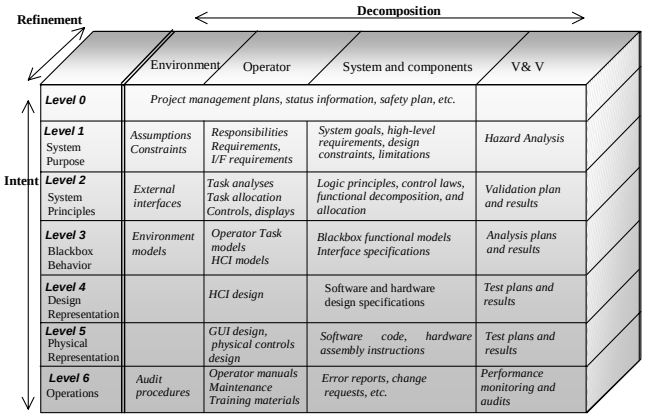


Figure 1 - Components of an intent specification.

In his paper "Designing to Support Adaptation," Rasmussen [11] states that an information system design should have content that faithfully represents the functional structure of the system, its operational state, and the boundaries of acceptable system operation. Many of these elements are contained in the model presented here, so that the user model conscripted for mode confusion analysis may actually offer itself as a valuable training and operator aid.

Approach

A controller (automatic, human, or joint control) of a complex system must have a model of the general behavior of the controlled process. Feedback via sensors to the controller serves to update the model so that it can remain consistent with the actual process being controlled. When a human shares control with automation, the distinction between automation and the controlled process can become difficult to perceive (or irrelevant) from the user's perspective. If an operator's mental model diverges from the actual state of the controlled process/automation suite, erroneous control commands based on that incorrect model can lead to an accident [8]. Mismatches between model and process can occur when: a) The model does not adequately reflect the behavior of the controlled system, b) Feedback about the state of the modeled system is incorrect.

In order to specify and validate these models, a user model that incorporates elements of a human task model is used. For an existing system, this model can be extracted from the operator's manual and other operator documentation and training materials for the given system. Ideally, the model would have preceded the built system so that the tasks, detailed automation specifications, and training and operator manuals will have been written from the user model.

The components of the graphical language, shown in Figure 2, refine on the set developed in [12] so as to better reflect information and process flow, as well as reduce diagram clutter. States (represented by square boxes) are steps required to complete a task, which in this study consist simply of checking variables and waiting for changes to occur. A transition is defined as the process of changing from one state to another and is represented by an arrow. Conditions that trigger transitions are called events, and an action (denoted by text with a gray rounded rectangle) describes a result or output from the transition.

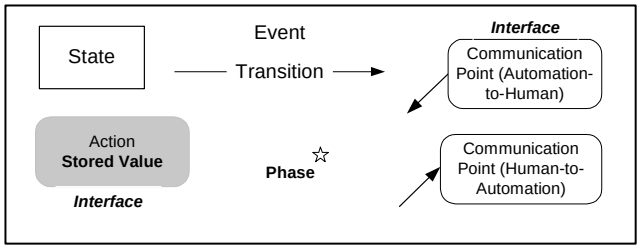


Figure 2. Components of user modeling language.

Values and parameters associated with automation action that are pre-determined (stored) appear in bold, and the sources (interfaces) where these values and parameters are found are indicated above or below the action ovals in italics. A communication point links different actors together. Rounded rectangles with down-arrows denote automation-to-human communication points, and italics above the communication point indicate the interface where that communication appears to the human. Similarly, up-arrows indicate communication from the human to the automation. Finally, a superscripted star indicates phase of automation or operation.

Case study of a helicopter hydraulic leak detection system

In order to test the user model, a case study was performed on the leak detection system of an actual military helicopter. The leak detection system was selected for analysis because this system is perhaps one of the least understood by pilots (based on the operational experience of one of the authors). Figure 3 shows the user model that was created with the helicopter's Pilot Flight Manual. It should be noted that this model does not necessarily reflect the aircraft's actual system operation; rather it is a graphical interpretation of the textual guides. Discrepancies or potential problems that are indicated by the model may be due to Flight Manual inaccuracies (which is a real-world problem), or reflect actual system problems. The authors' interpretation of the manuals is (however small) also a degree of freedom to be considered. When constructing such a model, it is important the paths that the design intended to occur are captured compactly and clearly. The extent to which this is accomplished largely determines its utility as an analysis tool. Numerous iterations of crosschecking manuals with model are generally required before the model stabilizes at its final form. This extensive time investment coupled with the uncertainty of manual accuracy are yet more reasons arguing for pre-design analysis emphasis, versus post-design.

System Description

Figure 3 shows the main components of the case helicopter hydraulic system: three hydraulic pump modules, two transfer modules, dual-set redundant primary servos (three servos each set), dual-set redundant tail rotor servos, and a pilot-assist module having a stability augmentation system (SAS) servo, Boost servo, and a Trim servo. The back-up pump provides redundancy by supplying hydraulic power to both No. 1 and No. 2 systems if one or both pumps fail. During nominal operation, the No. 1 hydraulic pump drives the first-stage tail rotor servo as well as the No. 1 transfer module, which in turn powers the first-stage primary servos of the main rotor. The first-stage tail rotor servo can be manually turned off by flipping the TAIL SERVO switch to BKUP. The No. 2 hydraulic pump drives the second-stage primary servos and the pilot-assist servos. Manual switches can individually turn off the pilot-assist servos. When the SVO OFF switch is moved to either the 1^{ST} off or 2^{nd} stage position, that stage of the primary servos is turned off (depressurized), but the two cannot be turned off at the same time. The back-up pump supplies emergency pressure to the No. 1 and/or No. 2 transfer modules whenever a pressure loss in them occurs. It also supplies pressure to the No. 2 stage of the tail rotor servo in case of: 1)

a pressure loss in the first stage of the tail rotor servo, or 2) low fluid level in the No. 1 system ("#1 RSVR LOW" message on the caution panel). A detailed schematic of the helicopter hydraulic system taken from the Flight Manual is shown in Figure A1 in the Appendix.

The hydraulic Leak-Detection/Isolation (LDI) system receives inputs from pressure switches, fluid level switches, and control switch positions to monitor the operation of the hydraulic systems. The user model in Figure 4 shows the sequence of actions and cueing performed by the LDI when a low fluid level is detected in the No. 1 hydraulic pump, provided that the pilot executes emergency procedures as directed by the Flight Manual. The acronyms *CP* and *AP* that appear above the communication points denote Caution Panel and Advisory Panel, respectively. The LDI assumes that the leak is in the #1 tail rotor servo, the back-up pump is engaged, and the #1 tail rotor servo is turned off as the #2 tail rotor servo is activated. If the leak stops, the #2 tail rotor is left in operation. If the leak continues (the leak could be in the transfer module, upstream from it, or in the first stage of the primary servos) and all fluid is lost from the #1 system, the #1 tail rotor servo automatically resumes operation (#2 turned off) when the back-up pump supplies pressure to the #1 transfer module. The emergency procedures then require that the pilot switch off the #1 primary servos, so that only the #2 primary servos (pressurized by the #2 pump) are powered. The #1 tail rotor servo continues to receive power from the back-up pump through the #1 transfer module. If the pilot does not shutdown the #1 primary servos, but the leak is actually occurring upstream of the #1 transfer module, the leak will cease. Otherwise, eventually the back-up pump will lose all pressure and the #1 primary servos, in addition to the #1 tail rotor servo, will stop functioning and result in loss of flight control. The LDI logic yields a similar sequence of events with the #2 hydraulic system (user model shown in Figure 5), except that the pilot-assist module is taken off-line when a leak is detected. As the pilot-assist module is not normally needed for safe operation of the helicopter (as opposed to the tail rotor), there is not a redundant set of pilot-assist servos. If the leak continues, the back-up pump activates and provides power to the #2 transfer module, and the pilot-assist module resumes operation. Emergency procedures dictate that the pilot then switch off the #2 primary servos.

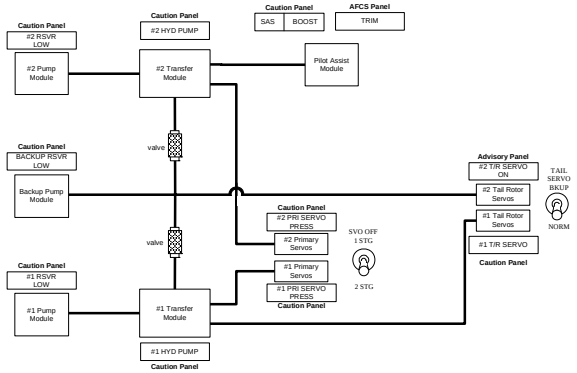


Figure 3. Simplified representation of case helicopter hydraulic system.

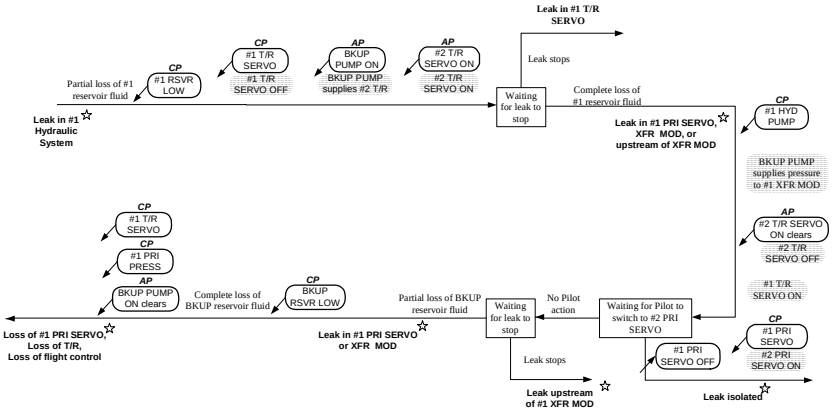


Figure 4. User model for #1 hydraulic system leak procedures.

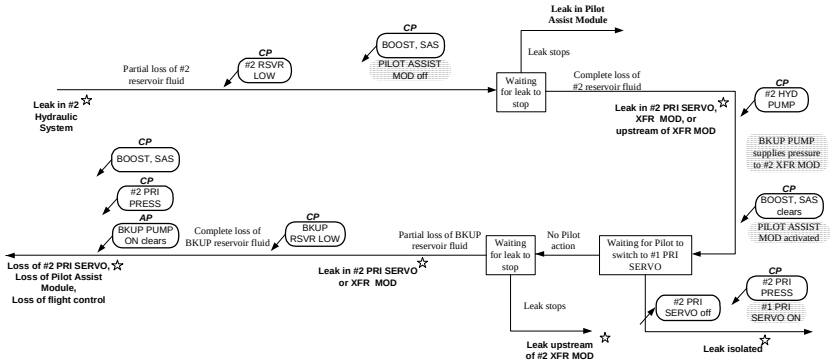


Figure 5. User model for #2 hydraulic system leak procedures.

Mode Confusion

Five of the six previously cited system features that can lead to mode confusion were found in this model and are presented in the following sections. The five features are: indirect mode changes, inconsistent system behavior, ambiguous interfaces, lack of appropriate feedback, and unintended side effects.

Indirect Mode Change

An indirect mode change occurs whenever there is a change in mode by the automation without explicit command from the operator. An especially useful feature of the user model is the ease with which indirect mode changes are recognized: they occur when shaded action ovals that are not preceded by an up-arrow communication point (i.e., pilot-directed). In Figure 4 there are six such instances during the evolution of a 'nominal' emergency: 1) the #1 tail rotor servo deactivated, 2) the back-up pump engaging, 3) the #2 tail rotor servo activating, 4) the #2 tail rotor servo deactivating, 5) the backup pump supplying pressure to the #1 transfer module, and 6)

the #1 tail rotor servo is activated. Indirect mode changes are not necessarily indicative of potentially hazardous pilot/machine interaction, but in conjunction with other system features, such as lack of appropriate feedback, indirect mode changes can become significant factors in system safety.

Referring to the *Leak isolated* phase in Figure 4, if a problem later develops with the selected #2 primary servo system, the Pilot Manual states that the #1 primary servos will automatically reactivate *if the backup system is not required to drive the #1 primary servos.* Barring this condition, the pilot must manually make the switch between servo systems. An indirect mode change can thus occur with the primary servos (rather important hardware) under certain – though by no means obvious – circumstances. The user model of this critical system feature is shown in Figure 6. Whether or not the backup pump is required for a given set of primary servos to function may demand a convoluted answer, especially if the emergency departs from 'textbook' expectations. In addition, as pilots are rarely presented with simulator scenarios that deviate from those addressed in the Flight Manual's emergency procedures, the backup pump status-automatic servo switchover nuance can generally be assured a short half-life in a pilot's memory. This issue of *hidden* mode change now introduces the next section.

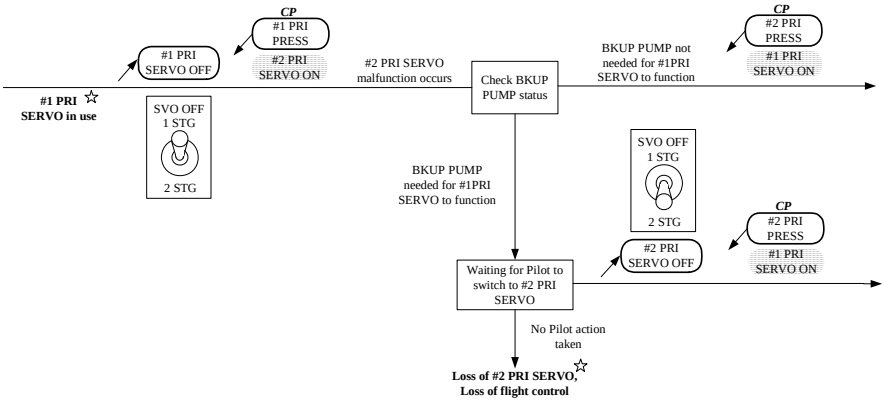


Figure 6. Scenario demonstrating LDI check of backup pump status.

Lack of Appropriate Feedback

Consider the following scenario where an automatic switch from the #2 primary servos to the #1 primary servos has occurred, as shown in Figure 6. Prior to this transition the pilot's manual servo switch is in the "1st STG OFF" position (the switch positions are shown for clarity in the figure). Following the automatic switchover the manual switch remains in "1st STG OFF," but the cue received by the pilot on the caution panel is "#2 PRI PRESS," indicating that the #2 primary servos are inoperative. To the unsuspecting pilot this would be highly suggestive of a near-term disaster – the primary servos are apparently not responding to the manual switch. Depending on the pilot, his/her response may range from cycling the servo switch to possibly pulling servo circuit breakers. At a minimum, the operator's confidence in

the system will have been deeply compromised. The two instances of inappropriate feedback in this scenario are: 1) No positive feedback indicating that an automatic transition has occurred between servos, and 2) The pilot's servo switch position and the automation cue reflecting its function are precisely in conflict.

Inconsistent System Behavior

Carroll and Olson define a consistent design as one where a similar task or goal is associated with similar or identical actions [3]. An example of inconsistent system behavior highlighted by the model in Figure 4 concerns the tail rotor servos. There is an advisory light for when the #2 tail rotor servo is operating (but not when the #1 servo operates), and a caution light associated with loss of pressure in the #1 tail rotor servo (but not when the #2 servo loses pressure). The cues are entirely skewed toward incorrect operation of the #1 tail rotor servo, which makes incorrect operation of the #2 tail rotor servo difficult and "unnatural" to detect. As will later be shown, knowledge of the #2 tail rotor servo status could be critical to safety of flight.

Looking at Figure 3, the manual switch controlling the tail rotor servos selects the servo intended for *operation*. Curiously, the manual switch controlling the primary servos selects the servo intended for *de-activation*. While training can engrain instinctive switch responses to canned emergencies, problems may occur when knowledge-based behavior must be employed and the system state interpreted from mixed-sense switches. Doubt as to primary servo selection can be compounded when an automatic switchover between servos yields a disparity between system state and switch position, as was discussed in the previous section.

Ambiguous Interfaces

Interface mode errors can occur when the computer maps multiple conditions onto the same output and the operator interprets the interface erroneously. During backup pump operation the only positive cue offered is an advisory light indicating the backup pump is on. Looking at the system diagram in Figure 3 and the user model in Figure 4, the lack of positive feedback as to which components the backup pump is actually driving (#1 or #2 tail servos, #1 or #2 primary servos, pilot assist servos) raises ambiguity – the pilot must infer this information from indirect cues and system knowledge. Having to infer information that is required to anticipate or recognize an automatic switch between primary servos (i.e., Figure 5) is a strong indicator of potential mode confusion. Cockpit crew coordination among the pilot community is an item that receives continual attention and training. Automation's role – either an integrated crewmember or a silent ringmaster – is no less important and can decisively affect pilot initiative.

Unintended Side Effects

The helicopter emergency procedures and system design described in the Flight Manual depict emergencies as linear, ratchet-type events. Close inspection of Figures

3 and 4, however, lead to some interesting "flies in the ointment." When the LDI sequence has been triggered due to low-level fluid sensing in the #1 system, the #2 tail rotor servo replaces the #1 servo and is driven by the backup pump. If the leak is in the #1 transfer module, then fluid will continue to be lost from the #1 hydraulic system until depletion. At that point the backup pump supplies the #1 transfer module with pressure and the #1 tail rotor servo is brought back online. However, since the #1 tail rotor is receiving pressure from the backup pump via the leaking transfer module, eventually the backup system will be depleted of all fluid, and loss of tail rotor authority will follow – not good for the home team.

This scenario could be avoided if the pilot were to immediately switch to the #2 primary servos as soon as the #1 low-level caution light occurred. The LDI has already transferred operation to the #2 tail rotor servo, so that the #1 hydraulic system is effectively isolated from the backup pump system. The user model in Figure 7 reflects this change of emergency procedures, which should be compared to the Flight Manual's procedures in Figure 4. Regardless of where the leak actually occurs in the #1 system, this procedure will generally increase pilot options and reduce vulnerability. A similar procedural change is recommended for the #2 hydraulic system leak.

In another scenario illustrating unintended automation behavior, the #2 tail rotor servo has been brought into operation by the LDI because of a #1 primary servos leak (low-level sensing in the #1 hydraulic system). Suppose now that a leak occurs in the #2 tail rotor servo. As there is no caution light associated with the #2 tail rotor servo, the pilot would never know about a leak until the low- level caution light for the backup system appeared.

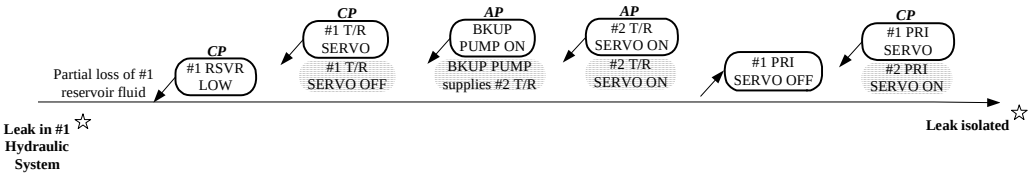


Figure 7. Proposed modification #1 hydraulic leak emergency procedure.

However, from the pilot's perspective, the leak could be occurring anywhere within in the backup system. Even if he/she were to guess correctly that the source of the leak was the #2 tail rotor servo, the LDI logic does not allow the #1 tail rotor to be re-selected for operation *until the #1 hydraulic system has lost all pressure*. One way around this Kafkaean scene might be for the pilot to: 1) Switch off the #1 primary servos, thus activating the #2 primary servos, and 2) Physically pull the #1 hydraulic pump circuit breaker, thus failing it. The LDI should then continue its programmed sequence and restore operation of the #1 tail rotor servo. This creative panic could be foregone by giving the pilot authority to choose either the #1 or the #2 tail rotor servo, assuming the pressure source for driving it was available.

For the final scenario, a leak has occurred in the pilot-assist servos and the LDI logic disconnects the pilot-assist module (see Figure 6). The pilot has preemptively selected the #1 primary servos. As it is night, the ship is a small deck (frigate), and there are high winds with rough seas, a shipboard recovery is only feasible with the boost-assist system. This system is part of the pilot-assist module that reduces high stick forces that the pilot must exert to move the primary servos. If the pilot attempts to reactivate the boost servo, the Flight Manual does not indicate whether the LDI would let this occur. If the boost system is allowed to activate, it will be drawing on the #2 hydraulic supply that is already low due to the initial leak. When this supply is completely lost, the backup pump would activate and pressurize the #2 transfer module (thereby continuing to drive both the pilot assist module and the #2 primary servos), but during the transition the boost power would be disconnected until it was manually reselected. An interruption in boost power during a shipboard landing would be bad, so rather than draw on the partially depleted supply of the #2 system it would be preferable to use the fully charged backup supply. In order to force the backup pump into operation and drive the pilot assist servos, the pilot would have to pull the circuit breaker to the #2 hydraulic pump – this is not standard operating procedure and definitely not an item practiced in the simulator.

As demonstrated by these examples, unintended automation behavior can manifest itself by leaving the operator helpless to perform a desired result, or else require that he/she "outwit" the automation by using unorthodox procedures.

Conclusion

When the user model was applied to an airliner's vertical descent guidance logic in [1], it was observed that the model not only enhanced detection of potential mode confusion features, but could also be useful as an operator display that showed current, previous, and anticipated system states. The case study in this paper emphasizes the fact that a user model can only reflect the *foreseen* modes of operation, be they normal or emergency modes. While it is not feasible to depict most of the possible event paths that could occur, a user model such as the one developed in this paper, in conjunction with a clear layout of the physical system (main components, interconnections, advisory and caution cues) can yield powerful insight into potential problems stemming from mode confusion. This user model clearly demonstrated that the helicopter's emergency procedures and advisory cues were exclusively tailored around the technology that had been installed to detect and isolate leaks, the LDI system. It is precisely at such "ramrod" design that Rasmussen [10] lowers his crosshairs: "A typical situation of this kind is facing plant operators and pilots, when rare operational conditions appear that the designers of the automatic control systems had not anticipated."

Wood states that "updating and calibrating our awareness of the *potential* paths (to failure) is essential for avoiding failures because we are only partially aware of these paths, and, since the world is constantly changing, the paths are changing. The effort to escape or avoid stale, limited views of the changing potential for failure is one portion of the process of building a safety culture" [13]. When humans play a role in

an automated process, they should be given the means to grasp the process in a way that stimulates the imagination. Referring to the operation of a work system, Rasumussen states that its "quantitative variables and the relational structure governing their interaction must be converted at the interface to a set of symbolic objects interacting through events in a virtual environment. The interface should therefore present a map of a symbolic landscape inhabited by objects – icons – representing states of processes, interacting mutually and with boundaries around territories of varying operational significance. This is important, not only to support the reasoning by an individual user, but also to give cooperating users an opportunity to point at and to discuss an external model" [11].

Presenting a user (onboard operations) with a model similar in concept to the one developed here, in conjunction with an iconic layout of the physical system, would satisfy most of the functional display requirements that Rasmussen cites. A method to enhance the perception of causality might employ lighting that portion of the model's path that is active, and by utilizing display persistence (varied as a function of path progression speed) a trail of decaying light would indicate path history. Based on vehicle-state trends and automation intentions, a future path could be indicated with a differently lit color. When a future conflict between automation intentions and the vehicle trends is predicted, the intended future path could flash and the variables of concern be displayed. This also helps to satisfy the requirement for representing the boundaries of acceptable system operation.

Finally, the user-physical model suite used in this study helped identify potentially serious shortcomings with a critical flight emergency procedure, and it facilitated a simple, powerful revision of that procedure.

References

[1] Bachelder, E. and Leveson, N., (In Press), *A Graphical Language for Describing Complex System Behavior: Applications to Design, Training and User Operation*, 20th Digital Avionics and Systems Conference, Oct 14-18, Daytona Beach, FL

[2] Carroll, J.M. and Olson, J.R., 1988, *Mental Models in Human-Computer Interaction*, in M.Helander (Ed). Handbook of Human-Computer Interaction, Elsevier Science Publishers, pp. 45-65.

[3] Degani, A., 1994, *Modeling Human-Machine Errors: on Modes, Errors and Patterns of Interaction*, Ph.D. Thesis, Atlanta, GA: Georgia Institute of Technology.

[4] Flach, J.M. & Dominguez, C.O., 1995, *Use-centered design*, Ergonomics in Design, July 19.

[5] Javaux, D., 1998, *An Algorithmic Method for Predicting Pilot-Mode Interaction Difficulties*, 17th Digital Avionics and Systems Conference, Oct 31, Bellevue, WA.

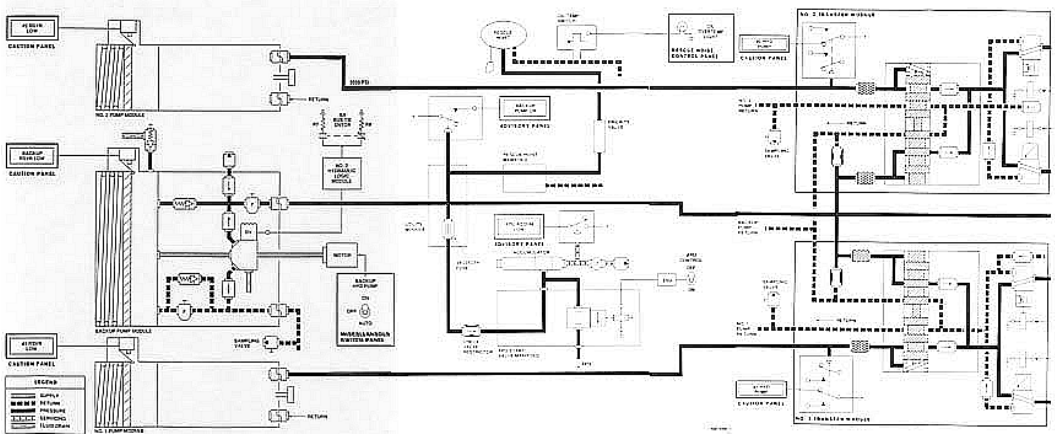
[6] Javaux, D. and De Keyser, V., 1998, *The cognitive complexity of Pilot-Mode Interaction*, in G. Boy & C. Graeber (eds.), Proceedings of the International

Conference on Human-Computer Interaction in Aeronautics, May 27, Montreal, Canada.
[7] Leveson, N., 2000, *Completeness in Formal Specification Language Design for Process Control Systems*, Proceedings of Formal Methods in Software Practice.
[8] Leveson, N., 1995, *Safeware: System Safety and Computers*, Addison-Wesley, New York.
[9] Leveson, N. 1997, *Analyzing Software Specifications for Mode Confusion Potential*, presented at the Workshop on Human Error and System Development, Glasgow, March.
[10] Leveson, N. and Palmer, E. (NASA Ames Research Center), 1997, *Designing Automation to Reduce Operator Errors*, in the Proceedings of Systems, Man, and Cybernetics Conference, October.
[11] Rasmussen, J., 1998, *Designing to Support Adaptation*, Hurecon, Denmark.
[12] Rodriguez, M. et. al, 2000, *Identifying Mode Confusion Potential in Software Design*, Digital Aviation Systems Conference, October.
[13] Woods, D. and Shattuck, L., (In Press) *Distant Supervision – Local Action Given the Potential for Surprise*, Cognition, Technology and Work.

Acronyms

BKUP: Backup
LDI: Leak Detection Isolation
PRI: Primary
RSVR: Reservoir
SAS: Stability Augmentation System
T/R: Tail Rotor
XFR MOD: Transfer Module

Appendix



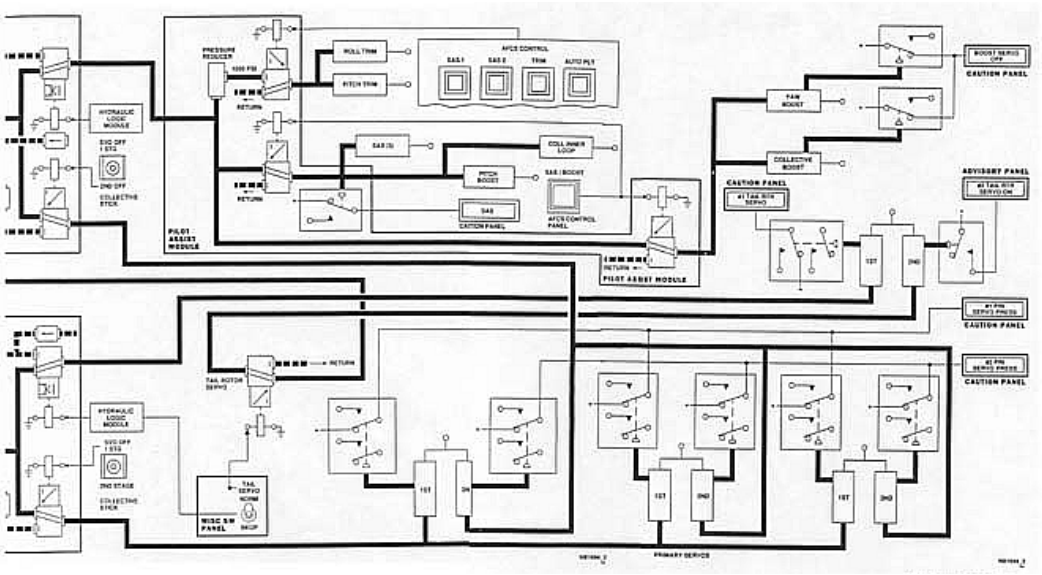


Figure A1. Case helicopter hydraulic system as represented in the Flight Manual.