Gerhard Fischer (and Hal Eden, Hiroaki Ogata and Eric Scharff): "Design, Learning, and Collaboration" — Spring Semester 2002

Course Projects

General Objectives

The central purpose of the course project is to gain an in-depth understanding of a theme relevant to the course. While we encourage you to do a project accompanied by an implementation of a new system or the further evolution of an existing system, we will also accept projects that engage in conceptual work accompanied by empirical analysis of existing approaches, systems, and websites. Projects need to be carried out through a learning-by-doing approach throughout the rest of the semester, preferably as a **collaborative** activity.

Schedule

due dates: post information on the respective date on the class website by 10:00am!

1. Monday, Feb 4: Initial Idea of a Course Project
   Discussion of Example Projects in Class by Instructors
   Brief Discussion of Projects Ideas in Class by Students

2. Monday, Feb 25: Project Proposal (including formation of teams);
   Brief Discussion of Projects in Class by Students

3. Wednesday, March 20: Progress Report

4. Monday, April 29: Final Report

Recommendation

To achieve something non-trivial during the semester, we strongly encourage you to work together in a group. You should see the project as an application and opportunity to apply and critically evaluate the themes that we are discussing in the course.

Remark: In the spirit of vertical integration, we would like to encourage experienced graduate students to serve as leaders of teams and approach and undergraduates to join their team.

Requirements for Projects

An Initial Description of your Course Project

Format: one page max or less

Things to Do:

1. Think about what you want to do! why is the problem interesting to YOU?
2. describe your project idea commenting on the following specific issues:
   - 2.1. goal: which problem do you want to address?
   - 2.2. objective: what do you want to achieve?
   - 2.3. means: which media/technologies do you expect to use?
   - 2.4. specific challenges: what do you consider the most challenging aspect of your project?
   - 2.5. relationship to course: in which way is your course project related to the course

Project Proposal

Format: A maximum length of 2 pages

Team Description: Members of the Team, Anticipated emphasis of individual contributions

Content — The proposal must contain the following sections - statement of the problem, rationale, technical approach and implementation. Each section will be graded on appropriateness, completeness and clarity.

1. *Statement of problem-*
   1.1. What is your project all about? Be specific. You should operationalize your terms in order to clarify the problem you are trying to address as well as the approach you will pursue. If appropriate: use literature citations and references to other systems to support your arguments and descriptions.
2. *Rationale -*
   2.1. State the reasons why you want to explore what you are. Why is this a good idea for a project? What do you believe you will learn by doing it? Derive the implications from your project to design, learning, and collaboration.
3. *for non-implementation projects:*
   3.1. develop null hypotheses for the questions you would like to investigate
   3.2. articulate clearly how your work will investigate issues beyond what is already known
4. *for implementation projects:*
   4.1. *Outline and justification of technical approach*
   4.2. *Implementation Plan*
5. *References* — List the key references, other systems, previous projects on which your work will be based.

Progress Report

Format maximum length of 2 pages.

Evaluation: Progress reports will be evaluated like the proposals, based on relevance, appropriateness, completeness and clarity. You will **not** be graded on how closely you adhered to your original plan.

Content — The progress report must contain a description of your progress against your original schedule. If you have changed your plans (based on your work), it must include a clear description of the revisions and arguments for them.

Final Report

Format A maximum length of 6 pages

Evaluation: The final report will be evaluated based on relevance, creativity, appropriateness, completeness, and clarity.

Content — The final report must include the following sections (it is encouraged to extend and reuse arguments from previous reports):

1. *Statement of the Problem* — it describes how your understanding of the problem has changed while you have worked on it over the period of the course
2. *Rationale* — it explains why is the problem interesting or important? Relate it to other systems and the literature! Why should someone else be interested in the problem chosen by you? i.e., tell about the contribution it makes to the knowledge of a community.
3. *Non-Implementation Projects:*
   - 3.1. articulate clearly your contribution
   - 3.2. describe how you advanced the knowledge (e.g., questionnaire, testing of developments, new conceptual framework, empirical data)
4. *Implementation Projects:*
   - 4.1. *Technical approach*
   - 4.2. *Description of the system*
   - *4.3. Description of the system behavior*
   - 4.4. *Evaluation of the program / system*
   - 4.5. *Potential further developments of your program /system*
5. *References* — List the key references, other systems, previous projects on which your work will be based.

Project Ideas for the Course

Capturing and Presenting Relevant Information

In Envisionment and Discovery Collaboratory (EDC) problem solving situations, construction activities and information relevant to the current problem are tightly coupled. One kind of information that would be useful in many domains would be resources from information providers, such as news feeds. Currently, there is no direct or automatic connection between information resources and the EDCs information space. People who find interesting articles must add them to the EDC by hand. In this project, you would explore how news feed or other similar streaming information sources could be used within the EDC. You would explore issues such as the capture, organization, and delivery of information. Relevant resources may include sites such as the Boulder Daily Camera. One example design problem would be deciding how automated the capture and delivery can / should be and what tools would be necessary to facilitate the information gathering process.

More Information:

http://www.cs.colorado.edu/~l3d/systems/EDC/demo/demo10.html

Reading:

Arias, E. G., Eden, H., Fischer, G., Gorman, A., & Scharff, E. (2000) "Transcending the Individual Human Mind—Creating Shared Understanding through Collaborative Design," *ACM Transactions on Computer Human-Interaction*, 7(1), pp. 84-113.

Contact Person: Eric Scharff

Gathering information from Geographic Information Systems

EDC problems typically involve some geographic or spatial component. Both the EDC domains (transportation, flooding) and the representations (maps, diagrams) are well suited for geographic information. There is a wealth of existing data in Geographic Information Systems (GIS) that professionals often use as part of their decision making process. Capturing and using this information could help improve the authenticity of some of the situations discussed in the EDC. Information such as census information, maps, expert models, and other GIS data could be tapped for use in an EDC planning task. In this project, you would look at what information exists in GIS systems and how this information could be used within the EDC. You would explore information such as finding the information that exists and the challenges of extracting and interfacing with existing tools. Relevant resources may include US census data and Boulders GIS resources. One example design challenge would be determining tradeoffs between importing data and having multiple tools used in a query process.

More Information: http://www.esri.com/

Reading:

study the above website carefully!

Contact Person:

Eric Scharff, Deng Ning (Deng.Ning@colorado.edu)

Capturing Feedback from Remote Participants

The EDC relies on a physical construction space. Unfortunately, only a limited number of people can interact with this construction space at the same time. Is it possible to increase the number of participants in the construction process? For large group meetings, technologies such as wireless computers or PDAs may help give a voice to people who are not around the table. In this project, you would explore how new technologies (such as wireless systems) may be used to involve a larger group of face-to-face participants. You would explore different strategies for getting people to participate, and technologies that may support that participation. Relevant resources may include existing work on voting and chat technologies. One example design challenge would be determining what kind of information would be contributed by individuals, and what social situations might be necessary to support large-scale interactions.

More Information:

Reading:

Abowd, G. D. & Mynatt, E. D. (2001) "Charting Past, Present, and Future Research in Ubiquitous Computing." In J. M. Carroll (Ed.) *Human-Computer Interaction in the New Millennium,* ACM Press, New York, pp. 513-535.

Contact Person: Eric Scharff, Shinichi Konomi

Context-Aware Computational Environments — Integrating Artifacts with the Decisions Surrounding them

Our past work centered on domain-oriented design environments has been based on the following simplifying assumption: all design activities happened *inside* the computational environment rather than some of them happening in the external world. With the EDC, we need to extend our approach by creating environments that integrate computational environments and (computationally enriched) external physical worlds with mechanisms capturing the larger (often unarticulated) context of what users are doing. For example: a fundamental shortcoming of the current prototype of the EDC is that there is no capture of the discussions in which stakeholders engage during design sessions. In this project, you would explore and investigate a variety of critical and important research problems, including the capture of design rationale, as it is articulated in the discussions and design sessions by integrating the artifact under construction with the discussions around it. This will address the failure of design rationale systems of the past that required extra efforts of scribes to document in the computational environment things that are happening in the surroundings.

Some of the following research issues can be explored in this project:

1. are context-aware environments most successful if constructed for specific domains because the domain-orientation will restricts the context and provides us with better mechanism to interpret the context?
2. because context-aware environments need to know more about other agents participating in collaborative decision making, will adequately designed "boundary objects" between users and systems be able to provide this

additional context and thereby provide richer and easier interaction?

3. which context elements can be captured *automatically* by devices, usage data, recommender systems, social navigation, read wear and edit wear and which context needs to be *explicitly* provided by humans?
4. how can the efforts and necessary skills be based on the interests and background knowledge of specific communities of practice?
5. Complex collaborative decision making processes are ill-defined problems in which context do not exists but emerge gradually. How do we capture the emergent context?
6. Assuming a substantial amount of context has been captured, how will the context be used to personalize information, and how can push technologies exploit the context to contextualize information to the task at hand?

More Information:

Reading:

Fischer, G. (2001) "Articulating the Task at Hand and Making Information Relevant to It," *Human-Computer Interaction Journal, Special Issue on "Context-Aware Computing"*, 16, pp. 243-256. — available at: http://www.cs.colorado.edu/~gerhard/papers/hci2001.pdf

Contact Person: Hal Eden

Capturing Feedback Between Meetings

The EDC emphasizes bringing people together to discuss problems face to face. However, not all problems can be solved in a single setting, and not every relevant stakeholder can be present at all meetings. Currently, the EDC only uses a simple Web annotation system to support this asynchronous discussion. How can we complement synchronous meetings with other asynchronous information sources? In this project, you would explore what features would be necessary for asynchronous interaction. More specifically, you could look at how Web tools (like discussions, annotation tools, outliners) can capture the results of meetings and how people who are not present can present their opinions. Relevant resources would include some of the sharing and annotation features provided by the Swiki. One example design challenge would be understanding what form user comments should take (such as voting, discussion, and so on) and how to summarize parts of a face-to-face meeting for people not present.

More Information:

Reading:

1. Moran, T. P. & Carroll, J. M. (Eds.) (1996) *Design Rationale: Concepts, Techniques, and Use,* Lawrence Erlbaum Associates, Inc., Hillsdale, NJ
2. Fischer, G., Lemke, A. C., McCall, R., & Morch, A. (1996) "Making Argumentation Serve Design." In T. Moran & J. Carrol (Eds.), *Design Rationale: Concepts, Techniques, and Use,* Lawrence Erlbaum and Associates, Mahwah, NJ, pp. 267-293.

Contact Person: Eric Scharff, Gerhard Fischer

"Virtual Stakeholders" (Critics) and Making Users Feedback Active

In most specific domains, some generally accepted rules emerge. For example in the transportation domain, one such rule might be: *"Two bus stops should not be further apart than 500 yards"*. These rules can be embedded in systems and "critique" (representing the design knowledge of virtual stakeholders) design activities as they take place. Critics are computational entities that can analyze a computer model of a problem and give feedback based on a certain perspective.

People participating in the EDC come to the table with a specific agenda and a set of personal constraints — and these constraints may be different from the critics existing in the system. Capturing people's own specific and additional constraints, helping them make the constraints explicit, and evaluating a design based on user constraints are all very important tasks. In this project, you would explore how a user may express their perspective in an active manner, perhaps in the design of critics. Relevant resources include existing critiquing systems and other active feedback systems (such as spelling correctors.) One example design challenge would be determining the kinds of constraints a user might want to specify in an active way, and whether it is feasible to represent those constraints in an automatic or semi-automatic fashion.

More Information:

Reading:

Nardi, B. A., Miller, J. R., & Wright, D. J. (1998) "Collaborative, Programmable Intelligent Agents," *Communications of the ACM,* 41(3), pp. 96-104.

Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G., & Sumner, T. (1998) "Embedding Critics in Design Environments." In M. T. Maybury & W. Wahlster (Eds.), *Readings in Intelligent User Interfaces,* Morgan Kaufmann, San Francisco, pp. 537-559.

Contact Person: Eric Scharff, Gerhard Fischer

Empirical Study of CodeBroker

One of the major advantages brought by object-oriented programming languages is class libraries that contain hundreds or even thousands of classes and methods that can be reused by programmers. However, mastering such library classes and methods presents huge learning challenges for programmers. It is impossible for programmers to learn all of them before they start programming. A practical approach is for them to learn on demand, namely, to learn a new class or method when it is needed during work. To support learning on demand, we need to re-think the design of the interface of class library systems. Such library system interfaces must be able to make programmers aware of the existence of the classes and methods that they do not yet know but can be reused in their current programming tasks.

CodeBroker is a library system that supports Java programmers in learning classes and methods on demand. As an autonomous software agent, CodeBroker continuously runs in the background of the programming editor--Emacs. It infers the task of programmers by analyzing the doc comments and signatures (syntactic definitions of methods) entered unto the editor, and autonomously locates and proactively delivers classes and methods that can possibly be reused in the current programming task. To ensure that only the classes and methods the targeted programmer does not yet know would be delivered, CodeBroker uses user models to represent each programmer's existing knowledge of the library. Because each programmer's knowledge changes, a user model in CodeBroker updates accordingly as the system learns from analyzing programs written by the user (programmer).

The CodeBroker system has been implemented and initially evaluated. More information, including a one page description and usage scenarios, can be found at http://www.cs.colorado.edu/~yunwen.

The project for this class is to empirically study the usefulness and usability of the CodeBroker system. There are at least two possible studies to be conducted.

3. **Controlled Comparative Study.** This study will involve two groups of programmers. One group will program with the support of CodeBroker, and the other, without. Project members have to recruit voluntary users, design small programming tasks, conduct the experiments and interviews, and analyze the collected data quantitatively and qualitatively. The most challenging part of the project is to design appropriate programming tasks. No programming is required for this project but knowledge of Java is required.

4. **Uncontrolled Long-Term Study.** Project members need to recruit voluntary programmers who program in Java quite often to use the CodeBroker system (runs in Linux). CodeBroker can automatically log most of the interactions between the user and the system. The goal of this study is to understand the impact of CodeBroker on programming in natural settings. Project members will periodically collect and analyze logs from users for quantitative analysis, and interview users to evaluate qualitatively the usefulness of the system.

Findings from those studies will lead to the improvement of the system and better and more efficient ways that programmers learn and reuse library classes and methods.

More Information: http://www.cs.colorado.edu/~yunwen.

Reading:

Ye, Y. and G. Fischer: "Supporting Reuse by Delivering Task-Relevant and Personalized Information", in Proceedings of ICSE'2002: International Conference on Software Engineering, Buenos Aires, Agentinia, May, 2002, [http://www.cs.colorado.edu/~gerhard/papers/1cse2002.pdf]

Contact Person:

Yunwen Ye, Center for LifeLong Learning and Design —email: yunwen@cs.colorado.edu

Empirical Study of High-Functionality Applications (HFAs)

do an empirical study of the opportunities and problems associated with high-functionality applications (e.g., Microsoft Office, Microsoft Word, Linux, Photoshop, EDC, CodeBroker) — this may include: develop a questionnaire and collect empirical data (e.g., at least among the members of our class, maybe the people at your workplace, …), think about techniques to overcome the problems associated with HFAs (e.g., to make them useable and useful, ….)

More Information:

http://www.cs.colorado.edu/~yunwen.

Reading:

Gerhard Fischer" "User Modeling in Human-Computer Interaction", Contribution to the 10th Anniversary Issue of the Journal "User Modeling and User-Adapted Interaction (UMUAI)" Vol. 11, No. 1/2, pp 65-86, 2001; http://www.cs.colorado.edu/~gerhard/papers/umuai2000.pdf

Contact Person: Gerhard Fischer