October 2006

HCI4: Module project

The goals of this exercise is give you some experience in developing your own system from scratch, including conception, building and evaluation. Unlike other assignments emphasis is put on building something that is original, and your design work that takes place before coding the system. Imagine you had just started work at a new startup development company, and they asked you to come up with a new idea and develop it sufficiently to pitch the idea to management. How would you go about doing that?

Overview

The first part of the project is developing you concept for a new system. You can choose whatever you would like to develop, and the concept will be marked on originality. Come up with something that isn't boring. Write quality code, like the excellent software engineers that you all are, but also be original, interesting, creative, imaginative. Impress and surprise us with what you build. We recommend that you draw on the course in particular the lectures on trackers and infrastructures. Context data may be generated and stored by any means you wish. However, you may also use existing usage monitoring tools, such as the User Action Recorder from the GRUMPS project, a keystroke-level monitor for Windows OS. Data can be shared via a shared filespace or kept in a database. The Elvin multicast toolkit is available for publishing data for client use in real time. There are also GRUMPS tools to let you send context data to a sharable SQL Server database.

A lab on 18th October will introduce the trackers and toolkits. Then, there will be workshop sessions (see the Calendar) during which one or more of us will be in the L4 lab, to discuss or advise on the exercise. Here are some examples to show the breadth of possibilities:

- A game of ping pong played across clusters of machines
- A memory tool for bloggers
- A tool that tracks the 'quality' of your online friends by scanning your instant messenger conversations, and builds a social network diagram of who you talk to, how much, and what sort of conversations you have.
- A monitor of 'interruptibility' that uses low-level keyboard events to indicate how much a user can be interrupted. This information is distributed to other users so that they can make decisions about interrupting each other.

Once you have come up with a concept we would like you to make sure that you talk to us to get some feedback. We're going to check that everybody talks to one of us, just to make sure that you get enough feedback on the idea its feasibility. In particular, we want to check that your project is not too ambitious and that you have thought about having something you can program in the time allocated. You can also email us for feedback. However, you should make sure that you send us your proposed concept by the 1st November.

The rest of the term you should work on developing the concept into enough shape so that you can demo it to us at the end of the course. We have set aside a block of time so that you can all demo your project idea. This will be a general 'pitch' of the concept and you should treat this like you would if you were demoing the system to the CEO of a startup company, who might ask you some difficult conceptual and technical questions about the system.

Project marks

Marks for the overall exercise are out of 18, as the exercise forms 18% of the course's assessment. The project will be evaluated based on a demo and a written report. You should make sure that

your concept is something that you will be able to build enough of to demonstrate the concept. That is to say, it'd be nice to build a new type of flying machine, but you're unlikely to be able to build enough of the concept in time to demo (unless, perhaps, you've done aeronautical engineering.) Submissions will be assessed on the grounds of technical quality, originality, awareness of social interaction issues, and aesthetic interest.

The final reports are due 12th January. In the final report your should any extensions to the system done after the demo, include a (very brief) user manual, and give a clear description of what your system was designed to do, what it did, and lessons you've learnt for future work on the concept. Again, make sure that the report is very clear about your concept, why you choose it, and why it's a good idea. At the very least you should justify what you built and why you built it. Discuss the people, programs and information involved in the scenarios of use you are designing for. Discuss and analyse them, based on the material from the course. You could also consider similarities and differences with regard to other systems described in the course, and design issues. Back up your opinions with evidence and analysis.

Deadlines

Discuss concept with Barry Matthew or Phil	**16th October**
Email concept to Barry	**1st November**
Demo	**after 29th Nov**
Final report	**12th January**