From Chaos to Cooperation: Teaching Analytic Evaluation with LINK-UP

Jason Chong Lee, Sirong Lin, C. M. Chewar, D. Scott McCrickard, Alan Fabian, Andrew Jackson
Department of Computer Science & Center for HCI, Virginia Tech, USA
{chonglee, silin, cchewar, mccricks, alfabian, andrewj}@vt.edu

Abstract: Our work seeks to design and develop a tool to enable the learning of human-computer interaction while contributing to a growing and evolving development environment. This paper focuses on teaching students about analytic evaluations, a popular evaluation method that can be conducted rapidly and inexpensively. With our module, designers and evaluators can debate claims about the interface being developed, helping to discover problems with a design—shown in our usability evaluation of the module. We discuss how the module can support design reuse and speculate on broader applications for this type of learning and development environment.

Introduction

Emerging areas of study like human-computer interaction (HCI), with a varied, multidisciplinary flavor, are difficult to teach in higher education settings. As with many emerging areas, HCI appears to many students as a seemingly random collection of topics, without the structure important to facilitate learning. Until that structure can be identified and leveraged in educational teaching tools, teachers will have difficulty in teaching the subject, and students will not gain a true appreciation for the field.

In keeping with the goals of the E-Learn conference, our efforts seek to design and develop a tool that will enable students to learn basic concepts of HCI while contributing to a growing and evolving effort. Key in our system are analytic evaluation methods for software interfaces, which enable students or design professionals to conduct a usability test in minimal time with few participants. Information from the evaluation is stored for reuse in other phases of development, and by future system developers. Important learning objectives in typical HCI courses and applied project work involve preparing an interface to be evaluated, conducting the actual evaluation, and interpreting the results to guide iterative system development.

While analytical usability evaluation methods are touted as the most cost-effective and desirable form of interface testing within industry (Nielsen, 1994), few tools are available to support related learning objectives. Unfortunately, there are no known analytic evaluation tools within an integrated design environment, which would be useful for digitally developing student project materials and design rationale. Like students of other design disciplines, interface designers encounter valuable learning opportunities when they have access to many examples of existing design products and multidisciplinary perspectives. Therefore, the most valuable design and evaluation tools should be optimized for browsing and retrieving reusable design knowledge, as well as sharing and distributing design activities beyond a single lab. With this high-level vision, our efforts target development and initial testing of such an analytic evaluation tool within an emerging web-based, integrated design environment—LINK-UP.

Background

Notification Systems Design with LINK-UP

To develop and validate our prototype design environment and analytical testing tool, we have constrained focus to a specific challenge within the field of HCI—developing notification systems. As a specific type of computer interface, *notification systems* typically are used for very brief information interactions, unlike most interfaces that are used for extended periods. Users value notification systems for reacting to and comprehending time-sensitive information, without introducing unwanted interruption to ongoing tasks (McCrickard, Czerwinski, & Bartram, 2003). Examples include news tickers, alerts or pop-up windows, automobile displays, system or network monitors, and context-sensitive help agents. We have found that notification system design problems can make very rich assignments for HCI students—requiring thoughtful consideration of multimodal design tradeoffs and information visualization techniques to effectively present interface state transitions to users (Chewar & McCrickard, 2003).

LINK-UP, or (Leveraging Integrated Notification Knowledge with Usability Parameters), is an integrated design environment for notification systems that allows designers to proceed through development activities while accessing and creating reusable design knowledge. Complementary to a scenario-based design approach (e.g., Rosson and Carroll's *Usability Engineering* (2002)), the LINK-UP system provides interactive modules and tools for activities that include requirements analysis (brainstorming) with task-models, participatory negotiation with end-user stakeholders and analytic and empirical evaluation (Chewar, Bachetti, McCrickard, & Booker, 2004). Through the progressive use of LINK-UP, developers create and reflect upon interface artifacts defined at ever-increasing fidelity. LINK-UP helps developers focus HCI concerns on the critical parameters of notification system design—controlling appropriate levels of user interruption, reaction, and comprehension, or *IRC parameters (*McCrickard, Chewar, Somervell & Ndiwalana, 2003)—with its integrated access to detailed notification design tradeoffs (or *claims*) established through hundreds of previous design efforts, empirical testing results, and concise summaries of psychology and human factors literature. As developers use LINK-UP, these basic research results are applied and put to the test with new design efforts, continuously advancing the state-of-the-art in notification research.

Design Knowledge Reuse and Accumulation

Many of the activities supported in the LINK-UP modules are intended to facilitate reuse and accumulation of design knowledge. Certainly, reuse can occur at any or all of the stages in the software design process. However, different benefits are available at each stage. Sutcliffe and Carroll point out that the earlier in the development process we could reuse design knowledge the larger the payoff is in the end (1999). As a scientific field within computer science, HCI efforts typically seek to eliminate usability problems and help users accomplish their task-related goals. Unfortunately, efforts to create reusable design knowledge in HCI attracted little attention compared with Software Engineering until the mid-1990s. In recent years, there has been increasing research interest in understanding how to create and benefit form reusable design knowledge. Carroll first proposed *claims* as a reusable knowledge object in his Task-Artifact Theory (Carroll and Rosson, 1992). To widen the scope of reuse, the knowledge contained within claims and their associated artifacts must be classified and generalized; fortunately, a schema and method for classifying claims is introduced by Sutcliffe and Carroll (1999). LINK-UP is intended to provide a framework for design knowledge reuse, coupling a claims library with a suite of design-support tools (Chewar, Bachetti, et al., 2004). Transforming these theories into reality is important for the HCI researcher.

Encouraged by initial acceptance of our broad approach to computer-aided design of software interfaces, recent efforts focus on development and validation of specific modules within LINK-UP. As we expect that LINK-UP will be part of an internationally disseminated suite of web-based tools to support HCI education and distributed project work, we are eager to receive feedback from and urge early adoption within the E-Learn community. Arguably the most critical module within LINK-UP, the work reported here relates to the *Analytic Module*.

Analytic Evaluation in LINK-UP

Generally, an analytic evaluation is a process in which evaluators (often experts) identify usability problems in a partially implemented interface by 'simulating' the way they think users will perform certain tasks. One of the fundamental ideas in HCI is that usability evaluation should start early in the design process, optimally in the stages of early prototyping. When critical design flaws are detected earlier, the chance increases that they can and will be corrected without requiring costly evaluation with actual users (i.e. empirical evaluation) or major code changes. The analytic evaluation module will ultimately support several different types of evaluations, such as the use of heuristics (Nielsen, 1994).

However, in this prototype, we employ a critical parameter comparison technique that contrasts the *design model* (expressing the designer's interpretation of user requirements) with the *user's model* (characterizing the actual user's experience)—concepts originally introduced by Norman (1986). The Analytic Module within LINK-UP provides a tool that facilitates execution of an analytic evaluation method, records evaluation results, and provides an estimate of what the user's model IRC parameters might be if the system were fully developed and tested with users (the *analytical model).* That is, to support the goal of early evaluation and introduce students to this essential practice, designers are able to get a preliminary sense of the levels of interruption, reaction, and comprehension that their system design will cause within users. To obtain the analytic model, designers prepare an overview of the key interface interactions (or a *system image*, which may include a system prototype description, interaction scenario, and prototype artifacts). Expert evaluators use this overview to walkthrough and analyze the interface with a series of questions established as an IRC analytic calculation method (Chewar, McCrickard, &

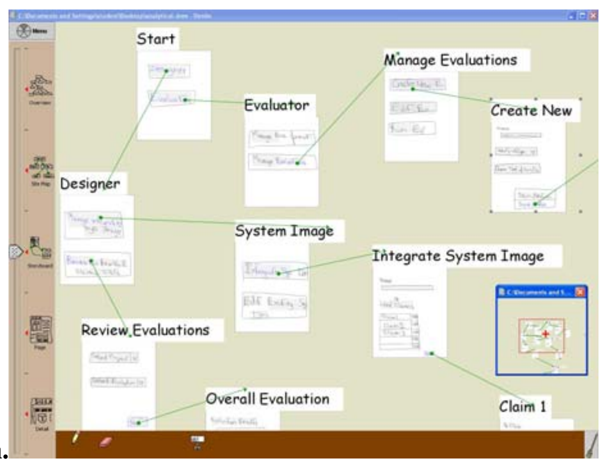
Sutcliffe, 2004). During the evaluation, experts also consider the designer's claim statements about psychological effects that the interface will cause in users. Claims can relate to problem spaces (e.g. describing situational characteristics and requirements) and be characterized by a design model IRC, or claims can express proposed design solutions. The module assists designers with matching problem claims with design claims—a process that will assist designers later in pinpointing the underlying causes, if the design model and analytic model IRC values do not match after the expert evaluation.

Analytic Module Development

Analytic Module Design

We began brainstorming about the design of the analytic module by considering all of the broad activities that users would need to accomplish as they proceeded through an analytic evaluation. Two important realizations emerged: first, the system would be used by two distinct types of users; second, the module would need to support several activities beyond just the evaluation execution. Each of these ideas is elaborated on below. Most of our early design work was conducted with rapid prototype development software that allowed quick storyboard sketching and linking of steps within the module (Figure 1a). In the early process, we were unconcerned with the specifics of information design—we focused solely on ensuring that all essential processes and steps were included. We received frequent feedback on several iterations of our rapid prototypes from members of our larger research group. As our ideas solidified, we emerged with a coherent task analysis and screen-flow for the analytic module.

a.



b.

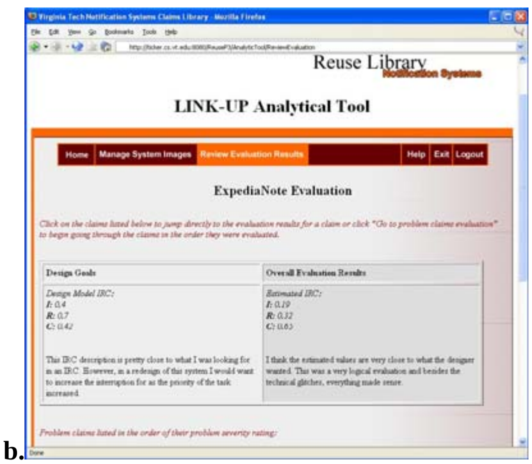


Figure 1a: Early storyboard of the analytic module; **1b:** Screen from the functional prototype used in pilot testing.

There are two types of users that will use the Analytic Module, designers and evaluators. Designers are the individuals who will develop the system to be evaluated using scenario-based design and LINK-UP's claims library. Evaluators are the individuals who will be assessing the system's usability. Ideally, they would be experts in HCI and be familiar with scenario-based design claims analysis (Rosson and Carroll, 2002). According to our needs analysis, we determined that the analytic module should be divided into three sub-modules: *Manage System Image*, *Evaluate System Image* and *Review Evaluation Results*. The three sub-modules correspond to the basic sequence of tasks (Figure 2). The designer's and evaluator's tasks will be described respectively below.

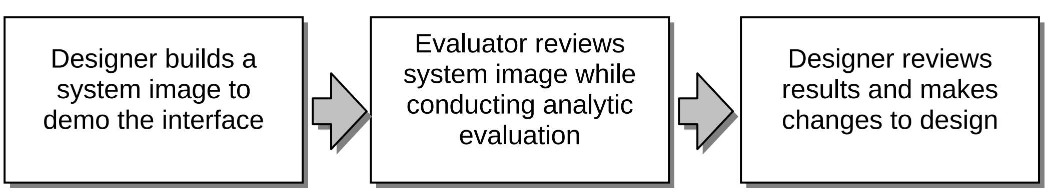


Figure 2: Basic sequence of tasks of the analytic module.

First in the task sequence, supported by the *Manage System Image* module, designers will either add a new system image into the module or edit an existing system image. To create a new system image, the designer will need to select an ongoing design project. Each design project will be associated with a list of problem claims developed during the requirements gathering and participatory negotiation phases of the LINK-UP system. The designer then links necessary design claims to each problem claim. Each design claim should solve part of or the whole problem addressed in the problem claims. Design claims can either be newly developed by the designer or be reused, drawn from the claims library. Artifact representations are also uploaded to describe the system image. An artifact representation shows some aspect of the system that is being designed. They can be pictures, screenshots or movies of the system interface. Before this process, the designer would have already established a design model IRC, which represents the intended IRC value for the system being designed. Once this process is complete, the system image will be ready to be evaluated. Designers can also edit existing system images after creating them in response to evolving requirements or to account for system redesign decisions.

Second, through the *Evaluate System Image* module, HCI experts can analytically evaluate system images. The analytic module will step the evaluator through the evaluation method and record the results of the assessment. During the process, the evaluator would first become familiar with a summary of the system image to develop an overall understanding of the notification system design. This is accomplished by browsing through the system description, the interaction scenarios, the design model IRC values, and the prototype artifacts. Then the evaluator steps through each problem-design claim pair and rates how well each design claim addresses the associated problem claim on a four-point scale (*major flaw, flaw, minor flaw,* or *no problem*). This rating is based on how well the upsides (positive psychological effects) expressed by the design claim either:

- Support the upsides of the linked problem claim,
- Mitigate the downsides of the linked problem claim or other design claims.

Evaluators provide comments for each claim pair to explain their ratings. At the end of the evaluation, the evaluator runs the IRC equation algorithm (based on the method introduced in (Chewar, McCrickard, and Sutcliffe, 2004)) to get the estimated IRC value for the whole system.

As the final step in the task sequence, in the *Review Evaluation Results* module, designers will be able to review the results of evaluations that experts have run on their system images. At the start of this interaction session, the designer will see overall evaluation comments and will be able to compare the design model IRC and estimated IRC for their prototype. This gives the designer a sense of how well the system might fulfill its most critical user goals, if fully developed. They can then view the ratings and comments for each claim pair to elaborate on specific, causal design features. The designer can use the ratings and comments to improve their designs. They can also revise the problem-design claim linkages, or even modify their original design claims. Since the claims and their evaluation results are archived in the claims library (and accessible beyond the single project), future developers will be able to benefit from this e-learning opportunity.

Analytic Module Implementation

In order to use the analytic module in an e-learning environment in a manner that would facilitate distributed project work, out-of-class assignments, and online education of computer science HCI courses (i.e., Usability Engineering), we designed and implemented the analytic module as a web-based tool. This strategy enables "anytime and anywhere" access for designers and evaluators. To achieve our goal, JavaServer Pages (JSP) technology and Java Servlet technology were selected to develop dynamic web pages, which enhance the interactivity between learners and our module. As explained in programming guides, Java Servlets are platform-independent, server-side modules that fit seamlessly into a Web server framework and can be used to extend the capabilities of a Web server with minimal overhead, maintenance, and support.

The module consisted of more than 20 dynamic html pages, generated by 13 servlets. We used a tabbed interface to provide users with flexible progression through the linear major task processes. Figure 1b depicts one of the primary screens within the version we used in our pilot testing (see previous section).

Development efforts were carried out using Eclipse [1], which was ideal for supporting multi-developer collaborative work in an online, integrated development environment (IDE). We also used Hibernate, the most popular Object Role Modeling (ORM) solution for Java, to access and update a MySQL database using synchronized and persistent Java-like objects. Together, these state-of-the-art web programming technologies and platforms provide maximum extensibility and flexibility within our module—making it an ideal prototype that can generalize to future online software design facilitation tools.

---

[1] An open source platform available at http://www.eclipse.org.

Pilot Testing Evaluation Methods

We conducted a three stage formative empirical usability evaluation of the analytic module—a process in which potential users actually used the prototype to accomplish real tasks. This evaluation was conducted after the development of the module prototype to get feedback from actual designers and study how it is used in a realistic situation (Rosson and Carroll, 2002). The evaluation had two goals, to establish whether:

- Designers can use the module to help identify and resolve design problems,
- Through use of the module, designers learn more about claims-based usability engineering.

The first goal arises because an overall objective of the LINK-UP system is to support the usability engineering process for notification systems and enhance HCI education activities (Chewar, Bachetti, McCrickard, and Booker, 2004). The second goal arises because many initial users, especially those in an academic setting, are unlikely to be experienced with claims and scenario based design. The analytic module addresses the user's tendency to learn about a new tool through active and informal, exploratory use (Carroll and Rosson, 1987).

In our pilot testing, we were assisted by seven participants who were all undergraduate students in a semester-long undergraduate research seminar, focusing on design and evaluation of notification systems. All of the students in the class were working on a design project intended to help them learn about notification systems, claims and claims-based usability engineering. The students had spent recent weeks individually designing a notification display that helps users monitor and react to constantly changing airplane ticket prices. At the start of our pilot testing, each student had a semi-functional prototype notification system to support this typical notification task.

The evaluation was run over a period of three weeks, with one hour-long session held each week. To prepare for the evaluation, participants added to the claims library their problem and design claims. In *Session 1*, the participants created the system image for their prototypes using the Analytic Module. The participants had the option of linking their own (new) design claims or reusing existing design claims to their problem claims. After the system images were created, each participant completed a questionnaire containing multiple-choice statements (rated on an eleven-point Likert scale) and general questions that required written answers (Figure 3). This questionnaire assessed the designers' opinions about how well the system image represented the prototype design. In *Session 2*, each participant evaluated the system image of a prototype developed by another participant in the last session. This was also followed by a questionnaire that gathered feedback on how effectively the participants were able to evaluate the prototype using the module (Figure 4). In *Session 3*, the participants reviewed the evaluation results for their system images and completed a form to express comments and reactions to all information resulting from the evaluation (quality of evaluator feedback and utility of the general approach). This session was followed by a questionnaire (Figure 5) that focused on whether the evaluation gave them useful design feedback and whether evaluators identified any problems with the claims. Each questionnaire is further summarized below.

Evaluation Results and Discussion

Based on the results from the first session (Figure 3), most of the students thought the analytic module matched their understanding of notification systems, claims, and IRC values. They also agreed that the sequence of steps in creating a system image made sense. Most students seemed to understand the concept of linking problem claims to design claims. However, the variation in the responses for these questions was high. Of the three participants that responded to question 5, only one thought the reused design claim matched the problem claim well. The other participants noted in the general question responses that the search functionality in the claims library was difficult to use and the design claims they found were only adequate matches. The system image was a fair representation of the student prototypes, but several students commented that there was not enough information in the image. One student thought including a link to a prototype executable, if it were available, would be useful.

Based on the results from the second session (Figure 4), students had a good understanding of the steps involved in evaluating a system image. Most participants did not find the process too confining or open-ended. However, one of the students noted an evaluator could only give a single rating for each problem and design claim pair, and there was only a single field in which to enter comments. There was no way to separate comments related to the design of the prototype and comments related to the quality of the claims themselves. In spite of this, all of the participants (acting as evaluators) had comments on possible improvements to their classmate's system design. We were encouraged that the majority thought there was enough information in the system image to effectively evaluate the prototype.

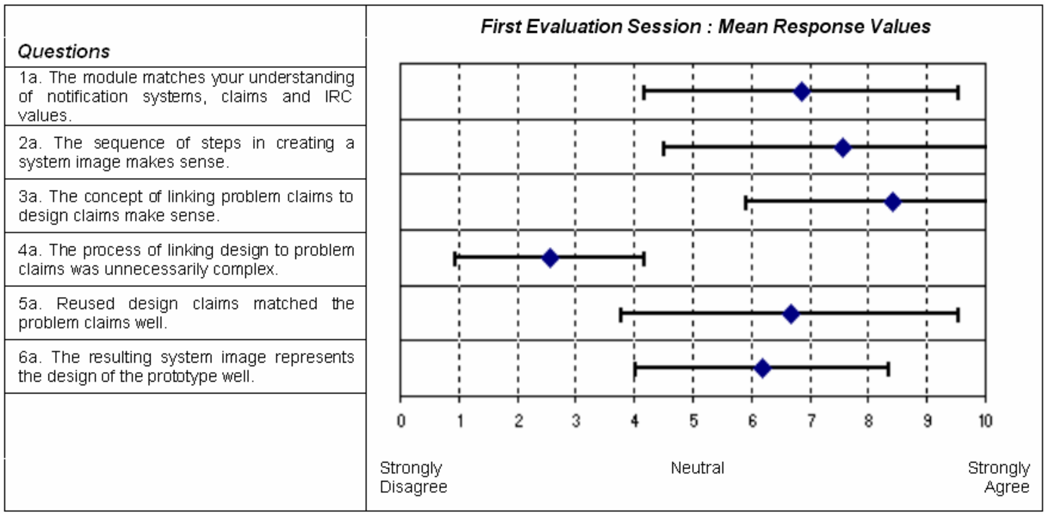


Figure 3: Means with standard deviations of the participant responses to the questions given in the first session of the evaluation. The answers were based on an eleven point Likert scale.

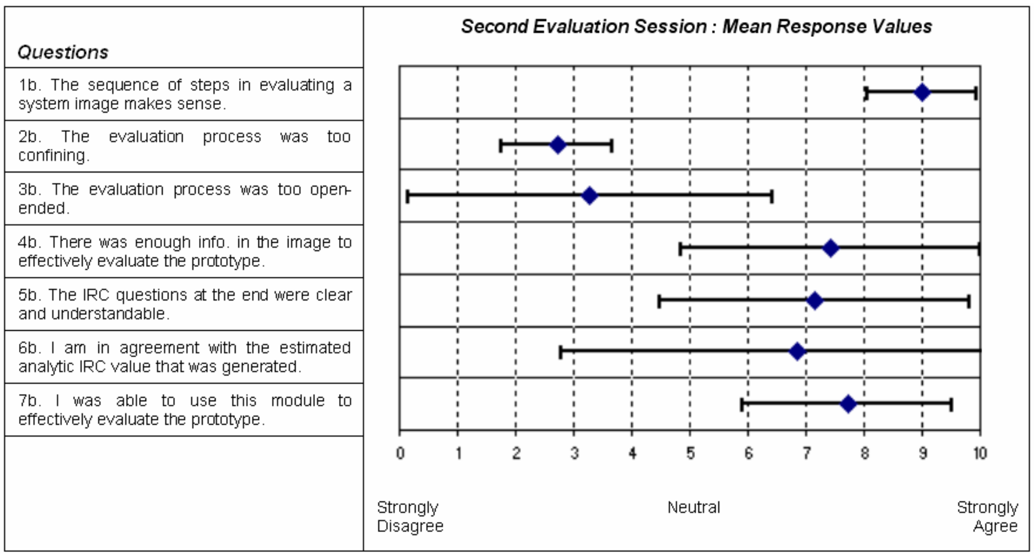


Figure 4: Means with standard deviations of the participant responses to the questions given in the second session of the evaluation. The answers were based on an eleven point Likert scale.

However, three of the participants would have wanted access to the prototype executable during the evaluation so he/she could see the prototype application in action while evaluating it. Two participants also thought it would help to have the prototype screenshots linked directly to the relevant design claims. Overall, the participants believed that they were able to use the module to effectively evaluate the prototype. All of the participants were able to identify at least one problem in the system images they evaluated. These problems were a mix of issues with the prototype designs and quality of the claims in the system image. In addition, three of the participants did not agree with the estimated IRC value that was generated at the end of the evaluation. Two of the participants found the questions they answered to generate the estimated IRC value were not worded clearly. The other participant thought the system image he evaluated would have a different IRC value depending on the state of the prototype.

Based on the results from the third session, only three of the seven participants made design changes to their prototypes based on their evaluation results. Several factors contributed to this low number. One student found that the participant whom evaluated his system image did not provide enough comments for him to make any design changes.

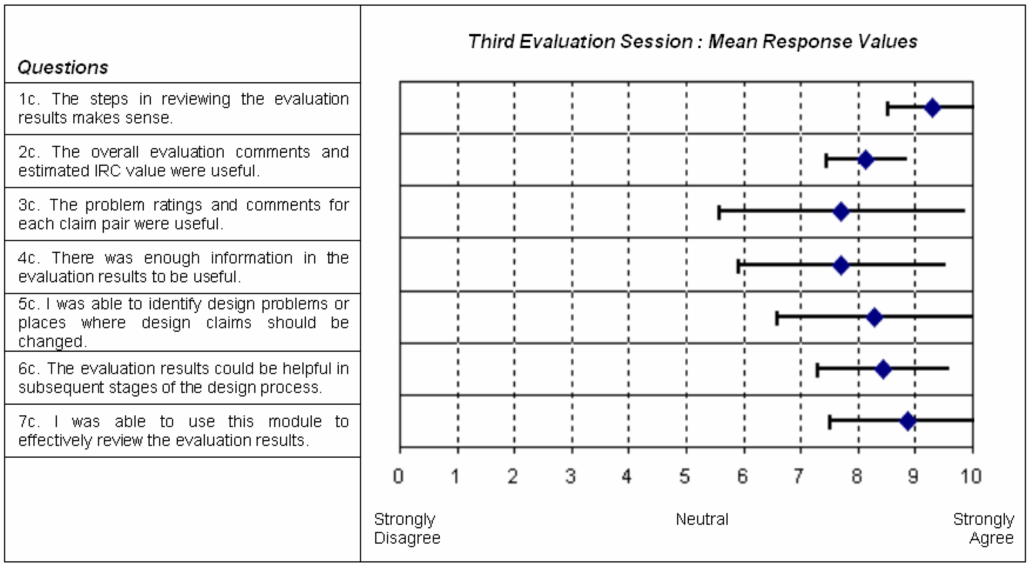


Figure 5: Means with standard deviations of the participant responses to the questions given in the third session of the evaluation. The answers were based on an eleven point Likert scale.

Four students did not agree with some of the design problems that their evaluators identified and wanted another evaluation of their system image with a different evaluator. Also, one student agreed with the minor flaws that had been identified in his design but he decided not to modify his design because, based on the information in the system image, he could not tell how other parts of his design would be affected if he addressed those problems. Nevertheless, most students saw how the evaluation results could be useful within the design process and thought that there was enough information in the evaluation results to help them improve their design (Figure 5). Compared to the evaluation results for the overall prototype designs, there were more cases of success in identifying, finding, and explaining problems with specific problem and design claims. When evaluators found a problem with a claim, six of the seven participants used the evaluation results to identify mistakes they had made when they first developed their problem and design claims and recognized how they could fix those problems. For example, one evaluator noted that the original system image developer did not create upsides or downsides for the problem claims he had developed, suggesting an incomplete analysis of the situational requirements. In addition, two students made the mistake of matching problem claims to the wrong design claims when they first created their system images—but the evaluator's feedback helped expose this breakdown in design rationale development. Two of the participants commented that it would have been helpful during their evaluations to have a separate area where they could make comments on the quality of the claims.

Conclusions

Overall, we can report that the designers and evaluators were able to understand and use the prototype of the Analytic Module as it was envisioned. We were quite pleased at the learning process that the module supported—beyond the mere execution of the analytic method. The pilot testing showed that distributed project work is feasible with the support of at least this module of LINK-UP. We were also pleased to observe a few cases of design knowledge reuse, and we are confident that more cases will be apparent as the tools are more fully integrated in design practice and as the claims library content develops further. The specific results obtained in our testing reveal many directions for the next iteration of prototype development. To summarize the major findings:

- A prototype representation (system image), expressed in terms of problem and design claims provides enough information for knowledgeable HCI persons to quickly evaluate prototype designs.
- The Analytic Module can help designers identify and resolve problems with their prototype designs in terms of design claims and how they address both the upsides and downsides of problem claims.
- The Analytic Module can help students learn about scenario-based design with claims analysis by giving targeted feedback through evaluations that can uncover misconceptions about claims and relationships between problem and design claims.

The results of our online evaluation tool demonstrates its potential for teaching HCI design concepts such as claims analysis and knowledge reuse to students in a distributed and collaborative environment. However, based on participant feedback, a future version of the Analytic Module should separate the process of rating the quality of claims from the process of rating the quality of the design they represent. Furthermore, supporting multiple evaluations for a single system image would strengthen the believability of the ratings. Together, these would allow quality claims by top students in a class to be beacons of use and reuse in future semesters.

To reflect again on the broader vision of LINK-UP, with this module (or with other similar tools), researchers from the arts and social sciences can become involved in distributed education and research efforts related to human-computer interaction. With the tool in place and initial design efforts available for demonstration, we have begun organizing events that inform partners about the opportunities available for collaboration in interface design research and teaching, made possible by LINK-UP. Some of these opportunities include use of the LINK-UP system and design knowledge content in other undergraduate and graduate classes, perhaps to support a special interdisciplinary design and testing activities or research assignments. Other opportunities might involve deeper investigation of specific design approaches (i.e., use of interactive public displays or haptic interfaces) or application areas (i.e., community computing, interfaces for disabled users, or groupware) as appropriate to a campus community, facilitated by web-based access to a repository of claims and testing results.

We also expect researchers from other disciplines to have standard design and testing methods that should be integrated into LINK-UP's tool suite, another direction for follow-on efforts. To promote these types of opportunities, our larger research group will host two structured workshop events, at which we will actively seek interdisciplinary collaborators interested in extending this e-learning solution. Although these initial efforts focus on a subset of the types of designs that can be done, we hope and expect members of the E-Learn community to embrace our methods and apply them to their own disciplines.

References

Carroll, J. M. and Rosson, M. B. (1992) Getting around the task-artifact cycle: how to make claims and design by scenario. *ACM Transactions on Information Systems (TOIS) 10* (2), April 1992, pp. 181-212.

Carroll, J. M. and Rosson, M. B. (1987) Paradox of the active user. *Interfacing though: cognitive aspects of human-computer interaction*, MIT Press, Cambridge, MA, 1987.

Chewar, C. M., Bachetti, E., McCrickard, D. S., and Booker, J. (2004) Automating a design reuse facility with critical parameters: Lessons learned in developing the LINK-UP system. In *Proceedings of the 2004 International Conference on Computer-Aided Design of User Interfaces (CADUI '04)*, Island of Madeira, Portugal, January 2004, pp. 236-247.

Chewar, C. M. and McCrickard, D. S. (2003) Educating notice developers of notification systems: Targeting user-goals with a conceptual framework. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA 2003),* Honolulu, Hawaii, June, 2003, pp. 2759-2766.-

Chewar, C. M., McCrickard, D. S., Sutcliffe, A. G. (2004) Unpacking critical parameters for interface design: Evaluating notification systems with the IRC framework. To appear in Proceedings of the 2004 Conference on Designing Interactive Systems (DIS '04), Cambridge MA, August 2004, 10 pages.

McCrickard, D. S., Chewar, C. M., Somervell, J. P, Ndiwalana, A. (2003) A model for notification systems evaluation—Toward assessing usability for multitasking activity. *Transactions on Computer-Human Interaction (TOCHI)*, Dec 2003, pp. 312-338.

McCrickard, D. S., Czerwinski, M., and Bartram, L. (2003) Introduction: Design and evaluation of notification user interfaces. *International Journal of Human-Computer Studies 8* (5), pp 509-514, May 2003.

Nielsen, J. (1994) Enhancing the explanatory power of usability heuristics. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI '94)*, Boston, MA, pp. 152-158.

Norman, D. A. (1986) Cognitive engineering. In D. A. Norman & S. W. Draper (Eds.) *User Centered Systems Design: New Perspectives on Human-Computer Interaction*, Lawrence Erlbaum Associates, New Jersey, pp. 31-61.

Rosson, M. B. and Carroll, J. M. (2002) *Usability Engineering: Scenario-Based Development of Human-Computer Interaction. Morgan Kaufman*, New York, NY.

Sutcliffe, A. G. and Carroll, J. M. (1999) Designing claims for reuse in interactive systems design. *International Journal of Human-Computer Studies 50*, March 1999, pp. 213-241.