AUTOMATION FOR TASK ANALYSIS OF NEXT GENERATION AIR TRAFFIC MANAGEMENT SYSTEMS

Maricel Medina, BSc, Lance Sherry, PhD
Center for Air Transportation System Research
George Mason University
Virginia, USA
mmedinam@gmu.edu, lsherry@gmu.edu

Michael Feary
NASA Ames Research Center
Moffet Field, CA, USA
Michael.feary@nasa.gov

***Abstract*—The increasing span of control of Air Traffic Control enterprise automation (e.g. Flight Schedule Monitor, Departure Flow Management), along with lean-processes and pay-for-performance business models, has placed increased emphasis on operator training time and error rates. There are two traditional approaches to the design of Human-Computer Interaction (HCI) to minimize training time and reduce error rates: (1) experimental user testing provides the most accurate assessment of training time and error rates, but occurs too late in the development cycle and is cost prohibitive, (2) manual review methods (e.g. cognitive walkthrough) can be used earlier in the development cycle, but suffer from poor accuracy and poor inter-rater reliability. Recent development of "affordable" human performance models provide the basis for the automation of task analysis and HCI design to obtain low cost, accurate, estimates of training time and error rates early in the development cycle.**

This paper describes a usability/HCI analysis tool that this intended for use by design engineers in the course of their software engineering duties. The tool computes estimates of trials-to-mastery (i.e. time to competence for training) and the probability of failure-to-complete for each task. The HCI required to complete a task on the automation under development is entered into the web-based tool via a form. Assessments of the salience of visual cues to prompt operator actions for the proposed design are used to compute training time and error rates. The web-based tool enables designers in multiple locations to review and contribute to the design. An example analysis is provided along with a discussion of the limitations of the tool and directions for future research.

Human Computer Interaction, Usability Analysis, Task Analysis, Probability of Faiure to Complete a Task, Trials to Mastrery.

I. INTRODUCTION

The evolution of the air transportation system to a "mature" industrial sector has resulted in cost differentiation as a primary means of competitive advantage for airlines. This cost imperative has flowed through the supply chain to aircraft manufacturers and Air Traffic Control. The result has been new business models (e.g. low cost carriers, outsourcing) and incentives for the supply chain vendors to reduce installation costs and operational costs (e.g. training, operational efficiency, and safety). Air Navigation Service Providers ANSPs) have embraced this challenge by privatization of Air Traffic Control, pay-for-performance, and the development of large-scale enterprise management and control automation such as Flight Schedule Monitor (FSM), Departure Flow Management (DFM), Surface Management Systems (SMS).

Human Computer Interaction has emerged as one of the ways to reduce costs by streamlining training as well as increase the efficiency of operators. For example, Boeing Commercial Aircraft Group funded a large internal R&D project with the specific design goal of reducing training costs and improving flight deck operational efficiency (Mumaw, Boorman, and Prada, 2006, Castor-Peck, personal communication). Several avionics vendors (Faerber, Vogl, and Hartley, 2007; Jacobsen, Chen, and Widemann, 1999), airlines (Fennell, Sherry, and Roberts, 2006), and NASA's Exploration Mission Directorate, Human Research Program (NASA, 2008) also have similar initiatives in place.

The most accurate evaluation of the usability of a product is achieved through experimental user testing (Nielsen, 1993). This type of approach is cost prohibitive and can only be conducted at the end of the development cycle when the cost of revisions is highest.

This paper describes a tool based on the Human Computer Interaction Process Analysis model (HCIPA) that this intended for use by software and design engineers in the course of their software engineering duties, to conduct usability analyses. HCIPA attempts to solve two very hard problems in the design of advanced automated systems. The first is capturing the details of operator-system interactions while performing a large number of mission tasks, task analysis. The sequence of operator actions and inferred mental operators for each task is then used to solve the second problem, making useful predictions of time to complete a task, repetitions required to master a task, and the likelihood of failure for failure infrequently performed tasks. This paper presents a web based tool that solves the first problem. Cog Tool (John, et al., 2004) is able to make accurate performance predictions for frequently performed tasks. We are in the process of developing related

ISBN: 978-0-615-20720-9

methods for predicting repetitions and likelihood of failure. Preliminary versions of models for making these predictions are reported in this paper.

Specifically, the tool enables designers and testers to describe the sequence of operator actions, and rapidly assess the trials to mastery (i.e. time to competence for training) and the probability of failure-to-complete for each task that can be performed by the product under design. The computation of these human performance measures is based on the specification of operator actions and an assessment of the salience of visual cues in the proposed automation user-interface to prompt the next operator action. The web-based tool also provides designers in multiple locations to view and contribute to the design and the usability evaluation.

This paper is organized as follows: Section 2 provides an overview of Human Computer Interaction (HCI) and introduces the Human Computer Interaction Process Analysis (HCIPA) method. Section 3 describes the tasks that can be performed by the functions of the tool. Section 4 provides case studies of usability analysis conducted with the tool. Section 5 discusses the limitations of the tool and directions for future research.

II. OVERVIEW OF HUMAN COMPUTER INTERACTION

Human-computer interaction involves the cognitive, motor, and visual activities of an operator using automation to perform a mission task (Card, Moran & Newell, 1983). The interaction between operator and automation follows a human action cycle of goal formulation, execution, and evaluation (e.g. Norman, 1988). The degree to which the content of the user-interface matches the "semantic space" of the operator determines the usability of the automation (Kitajima, Blackmon, and Polson, 2002).

Several techniques have been used to determine the usability of automation (Nielsen, 1992). The most accurate evaluation of the usability of a product is achieved through experimental user testing. Human subjects perform a list of tasks using the automation under test while observers take notes or record the operator's behavior. The aim is to identify problems on the product or features that users like and are easy to use. Techniques include "think aloud protocols" and eye tracking. Although quantitative data can be collected by measuring time to learn, speed of performance, and rate of human error; this approach is cost prohibitive and can only be conducted at the end of the development cycle when the cost of revisions is highest (Nielsen, 1994).

Alternative approaches that can be used earlier in the life-cycle, fall into two categories: Manual Inspections and Operator Performance Predictions. Manual inspections, such as participatory design (Muller and Kuhn, 1993), cognitive walkthroughs (Wharton, Rieman, Lewis, and Polson, 1994), heuristic evaluations (Nielsen, 1992), and other forms of expert reviews, have been shown to be effective in certain settings (Dumas, 2003) but are subjective and can be biased by group-thinking (Turner & Pratkanis, 1998). These methods also exhibit poor inter-rater reliability (Hetrzum & Jacobsen, 2003) due to differences in granularities of the task definition and the differences in the subjective ratings.

Automated tools, such as CogTool (John, Prevas, Salvucci, and Koedinger, 2004), seek to eliminate these two sources of poor inter-rater reliability by capturing actual end-user button pushes (to eliminate ambiguity in the task definition), and by estimating performance using human performance models such as Keystroke-Level Model (KLM), (Luo & John, 2005). These tools can also be used early in the development cycle.

CogTool, one of the first tools of this class, provides an easy way model skilled users' performance behavior through storyboards designs. To create the storyboards, CogTool users include the different screen shots on the tool and specify "hot-spots" or widgets on the screen shots to simulate the user interaction. The screenshots are connected though transitions. Once the screens are connected, the user interacts on the screenshots through the widgets, and CogTool generates an executable script of the actions performed by the user that can be processed by an Operator Performance Model such as KLM (Luo & John, 2005), ACT-R (Anderson et al., 1995) or CORE (Vera, Howes, McCurdy, & Lewis, 2004) to compute a prediction of expert time-on-task.

A. The HCIPA Method

HCIPA is a manual task/usability analysis inspection method that was designed to address issues with usability in the aviation and space industries (Sherry et al., 2002, 2006). Specifically, these industries were interested in evaluating usability for trials-to-mastery and probability-to-complete the task.

The HCIPA method has its roots in a model of pilot cognition (Polson, Irving, & Irving, 1994). This method also known as the RAFIV model (Sherry et al., 2002) decomposes tasks into six sequential steps: (1) Identify Task, (2) Select Function, (3) Access Function, (4) Enter data for Function, (5) Confirm and Save Data, and (6) Monitor Function. These steps are illustrated in Figure 1.

The first step is to identify a task based on various external stimuli such as visual cues (menu item, error message), hearing cues (warning sounds), and a request (e.g. checklist) or by remembering (e.g. recall from long-term memory). Operator proficiency is reduced when the user interface does not provide any guidance by salient visual cues (Sherry, Fennell, Feary, Polson, 2006).

Once the user knows what to do, the next step is to decide the right function to accomplish the task, which is to select a function. The function may be the name of a screen, the label on a button, a prompt or any other characteristic that tells the user to initiate the task. The more accessible the function is to the user, the higher the probability is to accomplish the task.

A set of operator actions are performed by a user in order to accomplish the task through the selected function. These operator actions are grouped under Access, Enter, Confirm and Save, and Monitor step.

The Access Step encloses the operator actions needed to access the function on the device. The goal for a designer is to reduce the number of operator actions needed to access the function.

ISBN: 978-0-615-20720-9

The Enter Step encloses the operator actions needed to successfully execute the function. The operator actions may include data entry, visual data evaluation, and communication with external devices or personnel.

The Confirm and Save Step are all the operator actions needed to trigger the function.

Finally, the Monitor Step encloses the operator actions needed to monitor any change on the system state after the function is triggered.

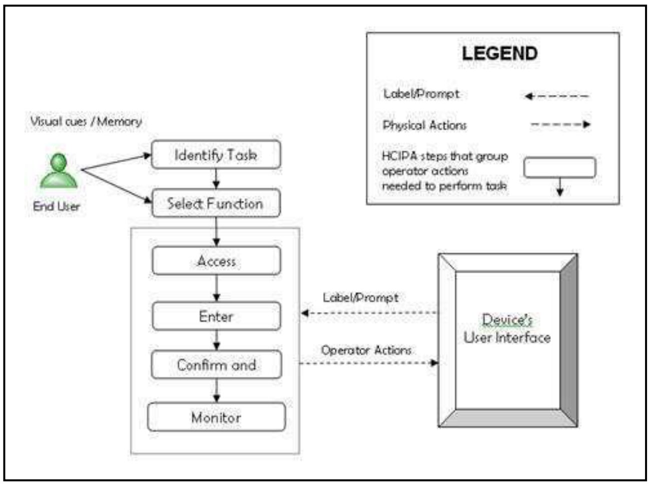

Figure 1. HCIPA Method

There are two basic classes of operator actions: (i) physical actions such as press a button or click on a link, and (ii) decision actions that cannot be viewed externally. A Task is executed by performing operator actions for each of the steps.

HCIPA estimates operator performance based on the minimization of memorized action sequences. When a user interface lacks clear labels, prompts, and/or organizational structure, additional training is required and operators must recall memorized action sequences (Sherry, Polson & Feary, 1998; Fennel, Sherry & Roberts, 2004).

The HCIPA approach has been successfully applied in several applications (Sherry, Polson & Feary, 2002**;** Sherry, Fennell, Feary, & Polson, 2006). The unguided manual process suffered from several issues: (1) ambiguity of granularity in descriptions of steps, (2) ambiguity is identification of salient visual cues, (3) problems in assessing salience of visual cues, (4) no method to determine trials-to-mastery or probability of failure to complete a task. The tool described in this paper is designed to overcome theses shortfalls and includes an affordable Operator Performance Model to compute trials-to-mastery and probability-to-complete the task.

III. THE E-HCIPA TOOL

e-HCIPA is a web based application developed to provide an automated way to apply the HCIPA method. The e-HCIPA is a free accessible web application; therefore, no username or password is required to use the tool. The current version of e-HCIPA runs only on Mozilla Firefox web browser and provides the following functionalities: Create a Task Analysis, Predict Operator Performance, Edit a Task Analysis, Delete a Task Analysis, and Generate PDF report (Task Analysis Report and User Guideline).

A. e-HCIPA Features

1) Create Task Analysis: Allows the user to create a new task analysis by inserting the device name, task name and function name. Once the device, name and function are saved, the labels for all steps are generated and the user can insert the operator actions for each step. Figure 2 shows the main screen of the tool

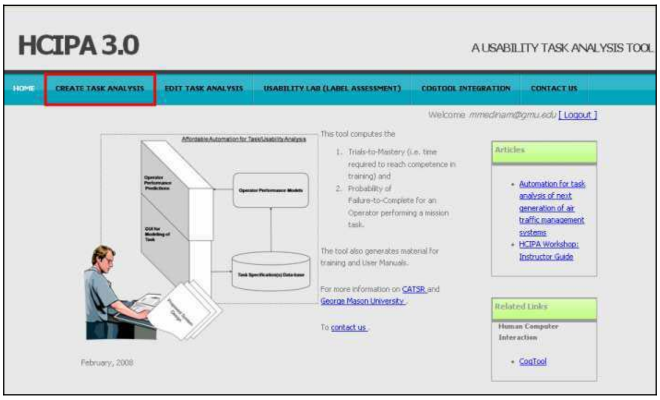

Figure 2. e-HCIPA Create Task Option

The operator actions may involve physical actions (press button, link), visual actions (read data from display field), audio actions (hear warning buzzer) or decision-making actions. Operator actions are automatically generated for the Identify Task and Select Function step based on the information entered on the task name and function name. The operator action for the Identify Task step is always generated as "*Recognize need to*:" concatenated with the task name entered by the analyst. The operator action generated for the Select function step is generated as "*Decide to use function*:" concatenated with the function name entered by the analyst. These two operator actions cannot be deleted by the user. The labels for the steps are created as follow:

- Identify Task Step: <*task name*>
- Select Function: <*function name*>
- Access Step: Access + <*function name*> + function
- Enter Step: Enter data for + <*function name*> + Function
- Confirm & Save Step: Confirm & Save data using + <*function name*> + Function
- Monitor Step: Monitor results of + <*function name*> + Function

The analyst can continue inserting operator actions for the Access, Enter, Confirm and Save and Monitor steps. Figure 3

ISBN: 978-0-615-20720-9

shows the screen where the operator actions are inserted and the salience assessment takes place.

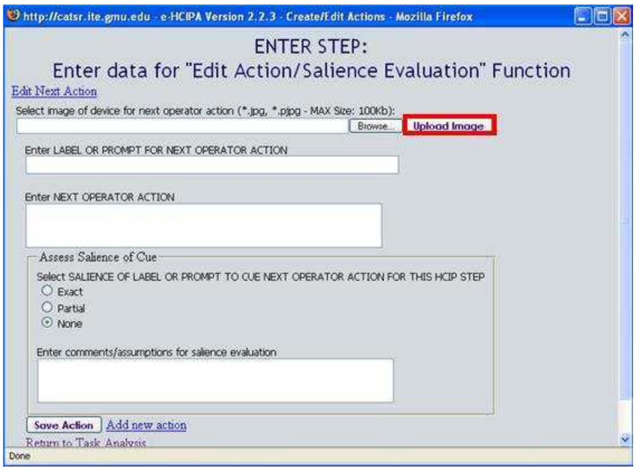

Figure 3. e-HCIPA Enter operator action

2) Predict Operator Performance: e-HCIPA calculates the two metrics based on the salience assessment conducted while inserting or editing an operator action: probability to fail a task, and trials to mastery the task. The probability of failure is calculated using (1), while the trials to mastery the task is obtained from (2). In (1), the maximum value used is 1. The values for the operator actions are calculated from the salience assesment using the following values: 0 for Exact, ¼ for Partial and 0 for None. Existing data support the prediction of trials to mastery a task (Bovair, Kieras, Polson, 1990).

$$\text{Probability to Failure} = 0.1753 * \sum \text{Operator actions} \quad (1)$$

$$\text{Trials Mastery Task} = 0.5916 * \sum \text{Operator actions} + 1.9632 \quad (2)$$

3) Edit a Task Analysis: e-HCIPA allows to modify any task analysis previously created. The device, task and function name can be changed at any time. If this is done, all the labels for the different steps will change also. The operator actions, including image, operator action description, label and salience assessment can be edited at any time. In order to edit a task analysis, the user must select the desired one from the list of task currently existing in the database.

4) Delete a Task Analysis: A task analysis can only be deleted by the person who created the task.

5) Duplicate a Task Analysis: A task analysis can also be duplicated. In this case, the system creates a new task with same content and images but it adds the *(Duplicate)* at the end of the Task Description. The person who duplicates the task becomes the creater of the new tasks.

6) Generate a PDF report: e-HCIPA allows to generate two .pdf reports. The Task Analysis Report contains all the operator actions grouped by step, including the trials to mastery and probability to complete the task, a thumbnail image, the label, the salience evaluation, and the salience comments. The User Guideline report contains all the operator actions inserted for the task and ordered sequencially. The User Guideline report can be used for training purposes.

B. e-HCIPA Technical Implementation

e-HCIPA has been developed using PHP 4.4.4 and MySQL database. Figure 4 shows the Entity-Relationship Diagram of e-HCIPA.

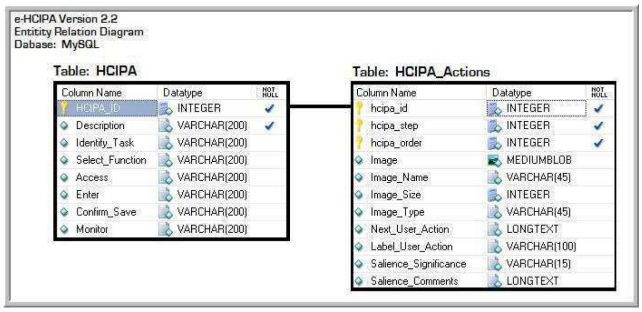

Figure 4. e-HCIPA Entity Relationship Diagram

The database table HCIPA stores the information for the device name, task description and function on fields Description, Identify_Task and Select_Function respectively. Once the user saves a new Task Analysis, e-HCIPA populates the rest of the fields on table HCIPA based on the information stored on the fields Identify_Task and Select_Function. Furthermore, two default operator actions are created: one for the Identify_Task step and the other one for Select_Function.

Table HCIPA_Actions stores all operator actions for the given task. The field hcipa_step is an enumerated field that keeps track of current step for the operator action. The values are: 1 for Identify_Task, 2 for Select_Function, 3 for Access, 4 for Enter, 5 for Confirm and Save and 6 for Monitor.

IV. CASE STUDY

An example HCIP analysis is illustrated below for and Air Traffic Management (ATM) System. The specific task is to run a ground delay program (GDP) at Chicago O'Hare Airport (ORD). Table I shows the input data used through HCIPA to analysis this task.

TABLE I. INPUT DATA FOR A HCIP ANALYSIS ON FMS737

Define Device, Task, and Function	
Device Name	Traffic Management System
Task Name	Run a [Ground Delay] Program at ORD" with General Parameters (Start/End Time/Duration, Arrival Fix, Aircraft Types, Carriers).
Function Name	GDT Setup: General.

ISBN: 978-0-615-20720-9

TABLE II. HCIP ANALYSIS FOR TASK "MODIFY DEPARTURE RUNWAY"

Step	Operator Action	Label	Salience Evaluation of Label to Cue Operator Action
Identify Task Run a [Ground Delay] Program at ORD" with General Parameters (Start/End Time/Duration, Arrival Fix, Aircraft Types, Carriers)	*Recognize need to: "Run a [Ground Delay] Program at ORD" with General Parameters (Start/End Time/Duration, Arrival Fix, Aircraft Types, Carriers)*	Bar Graph ORD Status	None
Select Function GDT Setup: General.	*Decide to use function: GDT Setup: General*	Tab labeled GDT Setup	Partial GDT stands for "Ground Delay Tool."
Access GDT Setup: General Function	Click on Tab labeled "GDT Setup"	Tab labeled GDT Setup	Exact Assume operator has domain knowledge to interpret menu items
Enter data for GDT Setup: General Function	Select "RBS++" on Program Type Pull-down Menu	Pull-down Menu: Program Type	Exact
	Set Start/End Time (type, move sliders, type duration)	General: Program Time: Start, End (or Duration)	Exact
	Select menu item "All" in pull-down menu "Arrival Fix:"	Pull-down Menu labeled	Exact
	Select menu item labeled "ALL" on pull-down menu labeled "Aircraft Types:"	Pull-down menu labeled "Aircraft Types:" menu item labeled "ALL"	None
	Type "ALL" into text field labeled "Carrier"	Text Field labeled	None
Confirm & Save data using GDT Setup: General Function	*None*		
Monitor results of GDT Setup: General Function	*None*		

An ATM specialist must be trained to read bar graph. There are scenarios when a GDP is not run even when the Hourly bars are in excess of the airport capacity (e.g. fog burn-off at SFO, pop-corn thunderstorms). ATM specialist requires significant training to define the parameters of the GDP. The Flight Schedule Monitor used to analyze this task offers no apriori decision-making support related to the parameters.

Table II shows all the operator action needed to complete this task.

The first column includes the HCIPA steps. The second column lists the operator actions. Note, that the operator actions of the Id Task and Select Functions steps are automatically generated by the tool. The third column lists the visual cue (if any) that prompts the next user action. The fourth column is an assessment of the salience of the cue.

 ISBN: 978-0-615-20720-9

Based on the salience evaluation and distribution of operator actions per HCIPA step, the estimated trials to mastery the task are 3.29, and the probability to fail the task is 0.39. Figure 5 shows the distribution of operator actions by HCIPA step and salience evaluation.

The operator needs to perform eight actions to complete this task. The most critical part is the first operator action. The current label is not obvious to perform the task through the selected function (salience evaluation is None). However, once the operator accesses the function, the visual cues are sufficient to complete the task.

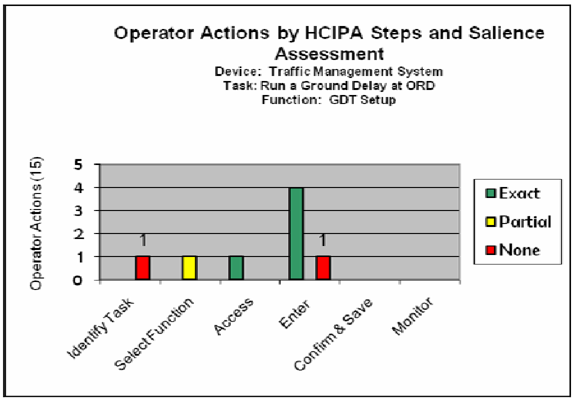

Figure 5. Operator Actions per HCIPA step

The use of HCIPA allows one to identify usability problems on the system for new ATM specialists. It also provides the benefit of generate a user guideline to train new operators on the analyzed task. The Appendix A shows the usability task report generated through HCIPA for this task.

V. Future Work

This paper describes a tool that this intended for use by software and design engineers in the course of their software engineering duties, to conduct usability analyses. Specifically, the tool enables designers and testers to rapidly assess the trials to mastery (i.e. time to competence for training) and the probability of failure-to-complete for each task that can be performed by the product under design. The computation of these human performance measures is based on the specification of operator actions and an assessment of the salience of visual cues in the proposed automation user-interface to prompt the next operator action. The web-based tool also provides designers in multiple locations to view and contribute to the design and the usability evaluation.

Beta testing of the tool is underway. Future work includes tool implementation, development of new functionalities, improvement of human performance model, and inter-rater reliability of the Assessment of the Salience of the Visual Cues.

- Tool implementation: The current version of tool has been tested on Mozilla Firefox and Internet Explorer. It is been developed using PHP 5.2. A security model has been implemented to allow certain functions to be accessed on the creator of the task. In terms of outputs, the current version only provides two reports on a .pdf format. These two reports will be also available in other format and, as needed; more reports will be developed, including the reports with graphs.
- New functionality: (i) hierarchical organization of task that allows to relate other tasks analysis as sub-task, (ii) provide API to enable import/export of models (e.g. with CogTool), (iii) development of a training laboratory by reusing task analysis description and images
- Operator Performance Model: The current model is based on empirical data from 4 experiments. Further work is planned to increase empirical data set and leverage existing models such as CORE, ACT-R, etc.
- Inter-rater Reliability of the Assessment of the Salience of the Visual Cues: The assessment of the salience of visual cues for prompting the operator's next action is critical for the accuracy of the tool. The current version of the tool relies on the assessment of the salience of the cue by the designer (i.e. None, Partial, Exact). This manual form of assessment suffers several issues. First, the assessment is reliant on the overlap of the designers "semantic state-space" with the end-users "semantic state-space." Recent studies have shown wide variance in semantic state-spaces and large differences between the semantic state-spaces of the designers and end-users. Second, even within a group of end-users and domain experts, the semantic state-space can exhibit a wide distribution. This issue will be investigated in two ways. First it is proposed to add a feature of the tool, loosely named, "Usability Lab." This feature will enable the collation of domain experts' assessment of the salience of the visual cues. Second, several automated techniques exist to automate the salience assessment. Latent Semantic Analysis, LSA (Landauer & Dumais, 1997; Kitajima, Blackmon, and Polson, 2000) and Scent-based Navigation and Information Foraging in the ACT architecture, SNIF-ACT (Pirolli & Fu, 2003) are two of these automated technique that will be researched to evaluate their feasibility to be included in e-HCIPA.

Acknowledgment

Thank you for technical assistance and suggestions from Peter Polson (University of Colorado), Mike Matessa (Alion Inc.), Karl Fennel (United Airlines). Thank you for support of the research to Steve Young (NASA), Amy Pritchett (NASA). This project was funded by grant from NASA – Aeronautics – Intelligent Integrated Flightdeck program, and by internal George Mason University Foundation Funds.

References

[1] The CogTool Project, Tools for Cognitive Performance Modeling for Interactive Devices. http://www.cs.cmu.edu/~bej/cogtool/.

[2] Anderson, J. R., John, B. E., Just, M. A., Carpenter, P. A., Kieras, D. E., & Meyer, D. E. (1995). Production system models of complex cognition.

 ISBN: 978-0-615-20720-9

In Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society (pp. 9-12). Hillsdale, NJ: Lawrence Erlbaum Associates

[3] Boorman, Daniel J. , Mumaw, Randall J. (2007) A New Autoflight/FMS Interface: Guiding Design Principles. Air Canada Pilots Association, Safety Conference.

[4] Bovair, S., Kieras, D.E., & Polson, P.G. (1990). The acquisition and performance of text editing skill: A cognitive complexity analysis. Human-Computer Interaction, 5, 1-48.

[5] Card, S., Moran, T. & Newell, A. (1983). The Psychology of Human-Computer Interaction. Hillsdale, NJ: Erlbaum.

[6] Castor-Peck, Steve (2007) Technical Memorandum to NASA Aeronautics – Intelligent Flight-deck Research Program. NASA-Ames Reseearch Center, Moffet Field,CA.

[7] Dumas, J. S. (2003) User-based evaluations. In J. Jacko and A. Sears (Eds.), The Human-Computer Interaction Handbook. (pp. 1093-1117) Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

[8] Faerber, R.A., Vogl, T.L. , & Hartley, D.E.. (2000) Advanced Graphical User-interface for Next Generation Flight Management Systems. In Proceedings HCI-Aero 2000. September 27-29, Toulouse, France. Pages 107-112.

[9] Fennell, Karl; Sherry, Lance; Roberts, & Ralph, Jr. (2004). Accessing FMS Functionality The Impact of Design on Learning NASA Technical Report (IH-051; NASA CR-2004-212837).

[10] Hertzum, Morten & Jacobsen, Niels Ebbe (2003): The Evaluator Effect: A Chilling Fact About Usability Evaluation Methods. In International Journal of Human-Computer Interaction, 15 (1) pp. 183-204

[11] Jacobsen, A. R., Chen, S. S., & Widemann, J. (1999) Vertical Situation Awareness Display. Boeing Commercial Airplane Group.

[12] John, B., Prevas, K., Salvucci, D. D., & Koedinger, K. (2004). Predictive human performance modeling made easy. In Human Factors in Computing Systems: CHI 2004 Conference Proceedings. New York: ACM Press.

[13] Kitajima, M., Blackmon, M. H., & Polson, P. (2000) G. A Comprehension-based model of Web navigation and its application to Web usability analysis. In People and Computers XIV, Springer, 357--373, 2000.

[14] Landauer, T. K. & Dumais, S. T. (1997) A solution to Plato’s problem: The Latent Semantic Analysis theory of acquisition, induction, and representation of knowledge. Psychological Review, 104, 211–240., 1997.

[15] Landauer, T. K., Foltz P. W. & Laham D. (1998). "Introduction to latent semantic analysis." Discourse Processes, 25, 259-284.

[16] Luo, L., & John, B. E, (2005) Predicting task execution time on handheld devices using the keystroke-level model, CHI '05 extended abstracts on Human factors in computing systems, April 02-07, 2005, Portland, OR, USA

[17] Muller, Michael J., & Kuhn, S. (1993) “Participatory Design” In: Communications of the ACM special issue on participatory design. 36(4), June 1993.

[18] Mumaw, R., Boorman, D. J., & Prada, R. L. (2006). Experimental Evaluation of a New Autoflight Interface. Proceedings HCI-Aero 2006, International Conference on Human Computer Interaction, Seattle, WA., September 2006.

[19] NASA (2008) Human Research Program. Exploration Missions Directorate. http://www.nasa.gov/directorates/esmd/acd/human.html

[20] Nielsen, J. (1992) “The usability engineering life cycle” In Computer Volume 25, Issue 3, Page(s):12 – 22, March 1992.

[21] Nielsen, J. (1994). “Guerilla HCI: Using discount usability engineering to penetrate the intimidation barrier," In: Randolph G. Bias and Deborah J. Mayhew (editors). Cost–justifying usability. Boston: Academic Press, pp. 242–272.

[22] Nielsen, J. & Landauer, T. K. (1993) A mathematical model of the finding of usability problems, Proceedings of the ACM INTERCHI'93 Conference (Amsterdam, the Netherlands, April 24-29), 206-213.

[23] Nielsen, J, & R.L. Mack (1994) Usability Inspection Methods. John Wiley & Sons, New York, NY, 1994.

[24] Norman, D.A. (1988) "The Design of Everyday Things." MIT Press

[25] Pirolli, P. & Fu, W.-T. (2003). SNIF-ACT: A Model of Information Foraging on the World Wide Web. In Proceedings of the Ninth International Conference on User Modeling, 2003

[26] Polson, P. G., Irving, S., & Irving, J. E. (1994). Final report: Applications of formal methods of human computer interaction to training and the use of the control and display unit. Washington, DC: System Technology Division, ARD 200, Department of Transportation, Federal Aviation Adrninistration.

[27] Sherry, L., Fennell, K., Feary, M., & Polson, P. (2006) Human–Computer Interaction Analysis of Flight Management System Messages. Journal of Aircraft, Vol 43, No. 5, September-October 2006.

[28] Sherry, L., Polson, P. & Feary, M. (2002) Designing User-Interfaces for the Cockpit: Five Common Design Errors, and How to Avoid Them. Paper to be presented at the 2002 SAE World Aviation Congress, Phoenix, AZ (November 5 – 7).

[29] Sherry, L., Polson, P., Feary, M. & Palmer, E. (2002) When Does the MCDU Interface Work Well? Paper presented at the the International Conference on Human-Computer Interaction in Aeronautics, Cambridge, MA (October 22-24).

[30] Turner, M. E., & Pratkanis, A. R. (1998). Twenty-five years of groupthink theory and research: Lessons from the evaluation of a theory. Organizational Behavior & Human Decision Processes, 73(2-3), 105-115.

[31] Vera, A., Howes, A., McCurdy, M., & Lewis, R.L.(2004). A constraint satisfaction approach to predicting skilled interactive performance. Proc. of the SIGCHI Conference on Human Factors in Computing Systems.

[32] Wharton, C., Rieman, J., Lewis, C., & Polson, P. (1994). The cognitive walkthrough method: A practitioner’s guide. In J. Nielsen & R. L. Mack (eds.). Usability inspection methods. New York, NY: John Wiley.

ISBN: 978-0-615-20720-9

APPENDIX A

Task Analysis Report

Flight Schedule Monitor - "Run a [Ground Delay] Program at ORD" with General Parameters (Start/End Time/Duration, Arrival Fix, Aircraft Types, Carriers)

Estimated TtM: 3.29

Estimated PFtC: 0.39

Device: Flight Schedule Monitor

Task: "Run a [Ground Delay] Program at ORD" with General Parameters (Start/End Time/Duration, Arrival Fix, Aircraft Types, Carriers)

Function: GDT Setup: General

Notes: Note 1: Air Traffic Specialist monitors Arrival Rates at an airport and decides that the predicted arrival rate in excess of the capacity for an extended period (e.g. 2 hours). This circumstance initiates the task to "Run a [Ground Delay] Program"

Note 2: General parameters is the 1st of 4 sets of parameters that must be entered.

1-IDENTIFY TASK:

"Run a [Ground Delay] Program at ORD" with General Parameters (Start/End Time/Duration, Arrival Fix, Aircraft Types, Carriers)

1.1 Recognize need to: "Run a [Ground Delay] Program at ORD" with General Parameters (Start/End Time/Duration, Arrival Fix, Aircraft Types, Carriers)

[Label]: Bar Graph ORD Status

[Salience of label/prompt to cue next operator action for this HCIP step]: None

[Comments/Assumptions]: Note 1: ATM Specialist must be trained to read bar graph. There are scenarios when a GDP is not run even when the Hourly bars are in excess of the airport capacity (e.g. fog burn-off at SFO, pop-corn thunderstorms).

Note 2: ATM Specialist requires significant training to define the parameters of the GDP. The tool offers no apriori decision-making support related to the parameters (only post entry with the Power Run)

Figure 6. Report generated on e-HCIPA for Task "Run a [Ground Delay] Program at ORD", Flight Schedule Monitor

ISBN: 978-0-615-20720-9