Cognitive Systems Engineering Educational Software (CSEES): Educational Software Addressing Quantitative Models of Performance

Matthew L. Bolton, Ellen J. Bass
Systems and Information Engineering
University of Virginia
Charlottesville, VA, USA
MatthewBolton@Virginia.edu, EllenBass@Virginia.edu

Abstract – *There is a lack of educational technology to support cognitive systems engineering topics such as models of human performance in dynamic environments. This paper describes the Cognitive Systems Engineering Educational Software (CSEES) system, an integrated toolset designed to facilitate curricula related to human judgment and decision-making performance modeling and evaluation. CSEES provides students with the means to generate and analyze performance data using multiple methods. It also includes documentation and tutorials. Its flexible design facilitates adding new judgment and decision making task environments. The paper describes the initial system implementation as a Microsoft Excel add-on, a preliminary evaluation as part of a graduate engineering course, and planned future work.*

Keywords: cognitive engineering, engineering education

1 Introduction

Cognitive systems engineering is a rapidly growing field encompassing a myriad of topics and techniques such as naturalistic study, cognitive modeling, cognitive task analysis, cognitive work analysis, ecological interface design, and modeling of human judgment and decision making in dynamic environments. To meet the need for researchers and practitioners, there are more than thirty engineering programs in the United States offering degrees specializing in the area and these programs continue to grow. Between 1998 and 2001, at least seven American universities hired faculty specializing in the area of cognitive engineering [6]. In the 2004-2005 academic year, at least 7 American and Canadian universities advertised for human factors candidates with one university (Embry-Riddle Aeronautical University) seeking 5 new faculty. Several of these positions called for cognitive engineers.

Education is a critical component of this growing field. Unfortunately instructors have a limited set of materials from which to choose for teaching purposes. We conducted an informal survey of the cognitive systems engineering and related programs in the United States by contacting representatives from the engineering programs listed on the Human Factors and Ergonomics Society's website (for a list of the eight responding programs, see http://cog.sys.virginia.edu/CSEES/SMC05/respond). Respondents provided information regarding curriculum, textbooks, and educational and related software. A limited number of textbooks are used by these programs – some of which are not specifically focused on cognitive systems engineering (Table 1). Instead of books, some instructors use course packets as the primary text for their cognitive engineering courses.

Table 1: Texts used in cognitive engineering courses

B.W. Niebel & A. Freivalds, *Methods, Standards and Work Design*, McGraw-Hill, 2002.

C.D. Wickens and J.G. Hollands, *Engineering Psychology and Human Performance*, Prentice Hall, 1999.

C.D. Wickens, S. Gordon, & Y.D. Liu, *An Introduction to Human Factors Engineering*, Prentice Hall, 2003.

D.A. Norman, *The Design of Everyday Things*, Basic Books, 2002.

J. Rasmussen, A.M. Pejtersen, & L.P. Goodstein, *Cognitive Systems Engineering*, Wiley-Interscience, 1994.

J. Rasmussen, *Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering*, Elsevier Science Ltd, 1986.

K.S. Gill, *Human Machine Symbiosis: The Foundations of Human-centered Systems Design*, Springer-Verlag Telos, 1996.

K.J. Vicente, *Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-Based Work*, LEA, 1999.

M.S. Sanders & E. J. McCormick, *Human Factors in Engineering and Design*, McGraw-Hill, 1993.

R.J. Jagacinski & J.M. Flach, *Control Theory for Humans: Quantitative Approaches to Modeling Performance*, LEA, 2002.

As the engineering education literature shows, instructors should engage in teaching practices such as establishing the relevance of course materials by providing realistic applications of the subject's theoretical constructs, balancing concrete and abstract information by using illustrations and demonstrations of course materials, grounding the material in experimentation, and promoting active learning [4]. With respect to educational technology, there are few resources for instructors to provide such learning opportunities. There are some freely available cognitive modeling tools such as SOAR (available at http://sourceforge.net/projects/soar/) and ACT-R (available at http://act-r.psy.cmu.edu/software/). Example models and tutorials are available for download. Commercial products are also available for academic institutions (for example, see CHI System's Inc.'s iGEN™: The Cognitive Agent Software Toolkit which has an academic version described at www.cognitiveagent.com/product_info_academic.htm). However, all of these modeling tools require many hours for students to learn how to build even the simplest models (for example, see http://act-r.psy.cmu.edu/tutorials/ as well as www.cognitiveagent.com/training.jsp for the suggested training for ACT-R and iGEN respectively).

For instructors wanting to survey different methods and therefore not wanting to devote many hours to any particular one, there are even fewer educational technology options. For that purpose, instructors try to incorporate off-the-shelf software including statistics and math packages, spreadsheets, and simulation packages. Some instructors have developed their own tools. For example, Dr. Bisantz at the University of Buffalo has developed a Java-based interactive visualization package to introduce Signal Detection Theory (see http://cog.sys.virginia.edu/CSEES/SDT/). However, in general, there is a lack of computational materials to support curricula focused on cognitive engineering topics.

At the University of Virginia (UVa), we are working toward the development of the Cognitive Systems Engineering Educational Software (CSEES) system. The concept is to promote inductive learning, experimentation, and active learning by supporting student participation in the generation, exploration, and analysis of human performance data. Our goal is to build the infrastructure for a flexible integrated educational system that provides tutorials and hands on experience with cognitive systems engineering methods. The system supports performance data generation and collection as well as analysis so that students can actively learn about the subject matter. The idea is that a student can learn about a methodology, collect simulated or actual performance data, and then use the system to analyze those data. With the integrated system, the student will not only be able to learn about a single methodology but also be able to model data using more than one technique in order to gain insight into the commonalities and differences between methods.

This paper describes the initial development and evaluation of CSEES for human performance measurement oriented curricula. The architecture is flexible so that new topics can be easily integrated into the system. Task environments simulate performance output or provide the ability for the student to perform tasks. The output data can be analyzed using multiple methods which allow students to make comparisons between the results. Integrated documentation, tutorials, and visualizations contribute to the data analyses and exploration by describing theoretical constructs while providing illustrations and demonstrations of concepts.

2 Education Software Specification

The CSEES software package is a Microsoft Excel add-on designed to enhance Excel by providing task environments, analysis methods, and help, all available through a specialized graphical user interface.

2.1 Task Environments

CSEES allows students to generate and collect their own data in order to enhance personal engagement as well as interpretation and understanding of the results. The purpose of a task environment is to allow performance data to be collected or generated and made available for analysis. In some cases, instructors may want to take advantage of applications already integrated into CSEES. In other cases, instructors may have other applications for the collection of performance data. CSEES facilitates this in two ways. Firstly, each task environment is run as a separate process activated dynamically through the graphical user interface. A new process can be added to the list of available task environments by placing an executable or shortcut in a folder in the CSEES installation directory. Secondly, data from a task environment can be imported to CSEES via Excel's data importer.

2.2 Analysis Methods

In addition to descriptive statistics, the initial set of methods supported by CSEES (signal detection theory (SDT) [5], fuzzy SDT [8], judgment analysis (JA) [3], and the skill score [7]) focus on analyzing performance on judgment tasks.

2.2.1 Signal Detection Theory

SDT models the detection of an event in a noisy environment. It focuses on the detection process in the presence of an evidence variable, "X", and noise. SDT assumes that the judge has a criterion value, C_h. When the properties of X exceed C_h, the judge asserts that the signal is present. The states of the world (signal or noise only) and the two possible responses ("yes", there is a signal or "no", there is no signal) create four classes of joint events: two are correct responses (hit and correct rejection) and two are errors (false alarm and miss) (Table 2).

Table 2. Signal Detection Theory Outcomes

		Signal + Noise	Noise only
Response	Yes	Hit	False alarm
	No	Miss	Correct rejection

From the outcomes, four probabilities are calculable:

- P(H|S): Probability of hit given a signal event (number of hits/number of signal events)
- P(FA|N): Probability of false alarm given noise (number of false alarms/number of noise only events)
- P(M|S): Probability of miss given a signal event (number of misses/number of signal events)
- P(CR|N): Probability of correction rejection given noise (number of correct rejections/number of noise only events)

SDT uses two parameters to model detection. Sensitivity is an index of the judge's ability to distinguish the signal from noise. Response bias is the judge's tendency to respond positively or negatively as a function of the four outcomes and the likelihood of a signal being present. With the assumptions of normality and of equal variance for the signal and noise distributions, sensitivity is the distance between the means of the signal and the noise scaled to the standard deviation of the noise distribution. The response bias is the likelihood ratio that an effect of the cutoff criterion is due to signal plus noise as opposed to noise alone.

In CSEES, students identify both the evidence (criterion) and judgment variables in the data set. They also supply the criterion value C_h and the conditional logic required to constitute a judgment of "yes". CSEES generates a spreadsheet that displays the criterion, the criterion threshold, and binary coded columns of data indicating when a signal, "yes" judgment, hit, miss, false alarm, and correct rejection occurred. Values for P(H|S), P(FA|N), P(M|S), P(CR|N), bias, and sensitivity are also reported. Students can then observe how the criterion threshold affects all of the results by either changing its value in the generated spreadsheet or by changing it in the signal detection interface and generating a new worksheet.

2.2.2 Fuzzy Signal Detection Theory

Fuzzy SDT can be used to model judgment performance when either the signal and/or the response are not Boolean. For example, fuzzy signal detection can be used when an operator is asked to make a probabilistic judgment about the likelihood of an outcome (see [2] for an air traffic conflict prediction example). Fuzzy measures provide ways to indicate the degree of evidence or certainty of an element's membership in a crisp set. With signal detection, where there are two non-overlapping categories (signal and noise), fuzzy measures can be used to assign the degree of set membership to both sets. The signal mapping function maps variables describing the state of the world into the set S (signal) with some membership degree in the range [0,1]. Mapping functions can map a single variable into the range [0,1], or can operate on some combination of variables. Similarly, the response mapping function assigns the result into the set R (response) with some membership degree in the range [0,1] based on a judgment of confidence that the signal is present, and/or the signal's perceived or reported severity, strength, or criticality.

The observed values of S and R are used to derive fuzzy set memberships in the calculation of H, M, FA, and CR. Implication functions are used for this purpose:

$$H = \min (S, R) \quad (1)$$

$$M = \max (S-R, 0) \quad (2)$$

$$FA = \max (R-S, 0) \quad (3)$$

$$CR = \min (1-S, 1-R) \quad (4)$$

P(H|S) is calculated by dividing the sum of the hit memberships of each event across the trials, by the sum of the signal membership values (S). To calculate P(FA|N), the sum of the FA memberships of all events is divided by the sum of the not-signal membership values (1-S). The final step involves the computation of measures from the fuzzy hit and false alarm probabilities. This step is essentially the same as in conventional SDT.

For fuzzy SDT, students use Excel's worksheet functions to define the mapping functions and generate the S and R membership values. When this data is identified in CSEES, a new excel worksheet is generated which displays membership values for H, M, FA, and CR as well as values for P(H|S), P(FA|N), P(M|S), P(CR|N), bias, and sensitivity. Because the computations in this sheet are dynamic, students can change the mapping functions in order to observe how the computations change as a result.

2.2.3 Judgment Analysis

Judgment analysis (JA) is based on probabilistic functionalism, which designates the organism-environment interaction as the primary unit of study [3]. It considers both internal (cognitive) and external (environmental) aspects of judgment. It has been used to analyze human judgment performance in many domains including medical diagnosis, weather forecasting, and air traffic conflict prediction (see [1] for an example).

JA uses the Lens Model; its commonly used form provides symmetric models of a judge and the environment. The left side of Figure 1 depicts the task environment in terms of the cues available and the environmental criterion (Y_e) to be judged. Cues and the criterion are related by statistical correlations known as ecological validities. The right side depicts the judge using the cues to render a judgment (Y_s) about the environmental criterion. Correlations between the cues and the judgments are the cue utilizations. The particular pattern of cue utilizations exhibited by a judge determines the cognitive judgment strategy. Achievement (the top arc reflecting the correlation between the judgments and the environmental criterion) will be maximized when the pattern of cue utilizations mimics the pattern of ecological validities.

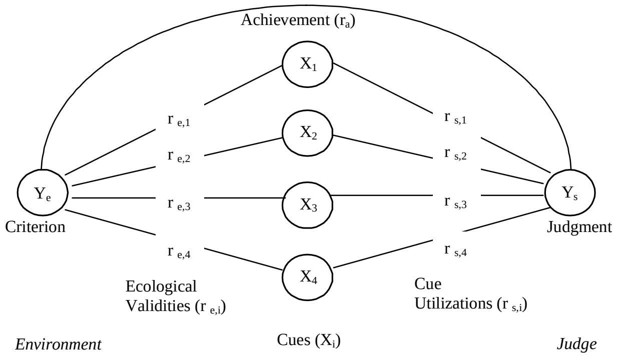


Figure 1. Double System Lens Model of Judgment

The Lens Model structure yields the Lens Model Equation (LME) [3]:

$$r_a = G\,\mathrm{R_e}\,R_s + C\sqrt{1-R_e^2}\sqrt{1-R_s^2} \qquad (5)$$

where r_a is achievement, G is linear knowledge, R_e is environmental predictability, R_s is cognitive control, and C is nonlinear knowledge.

With the LME, JA aids in understanding the source of less than perfect judgment by decomposing achievement. The first term in the LME is the product of linear knowledge, environmental predictability, and cognitive control. R_e is calculated as the multiple correlation of the environmental linear regression model. It represents a limit on judgment performance based on the predictability of the task environment. G indicates the level of judgment performance if the environment and the judge were completely linearly predictable. It is calculated as the correlation between the predictions of the two (environmental and cognitive) regression models. The adequacy of a judgment strategy (in terms of beta weights in the cognitive regression model) represents the linear knowledge. The consistency with which a judge can execute his or her strategy is captured by cognitive control, calculated as the multiple correlation from regressing the judgments on the cue values.

The second term in the LME deals with nonlinear effects not captured by the linear effects in the first term. The values of R_e and R_s appearing in this term have already been discussed. Nonlinear knowledge is calculated as the correlation between the residuals of the environmental and the cognitive linear regression models. Its role is to identify if the judge is capturing non-linear components in the environment that are not captured in a linear model.

CSEES gives students the ability to perform a double systems lens model analysis. The student identifies both the criterion and judgment data in addition to potential cues. In order to determine which cues to include in the model, CSEES gives students the means necessary to run both a correlations analysis between each of the cues and a full-subset regression analysis with the criterion and judgment for each cue combination. When the final set of cues has been selected, students can perform the lens model analysis. The results are presented in a separate excel worksheet which reports the full regression models and regression statistics, including predicted values and residuals, for both the judgments and criterion. It also computes and displays the values of each parameter in the LME. Finally, it generates a lens model diagram, similar to Figure 1, in which all the model specific values are reported. Students have the ability to generate models using different cue combinations in order to see how the parameters of the LME are affected.

2.2.4 Skill Score

As an alternative to regression-based approaches, Mean Square Error (MSE) has been used to measure judgment performance [7] where n is the number of judgments:

$$MSE_Y = \frac{1}{n}\sum_{i=1}^{n}(Y_{si}-Y_{ei})^2 \qquad (6)$$

Different decompositions of MSE have been considered by researchers (see [3] and [7] for examples). The form of the judgment performance standard is one such difference. Stewart [9] uses a constant judgment based on the average value of the criterion. The correspondence (i.e., “goodness”) of the standard is defined as MSE_R where $\overline{Y_e}$ is the mean of the criterion:

$$MSE_R = \frac{1}{n}\sum_{i=1}^{n}(Y_{ei}-\overline{Y_e})^2 \qquad (7)$$

The skill score (SS), defined as the ratio between the MSE of the judgments and the MSE of the standard subtracted from unity, is a derived measure of judgment performance. SS is positive when the judgments are better than the standard (MSE_Y < MSE_R). When the SS is zero, the judgments are as good as the standard (MSE_R =MSE_Y). When it is negative, the judgments are worse than the standard (MSE_Y > MSE_R).

$$SS = 1 - \left[{}^{MSE_Y}\!\big/\!{}_{MSE_R}\right] \qquad (8)$$

Murphy [7] developed the SS to decompose the MSE by substituting the equations for MSE_Y and MSE_R into the form of the SS above:

$$SS = (r_a)^2 - \left[r_a - {}^{s_{Ys}}\!\big/\!{}_{s_{Ye}}\right]^2 - \left[{}^{\overline{Y_s}-\overline{Y_e}}\!\big/\!{}_{s_{Ye}}\right]^2 \qquad (9)$$

The first term, the square of the LME achievement, has been described. The second term, called conditional bias or regression bias, measures whether the judge has appropriately scaled judgmental variability to situational variability. Conditional bias illustrates a tendency to produce judgments on an interval different than found in the true situation. The third term, unconditional bias or base rate bias, measures the overall bias in the judgments, illustrating a tendency to over- or under-estimate the criterion. This bias equals zero only when the mean of the judgments equals the objective base rate.

CSEES gives students two means of computing a skill score. In the first case Equation 8 is used. This requires the student to identify both the criterion (Y_e) and judgment (Y_s) data sets. When the analysis is run, a worksheet is generated which reports MSE_Y and MSE_R along with the skill score. The second uses Equation 9 where the r_a value comes from a double systems lens model analysis.

2.3 Help

The CSEES help system has yet to be completed. There will be two types of help available to students: documentation and tutorials. Documentation will serve two purposes. Firstly, it will explain the theoretical foundation of the different analysis methods used in the program. Secondly it will instruct the student on how to perform these analyses using the software by explaining how each step in an analysis method is performed and how it relates to the method's theoretical foundation.

Tutorials will augment the documentation by providing example based walkthroughs of the analysis process. Such tutorials will cover all aspects of the system, guiding the student through the collection of data using the task environments, importing the data using Excel, and analyzing it using one or more of the analysis methods.

2.4 Graphical User Interface

The CSEES graphical user interface guides the student through the generation and analysis of data. The primary interface (Figure 2) is a dialog box with menus and a toolbar to open, save, and create new excel files in the CSEES environment. It also gives students access to the task environments, analysis methods, and help.

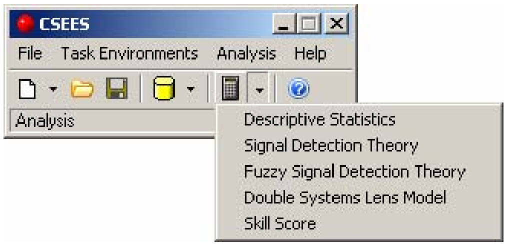


Figure 2. The CSEES Graphical User Interface

Task environments are activated via the Task Environments menu and the cylinder icon on the toolbar. The list of task environments is generated at run time when CSEES examines the contents of its *Task Environments* folder and generates links to the EXE, BAT, and LNK files. All task environments are launched as separate processes. Students and faculty can add new task environments to CSEES by placing a shortcut to the desired application in the *Task Environments* folder.

Excel itself serves as an important part of the CSEES interface by providing several important features. Firstly, it gives students a familiar spreadsheet like interface which allows them to enter and manipulate data. Secondly, it contains a flexible data importer which can be used to import data from the task environments. Finally, it allows for dynamic function based computation which permits students to inspect the CSEES generated computations in order to see how they were performed and to dynamically change parameters in the spreadsheet in order to see how they affect the results.

In addition to the spreadsheet interface, each analysis method has an associated dialog box which can be accessed through the analysis menu and the calculator button on the toolbar. These dialog boxes are used to collect user input. When activated, CSEES scans the active Excel worksheet in order to identify column based series of data (variables). Students then use the dialog box to identify which variables they want to use in the computation and enter any other relevant input. When the analysis is run, a new Excel worksheet is generated with the desired results as well as all the intermediate steps used to compute them.

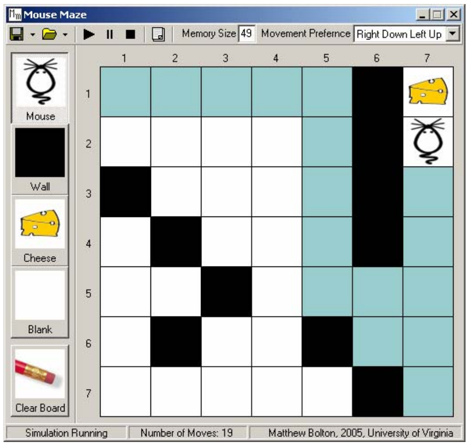
The CSEES help system can be accessed at any time through the help menu, by clicking on the ? button available on every CSEES dialog box, or by pressing F1. When completed, the CSEES help interface will provide access to both fully indexed and searchable content. CSEES help will be context sensitive, where users will be able to immediately jump to topics relevant to the task they are performing.

3 Results

As a proof of concept prototype, incremental versions of CSEES have been used to support a graduate course titled "Quantitative Models of Human Performance" in the Department of Systems and Information Engineering at the University of Virginia. The course surveys model-based quantitative/computational approaches to measuring human performance in complex dynamic systems. Many of the course assignments include applying the quantitative approaches to a human-machine system problem. Of the seven types of models covered in the course, CSEES currently supports topics in four: rule-based models, SDT, fuzzy decision theory, and JA approaches. To address rule-based models, CSEES includes a simulated navigation task environment and to support the others, a simplified air traffic conflict judgment task.

3.1 Simulated Navigation Task Environment

CSEES provides an opportunity for students to learn about rule-based models (where *if-then* patterns are used to control the actions of an agent based on the validity of assertions [10]) without having to learn how to build them. The simulated navigation task environment presents students with a scenario in which a robot mouse (the agent) attempts to find a piece of cheese in a maze (Figure 3). This is of educational interest because it provides students with an opportunity to observe how the task environment, agent preferences, and cognitive capabilities interact to impact an agent’s behavior.



Memory Size 49 Movement Preference Right Down Left Up

Simulation Running | Number of Moves: 19 | Matthew Bolton, 2005, University of Virginia

Figure 3. Simulated Navigation Task Environment

The effect of environment is illustrated through the students’ ability to control the maze. The students can construct the maze by specifying where the mouse can move (white squares), where it can not (black squares), where it starts, and where the cheese is located.

Students can control agent preferences by specifying how the mouse selects where he moves. The mouse can only interact with its environment by moving in one of four directions: up, down, left, and right. Its preferences are controlled by a forward chaining rule-based system. Students can select one of nine different move order preferences for the mouse. The first eight specify specific move order preferences, where a particular preference order is assigned to each of the four directions. The 9th option is to select a direction at random.

Regardless of the selected preference, the simulation has several rules built into it: the mouse will never return to its last position unless it is the only move available, the mouse cannot move into a black square, and the mouse will always move to the cheese when in an adjacent square.

The student also has the ability to manipulate the cognitive capabilities of the mouse by controlling the size of its memory. The mouse can remember having visited a finite number of previous positions (represented as shaded squares). The mouse prefers to move to an unvisited area over one that it remembers having visited.

CSEES displays the path the mouse follows and records the number of moves required to reach the cheese. It then stores this information in a data log along with a profile of the maze environment, the position of the cheese, the mouse’s initial position, its movement preference, and its memory capacity. This allows students to design experiments in which any of these factors can be varied.

3.2 Air Traffic Conflict Judgment Task

CSEES provides an opportunity for students to engage in a judgment task and then model their own performance data. The judgment domain is air traffic conflict prediction where students judge the probability that the ownship at the center of the display will conflict with another simulated aircraft (small triangle) flying at the same altitude (Figure 4). A conflict occurs when the traffic aircraft enters the protected zone of ownship, defined by a five nautical mile (NM) radius around the ownship (the innermost circle). Noise is injected into the speed, heading, and lateral position of the traffic. Each trial runs for a random preview time before pausing for the judgment. At the judgment time, a data entry area collects a probability of conflict judgment and provides control to initiate the next trial. To enter a judgment, the student uses a slide bar or moves a gray slide bar knob. The outcome of the trial (i.e., allowing the student to monitor whether the traffic conflicted with ownship and where it was relative to ownship at the point of closest approach) is the feedback provided. The program produces an output file with a record for each trial including the participant’s judgment, the criterion (also represented as a probability), and state data for each aircraft including sensor noise estimates. See [2] for more details about the task environment.

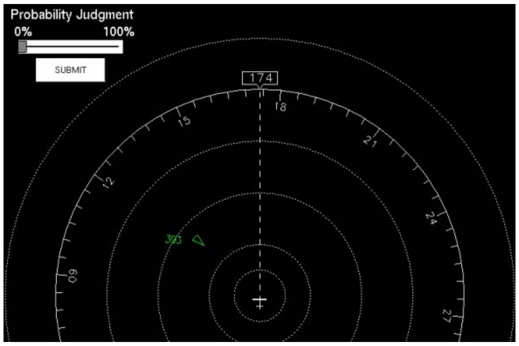


Figure 4. Air Traffic Conflict Judgment Task Environment

Students can use CSEES to analyze the results using a variety of different methods and observing how performance is modeled by each.

For example, a skill score can be generated simply by identifying the criterion and judgment variables. SDT can be used by identifying the criterion and judgment variables in addition to specifying a probability for the criterion threshold and the threshold for a “yes” judgment. Because both the judgment and criterion are probabilities with range [0, 1] they can be used in a Fuzzy SDT analysis (the criterion being S and judgment being R) without the student having to define mapping functions. Finally, a double systems lens model analysis can be performed by having the students select a set of cues from the aircrafts’ state data along with identifying the criterion and judgment.

4 Discussion and Future Work

During the fall of 2004, students taking the "Quantitative Models of Human Performance" course, a 600-level course offered at the School of Engineering and Applied Science (SEAS) at UVa, used the CSEES system to complete several of their homework assignments. During the course, students provided positive feedback about the software. In addition, at the end of the course, some of the students completed a course evaluation administered through the university (participation was optional). As part of the course evaluation, the students were asked whether the homework assignments helped them learn the course material. Most responses were the highest rating possible (Table 3). These responses were better than those for the other 600-level courses offered that semester in SEAS.

Table 3. Responses to the Course Evaluation Question: Did the Homework Assignments Help You Learn the Material?

Results for Quantitative Models of Human Performance - Fall 2004

Total	Mean	Std. Dev.	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
4	4.75	0.50	3	1	0	0	0
			75.00%	25.00%	0.00%	0.00%	0.00%

Results for SEAS, 600-level courses - Fall 2004

Total	Mean	Std. Dev.	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
604	4.26	0.91	297	216	55	26	10
			49.17%	35.76%	9.11%	4.30%	1.66%

The next stage of CSEES development will involve the construction of the context sensitive help and tutorial system. Future courses of "Quantitative Models of Human Performance" will use the latest version of CSEES, and will provide additional feedback. Furthermore, studies to analyze both the educational effectiveness and usability of CSEES are currently being planned.

In order to best serve the needs of the community, this project is intended to be a communal effort. A website (http://cog.sys.virginia.edu/CSEES/) is in the process of being developed in order to provide free access to the software. As such, we encourage professors and students to send any contributions and recommendations relating to any of these topics to the authors. We are convinced that with strong community involvement, this educational software can be a success.

5 Acknowledgments

The authors thank the representatives at the eight universities who responded to our questions about their courses. The authors thank Ann Bisantz for providing the Java-based SDT software. The authors thank the three anonymous reviewers for their feedback on an earlier version of this paper.

References

[1] E. J. Bass & A. R. Pritchett. "Human-Automated Judgment Learning: A methodology for examining human interaction with information analysis automation," Accepted to the *IEEE Transaction on Systems, Man, and Cybernetics, Part A.*

[2] E. J. Bass & S. Rudra, "Comparing three methods of measuring human judgment performance in an air traffic conflict prediction task," Proc. IEEE International Conference on Systems, Man, and Cybernetics, Hammament, Oct. 2002.

[3] R. W. Cooksey, *Judgment Analysis: Theory, Methods, and Application*, Academic Press, New York, 1996.

[4] R. M. Felder, D. R. Woods, J. E. Stice, and A. Rugarcia, "The Future of Engineering Education II. Teaching Methods that Work," *Chemical Engineering Education*, Vol. 34, No. 3, pp. 26-39, Winter 2000.

[5] D. M. Green and J. A. Swets, *Signal Detection Theory and Psychophysics*, John Wiley and Sons, New York, 1989.

[6] S. Guerlain, C. Hayes, A. R. Pritchett, and P. Smith, "Exercises/Techniques for Teaching Cognitive Systems Engineering," Proc. Human Factors and Ergonomics Society 45th Annual Meeting, Minneapolis, pp. 453-457, October 2001.

[7] A. H. Murphy, "Skill scores based on the mean square error and their relationships to the correlation coefficient," *Monthly Weather Review*, Vol. 116, pp. 2417-2424, Dec. 1988.

[8] R. Parasuraman, A. J. Masalonis, and P. A. Hancock, "Fuzzy signal detection theory: Basic postulates and formulas for analyzing human and machine performance," *Human Factors*, Vol. 42, No. 4, pp. 636-659, Winter 2000.

[9] T. R. Stewart, "A decomposition of the correlation coefficient and its use in analyzing forecasting skill" *Weather and Forecasting*, Vol. 5, No. 4, pp. 661-666, 1990.

[10] P. H. Winston, *Artificial Intelligence*, 3rd Edition, Addison-Wesley, Cambridge, 1992.