Low-Effort, High-Payoff User Interface Reengineering

Catherine Plaisant, Anne Rose, Ben Shneiderman, and Ajit J. Vanniamparampil, University of Maryland

Using a variety of low-effort, high-payoff strategies on six major software projects, the authors demonstrate that, in some cases, effective user interface reengineering can be accomplished in a matter of weeks instead of the year or more traditionally required.

Substantial user interface research and design experience has given us a deeper understanding of design principles and methodologies.[1-3] Through careful needs analysis and iterative testing, user-centered design methodologies can lead to useful and usable new system designs.[4] Typical development cycles take months for analysis and design and a year for development and testing. However, opportunities are arising for the low-effort, high-payoff reengineering of existing interfaces that will typically require just several weeks to implement.

Reengineering has been defined as the examination and alteration of a system to reconstitute it in a new form, and the subsequent implementation of that new form.[5] Process reengineering requires looking at a business's fundamental processes from a cross-functional perspective.[6] Any mode of reengineering would involve reengineering the business process as well, but we focus only on redesigning the user interface of existing systems. Although business processes may be affected, extensive revamping of the business process itself is beyond the scope of *user interface reengineering*, the term we use to describe the process of redesigning an existing interface to improve user performance and satisfaction while shortening learning time and reducing error rates.

TABLE 1
DIAGNOSTIC STRATEGIES

	Project					
Strategy	DJJ	NLM	HNS	LC	Corabi	CCS
Study the documentation	Moderate	Limited	Extensive	Limited	Moderate	Limited
Attend formal training	One hour		One day			
Discuss with managers	Yes		Yes	Yes	Yes	Yes
Discuss with designers	Yes		Yes		Yes	Yes
Discuss with users	Dozens		One in depth	Limited	Two only	
Observe users	Dozens	Usability test	Limited	Moderate	One day	Usability test
Conduct expert reviews	Yes	Yes	Yes	Yes	Yes	Yes
Administer questionnaire	320 users	19 usability testers		Dozens online		14 usability testers

The challenge for developers is to understand how the old user interface was designed and conceived,[7] and what constraints the new user interface must respect.[8] Case studies of user interface redesign have shown that its benefits for users and companies can be sizable.[9] But these case studies often profile either the redesign of small system components such as an individual screen layout or the thorough and iterative redesign of an entire interface.

For the work reported here, we extracted general reengineering strategies based on our experiences with redesigning six systems. Our case studies and strategy mandate minimal changes to the hardware and software requirements of the current system while emphasizing low-effort, high-payoff improvements. Our strategies constitute a subset of the general user-centered design methodology, adapted to the needs of short-term reengineering.

We provide guidance for managers and designers who are responsible for user interface reengineering by presenting the diagnostic strategy we used in the process, the improvements we identified, and the outcomes for each project. Our approach to each project depended on factors such as the size and complexity of the system, the number and types of users, and the customers' requirements and commitment in terms of time and capital. We hope to encourage others to apply and refine our strategies as they improve existing systems. This seems especially important because many successful systems are somewhat dated and thus present a grand opportunity to improve performance and job satisfaction.

SIX REENGINEERING PROJECTS

In each of the projects we participated in, the staff requested our assistance in reengineering the user interface; thus, they were open to making changes from the outset. Although our project interventions were highly varied, we learned valuable lessons from each. In all cases, we conducted a review using the diagnostic strategies we developed, then proposed short-term recommendations.

Some projects had clear time limits by which we had to complete our recommendations. This forced us to set our priorities and quickly assess what the client would be able and willing to implement. Other limitations included available financial resources, software tools, staff, and the staff's expertise level. Our reports always included assessments of these limitations, and we provided estimates of the effort required—in terms of the number of hours and level of expertise—to implement our recommendations. Project staff appreciated our sensitivity to their constraints.

Our work on short-term recommendations was usually followed by more extensive, long-term explorations of novel interfaces, but those fall outside the scope of this article. The following brief descriptions set the context for our work.

Juvenile Justice (DJJ). In the large Maryland Department of Juvenile Justice project, we evaluated the Information System for Youth Services used by more than 600 case workers. Short-term recommendations included changes that required only a low-level implementation effort and that offered advantages visible to all users. DJJ initiated action on some of the issues we raised, but postponed others to the more complete reengineering implementation, which is taking place currently.

Microanatomy (NLM). At the National Library of Medicine we critiqued and redesigned the interface of the Micro-Anatomy Visual Library System, an interactive computer system that lets library patrons or students view videodisk images of human cell structures.[10]

Network management (HNS). For Hughes Network Systems, we evaluated a complex satellite network configuration system based on numerous overlapping forms. We could not entirely master this complex system within the scope of our project, but did identify several problems and offered short-term recommendations that the project staff found useful.

Library card catalog (LC). We worked with the Library of Congress staff to improve access to the library online catalog for first-time users and to eliminate training classes.[11] The existing command-driven online interface to the catalog, Scorpio, was given a colorful touch screen interface called Access, designed to serve first-time users. Access has reduced the workload of the reference staff at the help desk, letting them help advanced users with complex searches.

Telepathology (Corabi). Our analysis of a remotely controlled microscope developed for pathologists by a small firm, Corabi Telemetrics, identified key issues

such as time delays, incomplete feedback, and interference.[12]

Home automation (CCS). At Custom Command Systems, we evaluated a home automation system for security, lighting, entertainment, and climate control.[13] Home owners used touch screens mounted in the walls or cabinetry to control all the equipment in their house. The redesigned interface was successfully used as a front end for several home automation systems.

DIAGNOSTIC STRATEGIES

Designers can use our diagnostic strategies to learn about and evaluate most user interfaces. The diagnostics' objective is to understand the functioning of the system and identify the key areas where substantial improvements can be achieved by minor interface reengineering.

Table 1 shows the three strategies we used for all projects: documentation study, observation of users, and expert review. We attended the formal training whenever possible and studied the documentation as a way to understand the users' backgrounds. The other techniques consist of discussions that were conducted depending on the availability of managers, designers, or users. Printed surveys or questionnaires complemented the discussions.

> Learning the work culture and adapting to it go a long way toward winning support.

Documentation: *Compile, then review all available documentation.* One of the easier ways to learn about any system is to peruse the available documentation: system specifications, design documents, user manuals, training videos, online help, and so on. An organization's annual report, or other such material that defines its goals and mission, helps you understand the organization's objectives and future direction. The documentation is useful as a reference during the interface reengineering process and may become the object of redesign itself. All projects had some sort of documentation available.

Formal training: *Attend training sessions and demonstrations.* Another diagnostic strategy is to attend the formal training programs and demonstrations with the users. In addition to giving us the same introduction to the system that the users receive, it also gives us valuable insights into the training process itself. Some of the deficiencies in the interface design become apparent at this stage. At DJJ, the youth information system was presented to groups of about 50 new users in one-hour training sessions. As warnings and tips were given to users, we learned the interface's major problems. At HNS, the standard week-long training was condensed into a one-day session that taught us enough to be able to observe users and follow their work.

Discussion with managers: *Identify goals, commitment, and resources.* Creating rapport with top management and gaining their support is essential for the successful implementation of the new design. Discussions with management can help identify the goals of the organization, their commitment to the redesign process, the metrics they use to measure success, the resources available, and the timeframe for implementing the reengineered design. In addition, management's requirements for executive summary reports and other statistical information can be identified at this stage. These discussions also help lay out and prioritize the benefits—like improved productivity and user satisfaction—that could be achieved by redesigning the interface. At LC, the main priority was to relieve librarians from helping patrons with simple queries. At DJJ, the priority was to improve data accuracy. In both cases, this knowledge let us focus attention on the most effective short-term improvements that satisfied the expressed goal.

Discussion with designers: *Identify resources and constraints.* Discussions with the design and maintenance staff help identify their goals, the system constraints, and the alternatives available. In some cases we were unable to contact the designers of the original system, but we always at least interviewed the team that would potentially implement our recommendations. Building a working relationship with the technical staff is essential for the successful implementation of the new interface. If the reengineering work is being done in-house, these discussions can aid in identifying the strengths and weaknesses of the design staff and will influence the rating of the effort level required for each recommendation. Obviously, recommendations for short-term improvements that require unavailable funds or staff are inappropriate.

Discussion with users: *Learn about their frustrations and expectations.* The methods and amount of time spent interviewing users vary greatly across projects. At DJJ, case workers used the system in many different ways and we spent a long time observing users so we could understand the different practices. At HNS and Corabi, the number of users was limited and their time precious, so we interviewed only a few. This can be compensated for by spending more time observing users doing their work.

Discussions with users must be carefully planned. During our informal discussions with DJJ users, identifying ourselves and gaining their confidence was of utmost importance because of the problems with previous design efforts.[14] Learning the work culture and adapting

TABLE 2
OPPORTUNITIES FOR IMPROVEMENT

	Project					
Opportunity	DJJ	NLM	HNS	LC	Corabi	CCS
Documentation	Yes		Yes		Yes	
System access	Yes			Yes	Yes	Yes
Data display	Yes	Yes	Yes	Yes	Yes	Yes
Data entry	Yes			Yes	Yes	Yes
Consistency	Yes	Yes	Yes		Yes	Yes
System/error messages	Yes	Yes	Yes	Yes	Yes	
Additional functions	Yes		Yes		Yes	Yes

to it go a long way toward winning support. Soliciting comments on overall system performance and asking open-ended questions will help identify the problems users face during their daily system use. Some of these problems might have simple solutions. The spontaneous first response from users can be extremely useful to spot needed short-term improvements. For example several Corabi telepathology system users told us that the system was “great, but I don’t use it much because it’s too slow,” which immediately pointed us to an important problem.

Observe users: *Watch users perform their routine tasks.* Observing users using the system gives feedback on the manner in which experts and novices react to different system responses. This proves especially useful when users cannot be interviewed easily. Observation helps identify specific bottlenecks that might be overlooked otherwise. Information about the hardware’s condition, the physical work environment, system response time, and so on can also be gathered this way.

At DJJ, we observed the effect of inoperative equipment and identified the most common handwritten “cheat-notes” used to remember cryptic codes. At HNS, we observed how difficult it was for users standing and talking on the phone to manage many similar-looking overlapping windows, or how often some error messages appeared. Those problems surfaced rapidly but would have been ignored without our direct observation of users. In some cases we conducted usability tests. In the two “zero-training” systems, NLM microanatomy and CCS home automation, more than a dozen users were brought to the lab and given representative tasks to perform.

Expert reviews: *Use the system to gain firsthand experience.* In our role as user interface reengineering experts, we used expert reviews to obtain an in-depth knowledge of the process flow and system procedures. Hands-on experience with the system is invaluable for this purpose. Where possible, the review team should have access to the actual system, although you may have to settle for a training system. We used this diagnostic strategy for all projects and, significantly, it generated the largest number of suggestions for short-term improvements, in particular for consistency and screen layout.

Questionnaire: *Administer questionnaires to get user feedback.* Another valuable tool we used for evaluating existing user interfaces was the Questionnaire for User Interaction Satisfaction.[15] QUIS was developed by the Human-Computer Interaction Laboratory at the University of Maryland. It is used widely in industry and academia for evaluating 71 interface features and can be customized to suit the specific system being evaluated. QUIS uses 1-to-9 scales, and helps identify the major interface problem areas, as perceived by the user.

The questionnaire offers some major advantages over personal interviews. For example,

♦ it can be administered to a larger population,

♦ it maintains anonymity,

♦ it facilitates system comparison with industry standards or with similar systems currently in use, and

♦ it provides for open-ended questions in which users can express their comments and suggestions.

To measure the relative gains of the new interface, questionnaires can be re-administered after the reengineered interface has been installed. We did exactly this on the microanatomy project, administering QUIS before and after the changes were made. The results showed that the revised interface generated higher ratings that were statistically significant for 19 features. A separate test of performance speed confirmed these benefits.

IMPROVEMENT OPPORTUNITIES

For short-term recommendations to be effective, the reengineering team must understand exactly how the organization functions and have a good working relationship with it at all levels. This ensures that when recommendations are presented they will be well received and acted upon. We stressed repeatedly that our role was to make recommendations only and that the responsibility for action lay with management and staff. This approach helped managers and technical staff maintain a sense of project ownership and reduced fears that we might take away some of their control.

A phased implementation of the new design increased the success rate of our intervention. We usually separated our reports into short-term and long-term recommendations for items such as data display, consistency, and system access so that our clients could decide how far to go in implementing them. For example, counters can be used to determine how often each screen is accessed, then the

most frequently accessed screens can have their layouts redesigned first. Providing such positive results quickly lets users see the benefits of the new design, which encourages them to contribute even more to the success of the project. A phased implementation also helps the design team better meet deadlines by setting a schedule of specific targets for each team member.

The six projects we analyzed showed a clear pattern of problems identified and corresponding low-effort, high-payoff opportunities for improvement. Specific problem areas consisted of documentation, system access, data display, data entry, consistency, system/error messages, and additional functions, as shown in Table 2.

Documentation. Three of the six systems lacked proper system documents. The first step is to provide better documentation for those parts of the system that have the most problems. Providing a comprehensive user's manual and a quick-reference guide can help users perform their functions more competently. Having sections of the documentation online—like help, a minitutorial, and a frequently asked questions list—makes it easier to complete tasks. We found this solution applicable to all nonpublic access projects. For example, DJJ users could not easily remember the cryptic codes for juvenile offenses: Discussions with managers and designers had eliminated a short-term move to graphical menus, so we recommended the addition of a rudimentary online help system for the offense codes.

> Even the most refined interfaces we reviewed had many obvious inconsistencies.

System access. We regularly found that system access could be improved by bringing distant equipment closer to the work site, opening frequently locked work rooms, repairing damaged equipment, increasing system speed and reliability, and simplifying access procedures. At DJJ, we recommended reduction in the number of steps to login to the system. The revised interface now has five steps instead of 10, and one password instead of two. The system also issues better warnings when the password is about to expire, making system access much easier. This change was greatly appreciated by all users.

Data display. The screen layout showed improvement potential for all projects using color, sorting and grouping of fields, and highlighting. For example, screens that used only uppercase characters were made more readable by using mixed characters and bolding important data, a simple but effective improvement. Many of the screens we analyzed contained obsolete information and obscure codes that were not useful. At the same time, new elements had been added to existing screens to meet changing needs. This contributed to screen clutter and made information retrieval difficult. We chose to limit recommended changes to a few critical screens.

Data entry. Improving data entry procedures can reduce the number of errors and speed performance, thereby improving user productivity. We looked for instances where the same data was being entered in different locations, which resulted in duplication of work. One way to eliminate redundant data entry is by displaying default information whenever possible. Limiting cursor movement to only editable fields, using a high-precision strategy for touch screen selection, making the cursor more visible, and having consistent key sequences are all measures that can accelerate data entry and reduce errors.

Consistency. This improvement opportunity addresses common action sequences, terms, units, layouts, abbreviations, spelling, capitalization, color, and so on within an application program; consistency also includes compatibility across application programs and with paper or non-computer-based systems.[3] Making the interface more consistent results in faster learning, higher speed, lower error rates, and better retention. In all six projects, we suggested improvements in the consistency of terminology used, sequence of operations, screen headers, field labels, and screen colors. Even the most refined interfaces we reviewed had many obvious inconsistencies. For example, the HNS network management interface used up to four different terms for the same object in the same window.

Error and system messages. Improving error and system messages is one of the easiest and most efficient ways to improve an existing system. More information can be conveyed by making messages more specific and by providing constructive guidance to the user. Using a positive tone and a user-centered style for the messages makes the user more comfortable with the system, especially in difficult situations. Displaying information that is not relevant, like the error code, only adds to screen clutter. Using a consistent format, terminology, color scheme, abbreviation list, and placement increases acceptance of the system. User satisfaction is further enhanced by providing feedback that indicates the changes and status associated with every user action.

Additional functionality. Discussions with users and feedback from the questionnaire let us identify additional functions that could easily be integrated and would greatly enhance user performance and satisfaction. After we consulted with the management and maintenance staff, modified versions of these features were incorporated in the revised design. Often, graphical representation and information visualization techniques can

TABLE 3
OPPORTUNITIES-BY-STRATEGY WITH ESTIMATED EFFECTIVENESS

	Strategy and Estimated Effectiveness							
Opportunity	Administer Questionnaire	Study the Documentation	Attend Formal Training	Discuss with Managers	Discuss with Designers	Discuss with Users	Observe Users	Conduct Expert Reviews
Documentation	Sect. 6.6	X	X				X	X
System access	Part 7					X	X	X
Data display	Part 4		X			X	X	X
Data entry	Part 6		X			X	X	X
Consistency	Part 5							X
Messages	Part 5&6		X				X	X
Additional functionality	Comments			X	X	X	X	
Estimate payoff	All parts			X		X		
Estimate cost				X	X			

be used to present more information more clearly in the same screen area. This helps in analyzing the information and makes information retrieval much easier. In the Corabi telepathology system, we recommended the addition of a slide overview, which provided context for the zoomed view and greatly facilitated navigation. In the home automation system, we identified the scheduling of devices as the most challenging component of the interface and suggested a direct manipulation interface using a timeline and flags. For the online catalog, we suggested additional features to cross-reference subject headings, which simplified navigation.

Summary table. The diversity of possible software development situations precludes us from giving managers simple guidance recommendations. In each of the projects we analyzed, we spent our initial visits determining what the major opportunities were and deciding which strategies to apply. Table 3 shows the eight diagnostic strategies we found effective for generating recommendations in the seven areas of opportunity for improvement. In addition, the bottom two rows include our estimates of which strategies were most effective in assessing payoff and cost. Not surprisingly, discussions with managers gave us the strongest input for these assessments, while the QUIS gave quantitative data about which of the 71 items would produce the greatest improvement.

We stress the importance of good communications among the reengineering team, the designers of the existing system, managers, and users. A clear schedule and explicit statement about the level of effort for the review and reimplementation ensures common understanding of the level and duration of effort as well as when expected payoffs should appear.

Our reengineering review process took from several days to several weeks. The outcome was a written report, usually ranking recommendations by anticipated level of effort and payoff. These recommendations were specific: proposed rewording of messages, new layouts for screens, or mockups of revised navigation routines. We encourage you to be equally specific in your recommendations. Finally, a public presentation of the recommendations supports the discussion process and encourages participation.

Although increasingly sophisticated design methodologies for developing new user interfaces exist,[1,2] low-effort, high-payoff user interface reengineering represents a new direction—and opportunity. Yet reengineering a working system is complex and risky because of the potential disruption to users and managers, their justifiable fear of change, and the lack of guarantees that such changes will be for the better.

Our largely positive experiences with the projects described here lead us to believe that user interface reengineering is a viable and important process. Low-effort, high-payoff improvement recommendations can probably be made for most existing systems.

Nevertheless, a narrowly focused user interface reengineering plan may be inappropriate when the major problems lie outside the scope of the user interface, such as inadequate functionalities, frequent crashes, and network problems. Attempts at improving less severe problems while ignoring deeper ones may be perceived as insensitive by the users. In such cases it is important to consider either making similar short-term improvements for other parts of the systems or postponing short-term user interface reengineering in favor of a more complete system reengineering. Similarly, the need for interface stability might outweigh the benefits of the short-term improvements if a complete reengineering is planned for the near future.

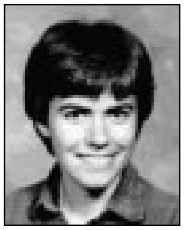
But most likely these proposed diagnostic strategies and opportunities for improvement are only a prelude to the much larger task of business reengineering, which implies extensive user interface reengineering. The increasing sensitivity to usability issues will foster better user involvement in the redesign effort, and the tracking of the short-term benefits will increase the credibility of the redesign team and its chances of success. ◆

ACKNOWLEDGMENTS

We thank the HCIL members who were involved directly or indirectly in the projects listed here and contributed to their successful completion. We also thank users who provided valuable feedback during the reengineering process, and the subjects in the usability testing of our prototypes. The preparation of this document was supported in part by the Maryland Department of Juvenile Services and by NSF grants NSF EEC 94-02384 and IRI-9615534.

REFERENCES

1. D. Hix and H. Hartson, *Developing User Interfaces: Ensuring Usability through Product & Process*, John Wiley & Sons, New York, 1993.
2. J. Nielsen, *Usability Engineering*, Academic Press, New York, 1993.
3. B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 3rd Edition, Addison Wesley, Reading, Mass., 1998, to appear.
4. T.K. Landauer, *The Trouble with Computers: Usefulness, Usability, and Productivity*, MIT Press, Cambridge, Mass., 1995.
5. E.J. Chikofsky and J.H. Cross II, "Reverse Engineering and Design Recovery: A Taxonomy," *IEEE Software*, Jan. 1990, pp. 13-17.
6. M. Hammer, "Reengineering Work: Don't Automate, Obliterate," *Harvard Business Rev.*, July-Aug. 1990, pp. 104-112.
7. *Design Rationale: Concepts, Techniques, and Use*, T. Moran and J. Carroll, eds., Lawrence Erlbaum Assoc., Hillsdale, N.J., 1995.
8. E. Merlo et al., "Reengineering User Interfaces," *IEEE Software*, Jan. 1995, pp. 64-73.
9. T.S. Tullis, "An Evaluation of Alphanumeric, Graphic and Color Information Displays," *Human Factors*, Vol. 23, 1981, pp. 541-550.
10. R. Chimera and B. Shneiderman, "User Interface Consistency: An Evaluation of Original and Revised Interfaces for a Videodisk Library," *Sparks of Innovation in Human-Computer Interaction*, B. Shneiderman, ed., Ablex Publishers, Norwood, N.J., 1993, pp. 259-271.
11. G. Marchionini, M. Ashley, and L. Korzendorfer, "Access at the Library of Congress," *Sparks of Innovation in Human-Computer Interaction*, B. Shneiderman, ed., Ablex Publishers, Norwood, N.J., 1993, pp. 251-258.
12. D. Carr et al., "The Effects of Time Delays on a Telepathology Workstation," *Proc. 16th Symp. Computer Applications in Medical Care*, Baltimore, 1992, pp. 256-260.
13. C. Plaisant, B. Shneiderman, and J. Battaglia, "Scheduling Home-Control Devices: A Case Study of the Transition from the Research Project to a Product," *Sparks of Innovation in Human-Computer Interaction*, B. Shneiderman, ed., Ablex Publishers Norwood, N.J., 1993, pp. 205-215.
14. A. Rose, C. Plaisant, and B. Shneiderman, "Using Ethnographic Methods in User Interface Re-engineering," *Proc. DIS '95: Symp. Designing Interactive Systems*, ACM Press, New York, Aug. 1995, pp. 115-122.
15. J.P. Chin, V.A. Diehl, and K.L. Norman, "Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface," *Proc. CHI '88: Human Factors in Computing Systems*, ACM, New York, 1988, pp. 213-218.



Catherine Plaisant is assistant research scientist at the Human-Computer Interaction Laboratory of the Institute of Advanced Computer Studies at the University of Maryland. Her work centers primarily on graphical user interface design and evaluation, with an emphasis on hypertext, public access information systems, and information visualization.

Plaisant received a general engineering degree from l'Ecole Nationale Supérieure des Arts et Métiers in Paris and a Docteur Ingénieur degree from the Université Pierre et Marie Curie in Paris.

Ben Shneiderman is a professor of computer science, head of the Human-Computer Interaction Laboratory, and a member of the Institute for Systems Research, all at the University of Maryland at College Park. He is the author of *Software Psychology: Human Factors in Computer and Information Systems* (Little Brown, 1980) and *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (third edition, Addison-Wesley, 1997) and the editor of *Sparks of Innovation in Human-Computer Interaction* (Ablex, 1993).

Shneiderman received a PhD in computer science from the University of New York at Stony Brook.

Anne Rose is a faculty research assistant in the Human-Computer Interaction Laboratory at the University of Maryland at College Park. She has worked as a software engineer at the Naval Research Laboratory in Washington, D.C., developing specification tools for real-time systems. Her research interests include software engineering methods and user interface design with an emphasis on information visualization.

Rose received a BS in industrial engineering from Pennsylvania State University.

Ajit Vanniamparampil, operations manager at SkyTel, Washington, D.C., is the business manager of information systems used in the Reseller Distribution Channel of SkyTel, a wireless messaging services provider. He has worked as an instrumentation engineer in the oil exploration field in Singapore and as a quality assurance engineer in the telecommunications manufacturing sector in India.

Vanniamparampil received an MBA in information systems from the University of Maryland and a B.Tech. in electronics and communication engineering from the University of Kerala, India.

Address questions about this article to C. Plaisant, A.V. Williams Building, UMIACS, University of Maryland, College Park, MD 20782 USA; plaisant@cs.umd.edu; http:/www.cs.umd.edu/projects/hcil.