Automatic Detection of Interaction Vulnerabilities in an Executable Specification

Michael Feary

NASA Ames Research Center
MS 262-4
Moffett Field, CA 94035, USA
michael.s.feary@nasa.gov

Abstract. This paper presents an approach to providing designers with the means to detect Human-Computer Interaction (HCI) vulnerabilities without requiring extensive HCI expertise. The goal of the approach is to provide timely, useful analysis results early in the design process, when modifications are less expensive. The twin challenges of providing timely and useful analysis results led to the development and evaluation of computational analyses, integrated into a software prototyping toolset. The toolset, referred to as the Automation Design and Evaluation Prototyping Toolset (ADEPT) was constructed to enable the rapid development of an executable specification for automation behavior and user interaction. The term executable specification refers to the concept of a testable prototype whose purpose is to support development of a more accurate and complete requirements specification.

Keywords: automation design, automation surprise analysis.

1 Introduction

This paper presents an approach to providing designers with the means to detect Human-Computer Interaction (HCI) vulnerabilities without requiring extensive HCI expertise. The goal of the approach is to provide timely, useful analysis results early in the design process, when modifications are less expensive. The approach consists of the development of computational Human-Automation Interaction (HAI) vulnerability analyses, integrated into a software prototyping toolset. The toolset, referred to as the Automation Design and Evaluation Prototyping Toolset (ADEPT) was constructed to enable the rapid development of an executable specification for automation behavior and user interaction. The term executable specification refers to the concept of a testable prototype whose purpose is to support development of a more accurate and complete requirements specification.

The paper specifically focuses on the evaluation of the HAI vulnerability analyses' operational validity. Operational validity refers to the effectiveness of the analyses in predicting actual difficulties in operation. The evaluation was performed by comparing vulnerability predictions from the tool to performance of participants operating two device prototypes created in ADEPT.

D. Harris (Ed.): Engin. Psychol. and Cog. Ergonomics, HCII 2007, LNAI 4562, pp. 487–496, 2007.

The metrics for determining whether or not an identified automation vulnerability causes difficulty in performance were defined as (1) failures in completing realistic tasks within a given amount of time, and (2) difficulty in predicting future automation behavior. The vulnerability analyses were designed to identify vulnerabilities known to cause difficulty in operation [1,2,3,4].

The vulnerability analyses include:

- Moded Inputs
- Armed Behaviors
- Automatic Behaviors
- Inhibited Inputs
- Similar Feedback

Moded Inputs Analysis

Moded Inputs are user interface objects (e.g., buttons, knobs, etc.) that may result in different device behaviors when an action is taken upon them (e.g., pressed, rotated, etc.). Moded Inputs may make it difficult to predict future automation behavior. A moded input is defined formally in ADEPT as a 'user action which, depending on the mode of the device, can lead to more than one device behavior'.

An example of a Moded Input in the study is a multi-function remote control that can control a television or a video recorder. Depending on the "mode" of the remote control, the power button on the remote control will turn the television or recorder on or off.

The vulnerability associated with moded inputs is most commonly referred to as "mode confusion" [3,5,6,7]. This difficulty is compounded if sufficient feedback is not provided.

Armed Behavior Analysis

Armed Behaviors are device behaviors that require more than just a user interface action for engagement. Armed behaviors are particularly troublesome because a delay usually exists between the user interface action and the time at which all of the conditions for engagement are satisfied.

An example from aviation is the Korean Airlines 007. Although the true cause of this accident may never be known, a probable explanation for this accident is that the crew had "armed" the inertial navigation system, but it did not meet all the necessary conditions for engagement. The crew did not notice that the aircraft was proceeding off-course when they transgressed Russian airspace and were shot down [8].

Automatic Behavior Analysis

Automatic Behaviors are device behaviors that engage independent of user input. Automatic Behaviors are similar to armed behaviors from a HAI standpoint. The difference between the two is that armed behaviors require a user interface action to initiate the automation behavior, while automatic behaviors do not. When coupled

with inadequate feedback, these behaviors have been referred to as "strong and silent" [4].

An example of an automatic behavior is airspeed envelope protection in certain aircraft that automatically engages and increases engine thrust if the airspeed drops below a specific airspeed threshold.

Inhibited Inputs Analysis

An *Inhibited Input* vulnerability is a user interface input action that does not result in a behavior change. Similar to Moded Inputs, Inhibited Inputs make it difficult to predict what the device will do after a user action. Although the vulnerabilities related to inhibited inputs are indirect, they greatly affect the user's understanding of automation behavior. As such, it is difficult to find incident and accident reports for which inhibited inputs are a contributing factor, however it has been documented in controlled studies that inhibited inputs create confusion.

Examples of inhibited inputs in the video recorder example used in the study, occur while the recorder is in the record mode, which disables all buttons except for the stop and power buttons.

Vakil (1998, 2000) and Javaux (1998) have identified an inhibited input vulnerability as the source of the confusion that can occur when a flight crew attempts to engage a new mode while Approach Mode is active on some modern commercial aircraft.

Similar Feedback Analysis

A *Similar Feedback* vulnerability is present when the same interface objects are used to display the information content for more than one device behavior. In addition to being identified as a vulnerability [9,10]. Similar Feedback vulnerabilities compound other automation surprise vulnerabilities. Feary et. al (1998) examined experienced pilots' knowledge of aircraft automation behaviors, and demonstrated a lack of pilot knowledge resulting from similar annunciations. The pilots in that study also showed significant improvements in automation behavior prediction when they were given feedback that matched the autopilot behavior.

In the aircraft involved in the Korean Airlines 007 accident, the display of the inertial navigation mode was the same whether the system was "armed" or "engaged". This made it difficult for the pilots to determine the actual automation behavior.

2 Method

The operational validity evaluation was conducted on two device prototypes. The first device—a video recorder remote control (Figure 1)—was chosen as an example of a "walk-up and use" device (i.e., a person familiar with the goals for using the device should not require special training).

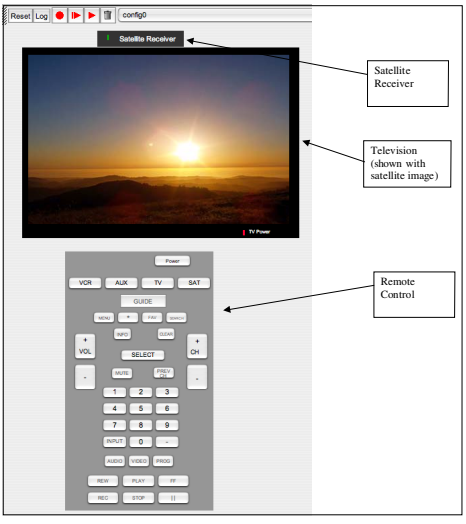

Fig. 1. Remote Control Prototype

The second device—an autopilot for a modern transport category aircraft (Figure 2)—was chosen as an example of a more complex device that does require specific training. The autopilot was chosen to present a challenge to the capabilities of the analyses: the design and training requirements for autopilots are very stringent due the safety critical nature of the commercial aviation.

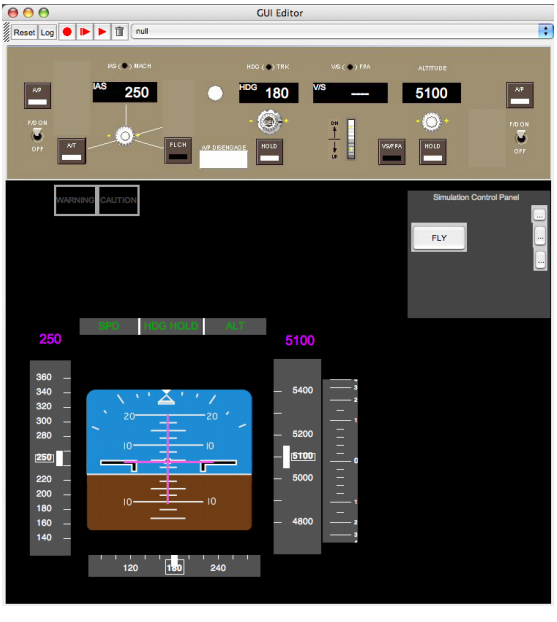

Fig. 2. The autopilot prototype

Participants
Eighteen pilots participated in the evaluation. The pilots ranged in age from 20–46, and each held at least a current private pilot's license. All pilots had a minimum of 200 hours of flight experience. The participants were also screened so as to have no formal Human-Automation Interaction training or expertise.

Procedure
The evaluation was conducted by analyzing the two devices in ADEPT, running domain expert participants through a representative series of tasks on the two device prototypes, and then comparing the results. The evaluation used two measures to evaluate the analyses' prediction ability: task performance, and automation behavior prediction performance.

An accurate prediction of task performance is the ultimate goal of the development of the automation vulnerability analyses, however it would be very optimistic to think that the automation surprise vulnerability analyses alone would be accurate predictors of task completion performance. There are many variables that affect the ability of a user to complete a task and only a small number of those variables are considered in the analyses.

To focus the evaluation of the analyses on automation surprise vulnerabilities, an additional measure referred to as automation behavior prediction performance was also evaluated. Performance on this measure was obtained by asking the participants to select the next automation behavior from a list of potential future behaviors.

Data from the tasks was collected in two ways. First, the experimenter collected task performance data on an evaluation form. The same experimenter (the author) collected data for all 15 participants. A more detailed backup of the data was collected through a data logging facility built into the prototype. Before each task, the experimenter activated the data logging facility, which kept a record of the participants' actions, automation behaviors and state values, and times.

3 Results

The results of the evaluation showed that the analyses predictions did not show significant correlation to the users' performance on the tasks, the predictions were a significantly correlated to the participants' ability to determine future automation behavior.. As described earlier this result can be explained by the influence of additional complexity in predictions of task performance, including the perceptual quality of the interface feedback, the familiarity with the task, and the ability to use heuristics (i.e. process of elimination).

Task Completion Analysis Results
Task completion performance was determined by the ability of the participant to complete the each task within thirty seconds, although none of the participants failed any task because to time limit constraints. The tasks were ranked in order of predicted difficulty, defined as the number if identified vulnerabilities present. Figure 3 shows the results of the participants' task completion performance compared with the task difficulty predicted by the analyses.

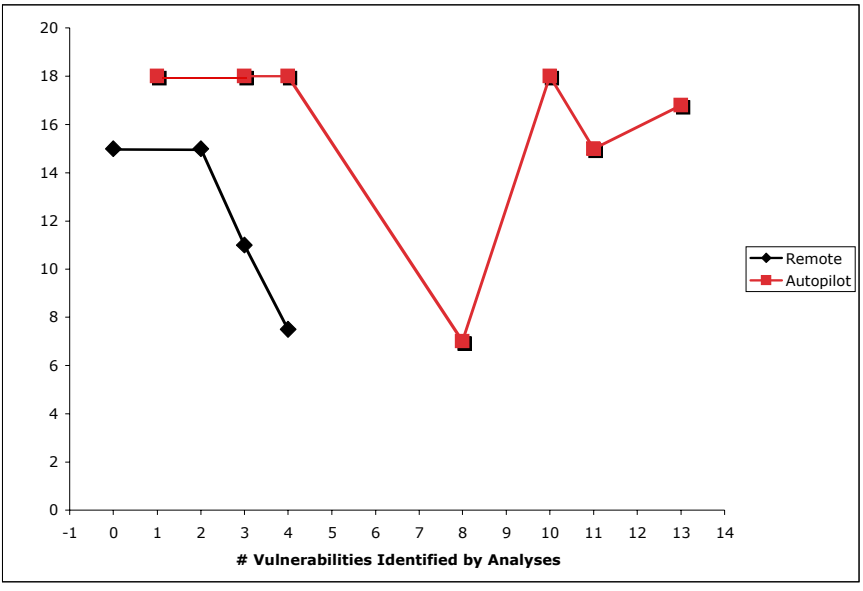

Fig. 3. Analysis Task Completion Results (note that two tasks contained 10 vulnerabilities, and five tasks contained 13 vulnerabilities)

A spearman rank correlation coefficient was computed and showed that the predictions for the remote control tasks were significantly correlated with user performance (r_s=0.97, p<.02, n=15). Although the analyses' predictions for the autopilot tasks were not as accurate, the results show a weak trend of prediction versus user performance (r_s=0.53, p<.1, n=18).

Automation Behavior Prediction Analysis Results

There were only two remote control behavior prediction questions. Both of these were related to which device the (video recorder or television) would be affected by a button press on the remote control.

The participants were asked six autopilot behavior prediction questions. Two of the six autopilot automation behavior prediction questions asked were not used for the evaluation. The questions, with only one vulnerability each, were not counted in the analysis for two reasons: the experiment prototype was determined to not have enough fidelity for the participants to reasonably be expected to answer the questions correctly; and both questions were related to airspeed envelope protection. The automation behavior questions evaluation required that the simulation be paused immediately before the engagement of the automation behavior, and given the rapid onset of airspeed envelope protection, it was not always possible for the experimenter to pause the simulation before envelope protection engaged.

Three of the remaining four questions required responses in terms of the future behavior of the autopilot. The fourth question (three vulnerabilities) asked the participant to predict a mode change, which involved computing an altitude value.

This value was judged to be answered correctly if the participant responded within 100 feet of the correct value.

Similar to the results of the remote control, the analyses were an accurate predictor of which automation behavior prediction questions participants would have difficulty with. Figure 4 shows the results of the participants' performance in response to the automation behavior prediction questions.

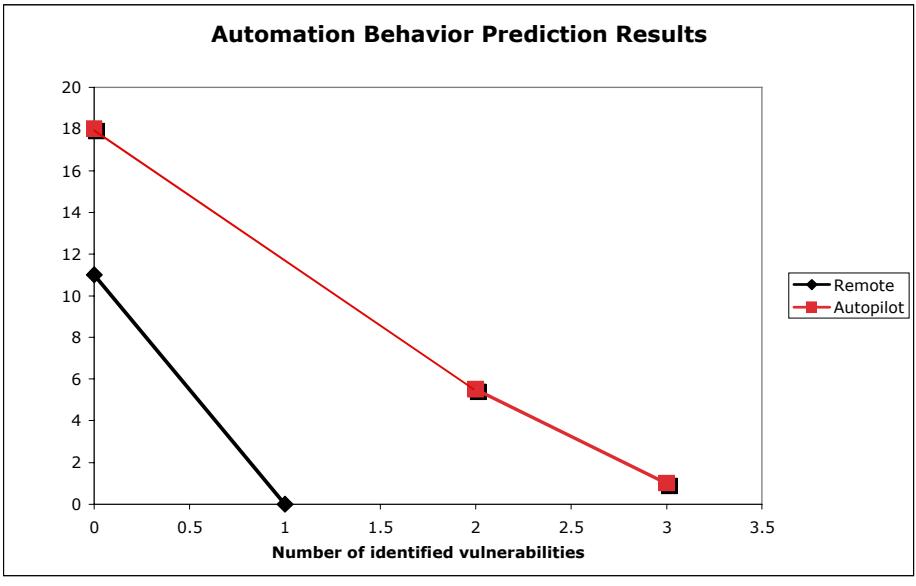

Fig. 4. Automation behavior prediction results (note that the two autopilot questions with two vulnerabilities each are marked by only one point)

Figure 4 shows that the analyses accurately predicted the difficulty of the remote control behavior questions. A spearman rank correlation coefficient was computed for the autopilot behavior questions and showed that the predictions were also correlated with user performance. (r_s=0.95, p<.05, n=18).

4 Discussion

Although the automation vulnerabilities identified by the analyses were poor predictors of task difficulty, the analyses were accurate predictors of the difficulty the participants faced when answering the automation behavior prediction questions. Like the results from the remote control evaluation, this is particularly encouraging, as this is the measure that focuses most upon potential Human-Automation Interaction difficulties.

It is also possible that the analyses' prediction of task difficulty may be better than it initially appears. A problem that became apparent during the experiment is that the button on the Lateral Target Selector knob is easily recognized when implemented in

a 3-dimensional hardware, but difficult to distinguish in the 2 dimensional software representation. As a result, even though it was described in the training, the participants incurred many extra actions looking for the lateral target selector knob on the first task.

There are a number of possible explanations for the analyses task performance results.

The most likely explanation for the inaccuracy of the analyses predictions is that the means of rating the vulnerabilities is immature. The analyses are not currently intended to provide ratings, however, ratings are needed evaluate the quality of the identified vulnerabilities The analyses are intended to be useful for detailed design work, and to identify areas that should be examined further. The results suggest that single vulnerabilities appear to be good predictors of difficulty versus no vulnerabilities, but are currently less useful for comparing two tasks with multiple vulnerabilities.

Closely related to this point, is the absence of weighting of the different vulnerabilities. It is extremely unlikely that all vulnerabilities would have the same effect on user performance. Prior to the study various methods were investigated for weighting the ratings of the vulnerabilities, however no empirical method for assessing the impact of the different vulnerabilities was identified. During the study it was observed that some vulnerabilities appeared to have a greater impact on task performance than others. Additionally, the results of the remote control study suggest the effect that differentiating the vulnerabilities could have improved the performance of the analyses predictions.

An example of the possible effects of weighting was alluded to earlier when describing the "Armed Behavior" vulnerability. The HAI vulnerability is affected by the length of time between user action and engagement (i.e. if the armed behavior engages immediately after the user action, there is not much of an HAI vulnerability). However, as described earlier, the analysis only identified the existence of an armed behavior, not the amount of time beween arming and angagement.

This may provide an explanation for the inaccuracy in the prediction the anomalous autopilot task (shown as the fourth autopilot task from the left in figure 4). This was an off-nominal task, and it is possible that off-nominal tasks and the associated behaviors may cause more difficulty than some of the other vulnerabilities. The remote control results suggest that the weighting may be important, as the analyses provided better results when identifying singular vulnerabilities, however further testing is needed.

Second, the analyses do not evaluate the "look and feel" of the interface. This absence is intentional, as the focus of these analyses is to examine what types of analyses can be formalized, and added as a supplement to traditional interface analysis, but has been proven to be a good indicator of usability difficulties. For example, the affordances work [13] and label following work [14] indicates that participants are likely to select interface objects which have labels which closely match the task description, even if they have been trained in the functions of the objects.

Third, the analyses do not account for sequence dependency or task context. The different patterns in data for the analysis of the automation behavior prediction questions and data for the analysis of the task steps seem to support this. This was

accounted for in the vulnerability point scoring by scoring all of the possible vulnerabilities regardless of sequence, however, the accuracy of the analysis predictions would likely increase if a more systematic method of analyzing the different possible sequences.

Fourth, the results for the prediction of automation behavior tasks suggest that the lack of evaluation of the monitoring behaviors may have impacted accuracy of the task predictions. The task decomposition accounts for monitoring behaviors and the vulnerabilities associated with the monitoring behaviors, however the experiments did not evaluate the effects of the vulnerabilities attached to the monitoring behaviors for the operational tasks evaluated. In contrast, the vulnerabilities associated with the monitoring behaviors could be the focus of an automation behavior prediction question, as illustrated by the first question in the autopilot evaluation which asked the participant to predict the when an armed behavior would engage.

Fifth, the autopilot results may have also been disrupted/exaggerated by a lack of aural and vestibular feedback. Participants who initially made in incorrect action but corrected the action before completing the task were scored as completing the task. A good example of this is the anomalous autopilot task shown in figure 4 (fourth task from the left). It is possible that with additional aural and vestibular feedback, the participants would have corrected their actions. Similar to the fourth point, the results of the automation behavior prediction questions support this, as the directed questions focus on automation knowledge rather than feedback, and monitoring skills.

Sixth, unlike the remote control, the autopilot is a device that is expected to require some amount of training to use. Training is provided specifically to mitigate the errors caused by a complex environment, a complex device, and/or a complex interface. Since the analyses make predictions based on the complexity of the device and interface, training may mitigate the impact of the predictions, and quality or comprehensiveness of the training may lead to differences in performance on certain tasks. For example a majority of training using autopilots for transport category aircraft is spent developing skills for responding to emergency or abnormal situations. As such the certain functions of the autopilot will receive more practice than others [11,15,16].

Seventh, it is possible multiple vulnerabilities may interact with each other to impact the predictions. For example, the autopilot contained inhibited behaviors, which, may impact the understanding of the user, and affect the way the user interacts with the device [11,15,16]. In fact the only autopilot input used by the participants that was not inhibited at some time during the evaluation was the Vertical Speed button, which would engage whenever the participant pressed the button. However it would not always engage the participant's desired mode. Again, the automation behavior prediction question data supports this by focusing on automation understanding, whereas the educated guesses by the participant may obscure the level of automation understanding when using task data alone.

Although the analyses predictions do not appear to accurately predict the difficulty of tasks for the autopilot for the reasons discussed above, the analyses did appear to accurately predict the difficulty of questions about the automation, and this is a significant finding towards the formalization of usability metrics from an automation behavior based perspective, compared to existing interface based usability techniques.

References

1. Billings, C.: Human-Centered Aviation Automation: Principles and Guidelines. In: NASA Technical Memorandum 110381, NASA Ames Research Center, Moffett Field, CA, USA (1996)
2. Bureau of Air Safety Investigation, Advanced Technology Aircraft Safety Survey Report, Bureau of Air Safety Investigation, Civic Square ACT, Australia (1998)
3. Federal Aviation Administration, The interface between flightcrews and modern flight deck system. FAA human factors team. Federal Aviation Administration. Washington, DC (1996), Available at http://www.faa.gov/avr/afs/interface.pdf
4. Sarter, N., Woods, D.D.: Strong, Silent, and 'Out of the Loop'. CSEL Report 95-TR-01, Columbus, OH, USA (February 1995)
5. Degani, A., Shafto, M., Kirlik, A.: Modes in Human-Machine Systems: Review, Classification, and Application. International Journal of Aviation Psychology 9(2), 125–138 (1999)
6. Sherry, L., Feary, M., Polson, P., Mumaw, R., Palmer, E.: A Cognitive Engineering Analysis of the Vertical Navigation (VNAV) Function. In: NASA Technical Memorandum 2001-210915, NASA Ames Research Center, Moffett Field, CA, USA (2001)
7. Thimbleby, H.: User interface Design. Addison-Wesley, Wokingham, England (1990)
8. Degani, A.: Taming HAL: Designing Interfaces Beyond 2001. Palgrave MacMillan, Hampshire, England (2004)
9. Vakil, S., Hansman, R.J.: Predictability as a Metric of Automation Complexity, In: Human Factors & Ergonomics Society 41st Annual Meeting, pp. 70–74 (September 1997)
10. Vakil, S.: Analysis of Complexity Evolution Management and Human Performance Issues in Commercial Aircraft Automation Systems. PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA. (2000)
11. Javaux, D.: Explaining Sarter and Woods' Classical Results. The Cognitive Complexity of Pilot-Autopilot Interaction on the Boeing 737-EFIS, In: the Proceedings of the 3rd Workshop on Human Error, Safety, and System Development (HESSD'98), Liege, Belgium, pp. 62–77 (June 7-8, 1998). Poller, M.F., Garter, S.K.: The effects of modes on text editing by experienced editor users, Human Factors, 26(4), 449–462 (1984)
12. Feary, M., McCrobie, D., Alkin, M., Sherry, L., Polson, P., Palmer, E., McQuinn, N.: Aiding Vertical Guidance Understanding. In: NASA Technical Memorandum NASA/TM-1998-112217, Ames Research Center, Moffett Field, CA (1998)
13. Norman, D.A.: Cognitive Engineering. In: User Centered System Design. Norman, D.A., Draper, S.W. (eds.) Lawrence Erlbaum Associates, Hillsdale, NJ, USA, pp. 31–61 (1986) Originally: Steps Towards a Cognitive Engineering. Technical Report, Program in Cognitive Science, University of California, San Diego (1981)
14. Polson, P.G., Lewis, C.H.: Theory-based design for easily learned interfaces. Human-Computer Interaction 5, 191–220 (1990)
15. Feary, M., Sherry, L., Polson, P., Fennel, K.: Incorporating Cognitive Usability into Software Design Processes. In: Harris, D., Duffy, V., Smith, M., Stephandis, C. (eds.) Human-Centered Computing: Cognitive, Social, and Ergonomic Aspects, vol. 3, pp. 427–431. Lawrence Erlbaum, Mahweh, NJ (2003)
16. Sherry, L., Fennell, K., Feary, M., Polson, P.: A Human-Computer Interaction Analysis of Flight Management System Messages, NASA/TM-2005-213459. Ames Research Center, Moffett Field, CA, USA (2005)