USER INTERFACES 2010

Proceedings of the
Eleventh Australasian User Interface Conference (AUIC 2010),
Brisbane, Australia, January 2010

Christof Lutteroth and Paul Calder, Eds.
Table of Contents

Proceedings of the Eleventh Australasian User Interface Conference (AUIC2010),
Brisbane, Australia, January 2010

Preface ...................................................................................................................... vii
Programme Committee ........................................................................................... viii
Organising Committee ............................................................................................. ix
Welcome from the Organising Committee ............................................................... x
CORE - Computing Research & Education ............................................................... xi
ACSW Conferences and the Australian Computer Science
Communications ...................................................................................................... xii
ACSW and AUIC 2010 Sponsors .............................................................................. xiv

Keynote

Revisiting the Human as an Information Processor ................................................. 3
  Andy Cockburn

Contributed Papers

The Effect of User Interface Delay on Thin Client Mobile Games ............................... 7
  Vipul Delwadia, Ian Welch and Stuart Marshall

A Comparative Evaluation of Annotation Software for Grading Programming Assignments .............................. 17
  Beryl Plimmer

Presenting Query Aspects to Support Exploratory Search ..................................... 27
  Mingfang Wu, Andrew Turpin, Simon J. Puglisi, Falk Scholer and James A. Thom

Usability of Navigation Tools for Browsing Genetic Sequences ............................ 37
  Paul Rutherford, Walt Abell, Clare Churcher, Alan McKinnon and John McCallum

Visualisation Tools for Exploring the Uncertainty-Risk Relationship in the Decision-Making Process:
A Preliminary Empirical Evaluation ........................................................................ 47
  Mohammad Daradkeh, Alan McKinnon and Clare Churcher

iAnnotate: Exploring Multi-User Ink Annotation in Web Browsers .......................... 57
  Beryl Plimmer, Samuel Hsiao-Heng Chang, Meghavi Doshi, Laura Laycock and Nilanthi Seneviratne

Life-Sketch - A Framework for Sketch-Based Modelling and Animation of 3D Objects ........................................ 67
  Burkhard Wünsche and Rong Yang

Design and Impressions of a Multi-User Tabletop Interaction Device ..................... 77
  Andrew Cunningham, Ben Close, Bruce Thomas and Peter Hutterer

Graph Drawing Aesthetics in User-Sketched Graph Layouts .................................... 87
  Helen Purchase, Beryl Plimmer, Rosemary Baker and Christopher Pilcher
Identifying Cultural Design Requirements for an Australian Indigenous Website

Reece George, Keith Nesbitt, Patricia Gillard and Michael Donovan

Author Index

vi
Preface

Welcome to Brisbane and the 11th Australasian User Interface Conference, the forum for user interface researchers and practitioners at the Australasian Computer Science Week 2010. AUIC provides an opportunity for user interface researchers to meet with colleagues and with other computer scientists, and aims to strengthen the community of researchers in Australasia.

The papers presented in these proceedings have been rigorously reviewed. Each paper received at least three reviews. Out of 21 submitted papers, 10 papers were selected for presentation and publication. The breadth and quality of the papers reflects the dynamic and innovative Australasian research environment.

We offer our sincere thanks to the people who made this years conference possible: the authors and participants, the program committee members and reviewers, the ACSW organizers, and the Australian Computer Society.

Christof Lutteroth
The University of Auckland

Paul Calder
Flinders University

AUIC 2010 Programme Chairs
January 2010
Programme Committee

Chairs

Christof Lutteroth, The University of Auckland, New Zealand
Paul Calder, Flinders University, Australia

Members

Matt Adcock, CSIRO, Australia
Jason Alexander, University of Canterbury, New Zealand
Mark Apperley, University of Waikato, New Zealand
Sandrine Balbo, The University of Melbourne, Australia
Terence Blackburn, University of South Australia, Australia
Michael Broughton, Defence Science and Technology Organisation, Australia
Paul Calder, Flinders University, Australia Sally Jo Cunningham, University of Waikato, New Zealand
John Grundy, Swinburne University of Technology, Australia
Richard Li, The University of Auckland, New Zealand
Christof Lutteroth, University of Auckland, New Zealand
Stuart Marshall, Victoria University of Wellington, New Zealand
Masood Masoodian, University of Waikato, New Zealand
David Nichols, University of Waikato, New Zealand
Cecile Paris, CSIRO ICT Centre, Australia
Rachel Patel, The University of Auckland, New Zealand
Tim Pattison, Defence Science and Technology Organisation, Australia
Phillip Relf, Defence Science and Technology Organisation, Australia
Christian Mueller-Tomfelde, CSIRO ICT Centre, Australia
Gerald Weber, The University of Auckland, New Zealand
Joanne Zucco, University of South Australia, Australia
Organising Committee

Co-Chairs
Dr. Wayne Kelly
Prof. Mark Looi

Budget and Facilities
Mr. Malcolm Corney

Catering and Booklet
Dr. Diane Corney

Sponsorship and Web
Dr. Tony Sahama

Senior Advisors
Prof. Colin Fidge
Prof. Kerry Raymond

Finance and Travel
Ms. Therese Currell
Ms. Carol Richter

Registration
Mr. Matt Williams

DVD and Signage
Mr. Matthew Brecknell

Satchels and T-shirts
Ms. Donna Teague
Welcome from the Organising Committee

On behalf of the Australasian Computer Science Week 2010 (ACSW2010) Organising Committee, we welcome you to this year’s event hosted by the Queensland University of Technology (QUT). Striving to be a ”University for the Real World” our research and teaching has an applied emphasis. QUT is one of the largest producers of IT graduates in Australia with strong linkages with industry. Our courses and research span an extremely wide range of information technology, everything from traditional computer science, software engineering and information systems, to games and interactive entertainment.

We welcome delegates from over 21 countries, including Australia, New Zealand, USA, Finland, Italy, Japan, China, Brazil, Canada, Germany, Pakistan, Sweden, Austria, Bangladesh, Ireland, Norway, South Africa, Taiwan and Thailand. We trust you will enjoy both the experience of the ACSW 2010 event and also get to explore some of our beautiful city of Brisbane. At Brisbane’s heart, beautifully restored sandstone buildings provide a delightful backdrop to the city’s glass towers. The inner city clusters around the loops of the Brisbane River, connected to leafy, open-skied suburban communities by riverside bikeways. QUT’s Garden’s Point campus, the venue for ACSW 2010, is on the fringe of the city’s botanical gardens and connected by the Goodwill Bridge to the Southbank tourist precinct.

ACSW2009 consists of the following conferences:

- Australasian Computer Science Conference (ACSC) (Chaired by Bernard Mans and Mark Reynolds)
- Australasian Computing Education Conference (ACE) (Chaired by Tony Clear and John Hamer)
- Australasian Database Conference (ADC) (ADC) (Chaired by Heng Tao Shen and Athman Bouguetta)
- Australasian Information Security Conference (AISC) (Chaired by Colin Boyd and Willy Susilo)
- Australasian User Interface Conference (AUIC) (Chaired by Christof Lutteroth and Paul Calder)
- Australasian Symposium on Parallel and Distributed Computing (AusPDC) (Chaired by Jinjun Chen and Rajiv Ranjan)
- Australasian Workshop on Health Informatics and Knowledge Management (HIKM) (Chaired by Anthony Maeder and David Hansen)
- Computing: The Australasian Theory Symposium (CATS) (Chaired by Taso Viglas and Alex Potanin)
- Asia-Pacific Conference on Conceptual Modelling (APCCM) (Chaired by Sebastian Link and Aditya Ghose)
- Australasian Computing Doctoral Consortium (ACDC) (Chaired by David Pearce and Rachel Cardell-Oliver).

The nature of ACSW requires the co-operation of numerous people. We would like to thank all those who have worked to ensure the success of ACSW2010 including the Organising Committee, the Conference Chairs and Programme Committees, our sponsors, the keynote speakers and the delegates. Special thanks to Justin Zobel from CORE and Alex Potanin (co-chair of ACSW2009) for his extensive advice and assistance. If ACSW2010 is run even half as well as ACSW2009 in Wellington then we will have done well.

Dr Wayne Kelly and Professor Mark Looi
Queensland University of Technology
ACSW2010 Co-Chairs
January, 2010
CORE welcomes all delegates to ACSW2010 in Brisbane. CORE, the peak body representing academic computer science in Australia and New Zealand, is responsible for the annual ACSW series of meetings, which are a unique opportunity for our community to network and to discuss research and topics of mutual interest. The original component conferences ACSC, ADC, and CATS, which formed the basis of ACSW in the mid 1990s now share the week with seven other events, which build on the diversity of the Australasian computing community.

In 2010, we have again chosen to feature a small number of plenary speakers from across the discipline: Andy Cockburn, Alon Halevy, and Stephen Kisely. I thank them for their contributions to ACSW2010. I also thank the keynote speakers invited to some of the individual conferences. The efforts of the conference chairs and their program committees have led to strong programs in all the conferences again, thanks. And thanks are particularly due to Wayne Kelly and his colleagues for organising what promises to be a strong event.

In Australia, 2009 saw, for the first time in some years, an increase in the number of students choosing to study IT, and a welcome if small number of new academic appointments. Also welcome is the news that university and research funding is set to rise from 2011-12. However, it continues to be the case that per-place funding for computer science students has fallen relative to that of other physical and mathematical sciences, and, while bodies such as the Australian Council of Deans of ICT seek ways to increase student interest in the area, more is needed to ensure the growth of our discipline.

During 2009, CORE continued to work on journal and conference rankings. A key aim is now to maintain the rankings, which are widely used overseas as well as in Australia. Management of the rankings is a challenging process that needs to balance competing special interests as well as addressing the interests of the community as a whole. ACSW2010 includes a forum on rankings to discuss this process. Also in 2009 CORE proposed a standard for the undergraduate Computer Science curriculum, with the intention that it be used for accreditation of degrees in computer science.

COREs existence is due to the support of the member departments in Australia and New Zealand, and I thank them for their ongoing contributions, in commitment and in financial support. Finally, I am grateful to all those who gave their time to CORE in 2009; in particular, I thank Gill Dobbie, Jenny Edwards, Alan Fekete, Tom Gedeon, Leon Sterling, and the members of the executive and of the curriculum and ranking committees.

Justin Zobel
President, CORE
January, 2010
The Australasian Computer Science Week of conferences has been running in some form continuously since 1978. This makes it one of the longest running conferences in computer science. The proceedings of the week have been published as the *Australian Computer Science Communications* since 1979 (with the 1978 proceedings often referred to as Volume 0). Thus the sequence number of the Australasian Computer Science Conference is always one greater than the volume of the Communications. Below is a list of the conferences, their locations and hosts.

2011. Volume 33. Host and Venue - Curtin University of Technology, Perth, WA.

2010. **Volume 32. Host and Venue - Queensland University of Technology, Brisbane, QLD.**


2008. Volume 30. Host and Venue - University of Wollongong, NSW.

2007. Volume 29. Host and Venue - University of Ballarat, VIC. First running of HDKM.

2006. Volume 28. Host and Venue - University of Tasmania, TAS.


1998. Volume 20. Hosts - University of Western Australia, Murdoch University, Edith Cowan University and Curtin University. Venue - Perth, WA.


1995. Volume 17. Hosts - Flinders University, University of Adelaide and University of South Australia. Venue - Glenelg, SA.


1990. Volume 12. Host and Venue - Monash University, Melbourne, VIC. Joined by Database and Information Systems Conference which in 1992 became ADC (which stayed with ACSW) and ACIS (which now operates independently).

1989. Volume 11. Host and Venue - University of Wollongong, NSW.


1987. Volume 9. Host and Venue - Deakin University, VIC.

1986. Volume 8. Host and Venue - Australian National University, Canberra, ACT.


1983. Volume 5. Host and Venue - University of Sydney, NSW.

1982. Volume 4. Host and Venue - University of Western Australia, WA.

1981. Volume 3. Host and Venue - University of Queensland, QLD.

1980. Volume 2. Host and Venue - Australian National University, Canberra, ACT.

1979. Volume 1. Host and Venue - University of Tasmania, TAS.

1978. Volume 0. Host and Venue - University of New South Wales, NSW.
Conference Acronyms

ACDC  Australasian Computing Doctoral Consortium
ACE  Australasian Computer Education Conference
ACSC  Australasian Computer Science Conference
ACSW  Australasian Computer Science Week
ADC  Australasian Database Conference
AISC  Australasian Information Security Conference
AUIC  Australasian User Interface Conference
APCCM  Asia-Pacific Conference on Conceptual Modelling
AusPDC  Australasian Symposium on Parallel and Distributed Computing (replaces AusGrid)
CATS  Computing: Australasian Theory Symposium
HIKM  Australasian Workshop on Health Informatics and Knowledge Management

Note that various name changes have occurred, which have been indicated in the Conference Acronyms sections in respective CRPIT volumes.
ACSW and AUIC 2010 Sponsors

We wish to thank the following sponsors for their contribution towards this conference.

**CORE - Computing Research and Education,**
www.core.edu.au

**Queensland University of Technology,**
www.qut.edu.au

**Australian Computer Society,**
www.acs.org.au

**CEED,**
www.corptech.com.au

**CSIRO ICT Centre,**
www.csiro.au/org/ict.html

**SAP Research,**
www.sap.com/about/company/research
Keynote:
Revisiting the Human as an Information Processor

Andy Cockburn
Department of Computer Science and Software Engineering
University of Canterbury
Private Bag 4800, Christchurch 8140, New Zealand
andy@cosc.canterbury.ac.nz

Abstract
In their seminal 1983 book ‘The Psychology of Human–Computer Interaction’, Card, Newell and Moran (ACM Fellows and Turing Award winners) introduced ‘the model human information-processor’. This model equipped interface designers with strong theoretical tools to predict human interface performance without the demands of implementation and evaluation.

In the quarter century since then, human–computer interaction research has been extremely successful in transitioning research ideas to commercial deployment, yet interface design remains something of an art that is dependent on time–consuming iterations of design, implement, and evaluate. Theoretical models of human performance are rarely used despite their potential.

In this presentation I will describe several of our recent projects seeking to improve the efficiency of everyday activities in computer use, including scrolling, text messaging, window switching, and navigating through menu and file structures. The overriding theme, however, is on using theoretical human performance models to inform design, explain and predict performance, and to generalise results obtained. Ultimately, the objective is to give all computer science graduates the equivalent of a ‘Big O’ complexity theory for user interfaces that allows them to design with assurance.
CONTRIBUTED PAPERS
The Effect of User Interface Delay in Thin Client Mobile Games

Vipul Delwadia  
Stuart Marshall  
Ian Welch

1 School of Engineering and Computer Science  
Victoria University of Wellington,  
PO Box 600, Wellington 6140,  
Email: vipul.delwadia | stuart.marshall | ian.welch @ecs.vuw.ac.nz

Abstract

Thin-client computing may be a solution to such problems as providing sophisticated applications on devices with low computational power, or providing reasonable access to digital artifacts whose distribution the copyright owner still wishes to protect. However, certain application domains have tight constraints around user interface response times, and the network aspect of thin-client computing may cause issues in this regard. We have conducted an experiment to identify how various delays added to mobile games affect players' performance and perceptions of the gameplay. By studying the effects of these delays, we aim to identify time-based performance parameters within which our future thin-client computing systems should work if they are to support all application domains.

Keywords: (H.5.2) User Interfaces, (H.3.4.b) Performance Evaluation

1 Introduction

Portable devices such as netbooks and mobile phones are getting closer and closer to becoming our primary (and occasionally only) computing device. Despite these devices' increasing computational capability and network capacity, it is still challenging to provide certain classes of applications that require significant memory or computation. One such example application domain is that of games. Mobile devices are still typically underpowered compared to standard desktops widely available in the consumer market.

Remotely providing applications on a central server via thin client computing is a potential solution, not only for typical mobile games on a mobile device but also for playing fully-fledged, resource-intensive games which cannot be played on a mobile device currently.

Thin client computing has been reborn as cloud computing. Applications such as office suites, photo editing and presentation tools are moving from running locally to running in the cloud with access via web browsers. This offers several advantages over running applications locally (Buyya et al. 2009, Miller 2008, Hayes 2008, Armbrust et al. 2009) including reduced costs, reliability, scalability and security, and perhaps most importantly, mobility.

Additionally, it may also offer a compromise solution to the problem of protecting copyrighted digital artifacts (such as the executable binaries of historical and present-day applications) while still affording access to users. Thin clients for single games already exist. For example, FreeRealms (Sony Online Entertainment 2009) and QuakeLive (id Software, Inc 2009) are played within a web browser with a platform-specific plugin installed. Recently there has been interest in developing more general thin clients that can be used to access a range of games hosted on a server. For example, OnLive (OnLive 2009) or Games@Large (Jurgelionis et al. 2009) are thin clients used to access a range of games and aim to lessen the hardware requirements for running a game by offloading activities such as 3D graphics rendering to a server that is accessed over broadband.

While these services provide the functional capability, importantly the performance requirements for accessing and modifying remote applications via local user interfaces are not fully understood.

Hosting games on a server and displaying them locally has the downside that it inevitably introduces latency that will affect users' response times. This latency is the time between user actions and the system's response, e.g. the time between pressing a button and the result of pressing the button happening on screen. Here, the response time is the actual, measured time rather than the theoretical maximum. Too much latency and an application such as a game may be unplayable making all the other benefits of the thin client approach moot.

1.1 Contributions

The main contribution of this paper is the carrying out of usability experiments to determine whether latency has a negative effect upon game play and to determine a latency benchmark for the particular types of games that we used for our experiments. We carried out the experiments into the effects of latency by inserting delays into the response time of two different games running on a mobile device. We measured player scores, level reached (for Asteroid Zone (Doue 2008) as Bomber 2 (Yank 2008)), the other game, only has three levels), whether users reported whether they noticed a delay or not and user experience as whether they were satisfied or not satisfied with the game play experience. We used score and level as a way to measure performance. Each user played the games with 0, 75, 150 and 300 ms delay. We have established that (note that we have used a 95% level of significance for all statistical tests):

- For Asteroids, a 2D single player shooter style game, that there is a significant negative correlation between performance (as measured by average score) and delay. Also that there is a significant negative correlation between self-reported player experience and delay, and a significant
positive correlation between user perception of delay and actual delay. We have also determined that user performance is similar with delays of 75 ms and 150 ms but negatively affected for 300 ms delay.

- For Bomber 2, a 2D side scrolling style game that there is apparently no significant correlation between self-reported player experience and delay, and a significant positive correlation between user perception of delay and actual delay. Interestingly, there is a significant positive correlation between performance and delay at the 75 ms delay level although there is not a significant correlation between performance and delay for 150 ms and 300 ms. This suggests that players actually do better at Bomber when there is a small delay!

Establishing what latency and delays are acceptable to users is important. Knowing this will allow developers of such systems to assess whether their implementations meet performance requirements. Subsequent to the very early work on acceptable response time for control systems done in the 1960s, more recent work has been done for multiplayer games where locally running games interact with game servers to allow multiple players to participate in a campaign or shared environment. However, the conclusions of these studies do not directly apply to our own interest in remote gaming because they are executing game code on the client as well as the server. We are working with a definition of remote gaming where the entire game engine and algorithms execute solely on the server. In this case, the client’s only purpose is to display updates from the server and relay user input to the server. Additionally some of the effects they observe may be due to not only the latency of the connection to the game server but also the latency of the connection of other players to the same game server.

1.2 Paper Overview

We conclude that although more work is required to tease out all the contributions to good performance, positive player experience and minimisation of noticeable delay, that our system should aim for a latency of 75 ms or better in order to be acceptable to users of 2D single player shooter and 2D side scroller games.

The rest of this paper is structured as follows: Section 2 provides an overview of the conclusions of work into measuring the effects of latency for multiplayer games. Section 3 described the conduct of the experiment. Section 4 overviews the results of the experiment. Section 5 discusses the results. Finally, Section 6 concludes the paper and provide some ideas for future work.

2 Related Work

In this section we look at a number of studies towards setting a benchmark for responsiveness, starting with Miller’s work on response times between users and computers (Miller 1968).

Miller proposed a number of different minimum response times corresponding to various activities based on his opinion as a behavioural scientist. (Miller 1968). Among the activities presented, the lowest (and most relevant) response time proposed is 0.1 seconds, i.e. 100 ms, for the response to control activation such as the click of a typewriter key.

While Miller’s work is important in establishing a guideline very early in the computing industry, studies more recently have looked at the effects of latency on user performance in games. In these studies, the latency is essentially the time difference between the server and client’s version of the current status of the objects in the game. One study (Sheldon et al. 2003) looked at the effect in the real-time strategy game Warcraft III (Blizzard Entertainment 2009). In this case they found that for long and semi-automatic tasks, such as building, there was no correlation between latency and performance. For more involved tasks, mainly combat, the correlation coefficient was 0.01 for unit scores, and the coefficient 0.07 for proportions of games won by a client affected by latencies ranging from 0–1600ms. While this is promising, it is difficult to be conclusive about the results as their study had a limited number of test subjects (2–4).

Another study (Beigbeder et al. 2004) looked at the effect of both loss and latency on a first person shooter, Unreal Tournament 2003 (Epic Games, Inc 2009). Here they found that generally packet loss or latency has no noticeable impact on a player’s performance. However, the one area where it is noticeable is in precision shooting, where the correlation between induced latency and hit fraction is 0.93, and the mean hit accuracy suddenly drops approximately 35%. While this study had a significant amount of data (over 200 experiments), this specific test only comprised of 2 test subjects doing a 10 minute experiment 3 times each.

Both of these studies relate to multiplayer games where a central server only coordinates game play between game programs running on players’ desktops. It is uncertain whether the results can be extended to our situation where the client does minimal processing and the game executes on a central server. Also, these studies did not establish any guidelines for what might be acceptable latency for users.

We have previously conducted a pilot study (Delwadia et al. 2009a) centered around our own RemoteME (Delwadia et al. 2009b) thin-client system. The purpose of the pilot study was to undertake a preliminary exploration of the effect of inserting delays (on top of RemoteME’s in-built delay due to the execution time of the system’s own code) on the test subject’s scores and the perceived impact on their playing experience. In this study the delay is inserted such that the time between the user’s actions and the system’s response is varied, simulating the situation in a remote control system. This study, however, only involved two test subjects, and there were very few data points, so the results were inconclusive. While there potentially was correlation between delay and the subjects’ scores, the experimental issues meant that the results are not as strong as we would have liked. This paper builds upon our experiences gained in our pilot study experiment and attempts to derive statistically significant results.

3 Experiment

In this section we discuss our experiment into how test subjects using software systems are affected by the inclusion of delays into the user-action/system-response cycle.

We chose two 2D action games for our experiment because such games require constant input and frequently update the display. Choosing these games for our experiment will magnify the effects of latency. Further, games by nature are competitive, and so any delay has a negative effect on the player’s “performance”. We decided that performance could be ap-
proximated by recording player score and (in the case of one of the two games) the level reached.

We simulated network latency by artificially adding delays between the user’s actions and the system’s response, and measure the effect it has on the participants by asking them to rate the delay, as well as recording their score in the game and their accuracy. The inserted delay is hereinafter referred to as simply delay. The delay is set to one of the three values 75ms, 150ms or 300ms, and each value is repeated three times, for a total of nine games. Each game has a randomised order of the delays to control for learning effects. The games were implemented for a mobile phone (Nokia N95).

3.1 Hypotheses

The main purpose of our experiment was to accept or reject the following hypotheses:

H0 There is no correlation between a player’s performance in terms of score or level reached and delay.

H1 There is no correlation between a player’s perception of delay or player experience.

Note that as Bomber has only three possible levels whereas Asteroid has five, we decided to restrict H0 for Bomber to measuring score rather than level.

In addition, to attempt to identify an acceptable range of delays. We defined the following hypotheses:

H2 There is no difference in average performance in terms of score between no delay and 75ms delay.

H3 There is no difference in average performance in terms of score between no delay and 150ms.

H4 There is no difference in average performance in terms of score between no delay and 300ms.

Testing each of these hypotheses allow us to determine the delay levels that offer no degradation in performance with respect to our baseline delay of 0ms.

3.2 Design

We chose a within-subjects design for our experiment to attempt to control for differences in skill between players. The experiment consisted of fourteen participants playing two games a set number of times in two sessions. The two sessions were separated by at least ten days. Each player played games of Asteroid Zone and Bomber 2 natively on the phone.

In the first session each participant plays each game eight times. The games themselves are unmodified, and are running natively on the mobile phone. The players were accurately told that there would be delay in the games. The purpose was to allow users to become familiar with each game. In the second session, the participants play each game (with the delay) nine times. After each play through of each game, the participants were asked whether they noticed a delay and if it had a significant impact on their gameplay.

Across the sessions the subjects’ scores were recorded, along with other game play data such as accuracy. Using this data we performed some analysis to see if there were correlations between the delays; decreases in the players’ scores (and accuracies); and the players’ perceptions of the noticeability of the delay and its impact on their playing experience.

3.3 Artifacts

Both the system artifacts in our experiment were shoot’em up games. This was deliberately chosen as these games require quick responses. Both games were relatively simplistic examples of their domain, and this was also a deliberate decision as the incorporated delays would form a significant portion of the overall computation time. Both games were also freely available under open source licenses, making modifications for the purposes of our experiment easy.

Asteroid Zone

The first application is Asteroid Zone, a simple shoot’em up space game. It consists of simplistic graphics and narrative, Asteroid Zone is a real-time game in the sense that noticeable lags between user input and the in-game response can detract from the game’s playability.

Figure 1 shows a sample screenshot of the game. The game draws the ship and asteroids using the basic drawing methods associated with a Canvas object, rather than using special images.

The player’s ship is controlled using a few keys: 4 and 6 to rotate left and right, 2 to accelerate and 5 to shoot. There is no deceleration, so once the player starts moving it is difficult to come to a stand still. When any object reaches the an edge it appears on the opposite edge of the screen, with the same momentum
Figure 2: An action shot of the other game used in the user study, Bomber 2. This game features modern graphics and complicated real-time game play, including enemies shooting back. These elements will be made more difficult when the user's input is delayed.

and trajectory.

Bomber 2

The second application is Bomber 2, a side-scrolling action game. The graphics are more advanced than Asteroid Zone, with a scrolling background, animations for explosions and decoration like trees and clouds. The objective is to destroy the targets (marked with yellow triangles) by shooting or bombing them with your fighter plane. Once all the targets are gone the level is complete and the player moves on to the next level, with a different layout and arrangement of enemies and targets. If the player’s ship is killed by enemies or collision, the player returns to the start of the map with a fresh set of bombs. The player has five lives per level, and six bombs per life. Lives are reset once a level is complete.

Figure 2 shows a sample screenshot of the game. This game draws the various objects using images loaded into memory, and the objects are drawn using a GameCanvas object.

Controls for Bomber 2 are limited to 4 buttons: 2 to rotate clockwise, 8 to rotate anti-clockwise, 5 to launch a bomb and either 1 or 3 to fire bullets.

Game Modifications

In order to reliably set the delay between the user actions and the system’s response, modifications were made to both games to provide this functionality. The strategy is to suppress every user action for the desired period of time, and then trigger the action once that period has elapsed. In this study we set the period of time - the delay - to be one of three values: 75ms, 150ms or 300ms.

For both games the implementation is the same, a secondary thread keeps a queue of the user actions along with the new time they should be fired at, and approximately every 20 milliseconds the thread triggers actions in the queue that are due. Whenever a user presses a key it is added to the queue along with the time that it is due to be triggered. This is illustrated in Figure 3.

In addition to providing delay functionality, logging capabilities were added to each game. For both games all user input is logged, i.e movement and combat. On top of this, Asteroid Zone now tracks when an asteroid is hit and logs this, as well as when the player dies. Once the game is quit, the log is saved to a memory card installed in the phone, which includes the actions, as well as the final score, the number of shots fired and landed, and value that the delay was set to. Bomber 2 tracks when each type of ammunition is fired and when it hits. Further, the type of object hit is recorded, both in terms of target/non-target and house/tank/etc.

These game modifications were possible to be made because both games were open source with freely available code - one of the reasons for choosing these games over other J2ME games.

Mobile Phone

All the test subjects used the same Nokia N95 mobile phone during the experiment, shown in figure 4. The Nokia N95 is two years old and represents a reasonably common yet sophisticated mobile phone currently on the market. The phone has 64 MB of RAM and a 320x240 pixel display.

The Nokia N95 has a 12-key keypad that can be hidden within the device and slid out as required. The phone also has additional softkeys and a four-way directional control surrounding a central button. Test subjects used the keypad with the appropriate number keys as described previously.
Figure 4: The Nokia N95 mobile phone, with sliding keypad exposed. The Nokia N95 has a 12-key keypad, with additional softkeys and a four-way directional control surrounding a central button. This image was retrieved from the Wikimedia Commons and is licensed under the GNU Free Documentation License by the image’s author: Asim18.

Test Subjects

The test subjects were varied in age, and were mostly staff and students of the School of Engineering and Computer Science at Victoria University. There were 11 males and 3 females for a total of 14 participants. We did not ask if any of the subjects had used that particular mobile phone before, or if they had played either of the games before. Overall all the participants were familiar with computers, and all have a computer science background.

3.4 Procedure

Each test subject played each game eight times in the first session and nine in the second session. Each session was separated by a period of at least ten days, so that the subjects didn’t improve their playing ability significantly between the sessions. The subjects were allowed to play a couple of practice games before each session and for each game, these were not delayed at all. This was to allow the subjects to (re-)acquaint themselves with the controls and gameplay, and to prevent their first games from suffering. Each game was played following this sequence of actions:

1. The experiment controller looked up the delay value to be set for this playthrough, and set the game’s delay to that value. If it was the first session, then the delay was set to 0 ms. The delay values are pre-generated, and each delay level is repeated three times. The order of delays is randomized for each subject.

2. The controller then completes the initialization for the game, and then passes the phone to the subject. The subject then begins the game by pressing the designated button.

3. Once the playthrough is complete (either by the subject exhausting all their lives or finishing the final level to be played), the subject passes the phone back to the controller.

4. If it was the second session (with delay), the controller asked the subject to rate the following statements from 1 – 5, where 1 is strongly disagree and 5 is strongly agree:
   - There was a noticeable delay between my actions and the system’s response.
   - The delay between my actions and the system’s response negatively affected the playing experience.

5. If there were remaining games to play, the test subject and controller return to step one.

4 Results

Here we present the results of the user study. The main calculations are correlations between our independent variable (delay) and our dependent variables where the dependent variables are are the user’s answers to the two questions asked, as well as their score.

4.1 Asteroid Zone

Taking all the test subject’s data together, the Pearson correlation for the delay vs score is $r = -0.201$ (H0) (see Figure 5). This is a slightly negative correlation, but it is not as strong as we expected. However, it is statistically significant at a 95% level. Taking delay vs level the correlation coefficient is $-0.217$, and is graphed in Figure 6.

If we take the scores for each delay value we get mean scores and student’s t-Test values as given in Table 1. We accept the null hypothesis for the first two t-Tests, 75 ms vs 0 ms (H2) and 150 ms vs 0 ms (H3), because the t-Test value is above 0.05, but we reject the null hypothesis for 300 ms vs 0 ms (H4). This indicates that while delays of 75 ms and 150 ms may be acceptable, a 300 ms delay for Asteroid Zone gives a lower mean than no delay, and so it negatively impacts the player’s experience.

For the two questions which were asked, “there was a noticeable delay between my actions and the system’s response” and “the delay between my actions and the system’s response negatively affected the playing experience” (H1), Tables 2 and 3 show the frequency of each response (1–5) given the delay.

<table>
<thead>
<tr>
<th>Delay (ms)</th>
<th>Mean Score</th>
<th>SD</th>
<th>df</th>
<th>t-Test (p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>484</td>
<td>229</td>
<td>124</td>
<td>n/a</td>
</tr>
<tr>
<td>75</td>
<td>549</td>
<td>253</td>
<td>40</td>
<td>0.128</td>
</tr>
<tr>
<td>150</td>
<td>475</td>
<td>227</td>
<td>41</td>
<td>0.825</td>
</tr>
<tr>
<td>300</td>
<td>347</td>
<td>219</td>
<td>41</td>
<td>0.001</td>
</tr>
</tbody>
</table>

Table 1: Mean scores and t-Tests (vs no delay) for Asteroid Zone at each delay level

<table>
<thead>
<tr>
<th>Delay (ms)</th>
<th>User Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
<td>25 13 1 2 0</td>
</tr>
<tr>
<td>150</td>
<td>16 17 5 2 1</td>
</tr>
<tr>
<td>300</td>
<td>6 8 14 8 5</td>
</tr>
</tbody>
</table>

Table 2: Frequencies of user response values at each delay for question one, Asteroid Zone

<table>
<thead>
<tr>
<th>Delay (ms)</th>
<th>User Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
<td>30 7 3 0 1</td>
</tr>
<tr>
<td>150</td>
<td>22 13 4 1 1</td>
</tr>
<tr>
<td>300</td>
<td>7 14 14 6 0</td>
</tr>
</tbody>
</table>

Table 3: Frequencies of user response values at each delay for question two, Asteroid Zone
Table 4: Mean scores and t-Tests (vs no delay) for Bomber 2 at each delay level

<table>
<thead>
<tr>
<th>Delay (ms)</th>
<th>Mean Score</th>
<th>SD</th>
<th>df</th>
<th>t-Test (p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>313</td>
<td>92</td>
<td>108</td>
<td>n/a</td>
</tr>
<tr>
<td>75</td>
<td>346</td>
<td>89</td>
<td>41</td>
<td>0.047</td>
</tr>
<tr>
<td>150</td>
<td>325</td>
<td>99</td>
<td>41</td>
<td>0.487</td>
</tr>
<tr>
<td>300</td>
<td>317</td>
<td>83</td>
<td>41</td>
<td>0.789</td>
</tr>
</tbody>
</table>

Table 5: Frequencies of user response values at each delay for question one, Bomber 2

<table>
<thead>
<tr>
<th>Delay (ms)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
<td>11</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>25</td>
<td>12</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>300</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 6: Frequencies of user response values at each delay for question two, Bomber 2

<table>
<thead>
<tr>
<th>Delay (ms)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
<td>38</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>34</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>300</td>
<td>22</td>
<td>13</td>
<td>5</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Here we calculated Spearman’s rank correlation coefficient to be 0.557 for the first question, and 0.540 for the second. These are quite high given the number of data points, and so we can say that there is a strong correlation.

4.2 Bomber 2

Bomber 2 has weak results. Again, if we take all the test subject’s data, the correlation coefficient for the delay vs score is \( r = 0.017 \) (H0). A coefficient this low suggests that there is no correlation at all, and it is not statistically significant. The mean scores and t-Tests for each delay level (vs no delay) are given in Table 4.

When all delays are considered we accept the null hypothesis H0 for Bomber. However, we also find that when we explore the other hypotheses that the null hypothesis is rejected for the lowest delay value, 75 ms (H2), but accepted for the two higher values (H3 and H4). This is a highly unusual result, somehow suggesting that delaying the game by 75 ms may give a better score than not at all! It could be that players are overcompensating at the higher levels or that introducing delay into the game slightly makes it actually easier to play. Note that examination of the Asteroid results seems to suggest a similar pattern, however the rejection of the null hypothesis suggests that the average higher score at 75 ms is not statistically significant.

As for the questions asked of the users, the frequencies are shown in Tables 5 and 6. Again we calculated Spearman’s rank correlation coefficient, resulting in 0.480 and 0.518 for the first and second questions respectively (H1). While these are significant values, they are not as large as for Asteroid Zone. Further, looking at the frequency tables we can see that while the users on the whole gave higher ranks to the larger delays for the first question (“there was a noticeable delay . . .”), they did not give as high values for the second question (“the delay . . . negatively impacted my playing experience”) for the higher delays. Perhaps the gameplay for Bomber 2 is such that the users can compensate for delay, to the point where it doesn’t affect their gameplay or affects it very little compared to Asteroid Zone.

4.3 Experimental Issues

There were several design issues discovered during the experiment.

The first issue is that the participants were much better at the game than anticipated, and so we limited the total playtime to the natural “end” of the play through. Specifically, for Asteroid Zone, rather than waiting for the player to exhaust all their lives, we stopped the game after the end of level 5, the “last” level in the game. For Bomber 2, we stopped the game after the end of level 3 (the “last” level is level 4). Even with these adjustments, the sessions lasted for an unreasonable long time (over four hours), and so we reduced the number of plays of each game from 16 to 8, for both Bomber 2 and Asteroid Zone, bringing the majority of sessions down to under two hours.

The second issue is an extra command in Asteroid Zone, teleport, that places the player’s ship in a new, random position on the screen. While this seems like a useful command, it results in a noticeable delay when used. There is no limit on the number of times you can use teleport, nor is there a minimum time between uses. Furthermore, it makes no guarantees on the placement of the ship and the player may end up next to or in an asteroid. For these reasons we avoided telling the user about teleport unless they asked, at which point we advised the user not to use the command. Overall, teleport was pressed 18 times out of a total of over 200,000 commands, thus we left the data with teleports in.

The third issue was somewhat beyond our control. The manner in which Bomber 2 is typically played results in periods where the player doesn’t need to press any buttons. The mobile phone is configured to dim the display after a period of inactivity, and so the display dims while the game is still being played. When this occurred during the user’s play, they were advised to press an unbound key to wake the display without affecting the game.

5 Discussion

The aim of this user study was to determine the point at which the delay became unacceptable. From the results, we can say that for Asteroid Zone, while 75 ms and 150 ms may be acceptable delays, at 300 ms the user’s experience is negatively affected. The specific point is not known, but it is assumed to be between 150 ms and 300 ms delay. One user, while playing at 150 ms, said that “any delay under this is good”, and while playing at 300 ms, “if it’s this delay I’d stop playing”. Another user commented that “delay is a lot harder when you’re moving around”.

For Bomber 2, we can’t make any judgements on which delays are unacceptable. The results were inconclusive, suggesting that some delay (75 ms) is better than no delay! It may be the case that more data is required for Bomber 2 at the higher end, with delays greater than 300 ms. On the other hand Bomber 2’s game play may be such that large amounts of delay are easier to compensate for, but they still affect the player’s experience negatively. One user commented that “this is more noticeable than asteroids”, however another said that the delay on Bomber 2 was “way easier to compensate” and that it “any delay is acceptable”.
5.1 Observations

There were other interesting results from the data which we didn’t run any advanced analysis on, such as the average playing time for each delay level. For Asteroid Zone, the average run time is 284 seconds with no delay, and this consistently decreases as delay increases until 436 seconds at a 300 ms delay. For Bomber 2 the numbers are quite different, starting with no delay, and this consistently decreases as delay increases until 434 seconds for 75 ms, 150 ms and 343 seconds for 75 ms, 150 ms and 300 ms delays respectively.

We also performed some analysis on sequences of actions just for Asteroid Zone, based on a user’s comment that we should “look for overturning, where I go past the point and then turn back”.

Consider a sequence of actions: right right right. With no delay, the probability of the next action being right is 0.672, and left is 0.328. At 300 ms delay, the probability that the next action is right is 0.762, a slight increase, however the probability that the next action is left doubles to 0.070. Going the other way, where the sequence of actions is left left left and the next action is left or right, the probabilities barely change between no delay and 300 ms delay. These numbers suggest that while users may adjust their game play when delay is increased, it is not consistent between sequences.

6 Conclusion

We are interested in identifying how delays built into the user-action/system-response cycle affect user performance and perception. In this paper we have presented the results of an experiment we conducted with 14 participants and 2 mobile computer games. Based on the results we conclude that a remote thin-client system will need to have a latency of 75 ms or better to be acceptable for 2D single player shooter and side scroller games. This result has implications for the subsequent design of our thin-client systems as it provides a measure against which to assess timing performance for our system’s various subcomponents.

6.1 Future Directions

There are several ways to advance this work. One limitation of the study is that all participants had some background in computer science. It would be interesting to see whether the results also hold for other demographics. There are also application domains other than games to explore (and also other types of games, such as those requiring extensive use of 3D graphics libraries), and a more refined experiment would also include a finer-grained range of delays around the 75 ms mark.

References


Proc. 11th Australasian User Interface Conference (AUIC2010), Brisbane, Australia
Figure 5: A scatter chart of the scores vs delays for all the test subjects for Asteroid Zone.

Figure 6: A bubble chart for Asteroid Zone, showing the level vs delay for all subjects, where the size of the bubble represents the proportion of data at that level.
Figure 7: A scatter chart of the scores vs delays for all the test subjects for Bomber.
A Comparative Evaluation of Annotation Software for Grading Programming Assignments

Beryl Plimmer
Department of Computer Science
University of Auckland
Private Bag 92019, Auckland 1142, New Zealand
beryl@cs.auckland.ac.nz

Abstract

Commenting on a student’s computer program with red pen ink annotations is not possible with current software and paper program ‘listings’ are a relic of a bygone era. Yet ink annotations are the easiest way to provide rich feedback to the student. We have developed and evaluated Penmarked as a software solution to this problem. It supports free-form ink annotations and, importantly, associated marking tasks of gathering and returning assignments and recording grades. The evaluation against paper and digital marking systems showed it to be faster and more effective. From a wider perspective Penmarked demonstrates the intricacies of providing totally paperless environment.

Keywords: Digital Ink Annotation, task support, assignment grading.

1 Introduction

Annotating student assignments with a red pen is a basic recording mechanism for teachers. The annotations provide commentary to the student on the marker’s response to the work and also provide ‘backtalk’ (Goldschmidt 1999) to the marker to assist with grading decisions. Annotating paper scripts is simple, however we are increasingly moving to digital environments where there is no paper copy of the work. In these environments students submit their work into a digital drop-box, the marker reads and marks it from this repository and returns feedback and grades to the student electronically. Digital alternatives to paper annotation include a marking schedule which incorporates space for off-document comments or digital annotation tools to provide in-situ comments. Either of these alternatives presents problems to the marker and student. Off-document comments are slow to construct and understand, and usually restricted to text. Digital annotation is available in some software applications (such as adobe acrobat) but interfacing these general tools effectively to student management systems and incorporating grade recording is difficult. Furthermore computer program code presents extra challenges because of its non-linear structure and multi-file nature.

There is a conflict here, ink annotations are the traditional and, we believe, easiest way to provide rich feedback to students. However electronic submission systems make ink annotation of paper copies difficult to support. There are many, most locally developed, systems to support off document comments as a part of assignment marking (our department has at least 3). But writing off document comments is tedious and therefore often neglected. As a consequence of these difficulties students are missing out on personalized, meaningful feedback on a critical part of their learning experience.

Computer programs pose some unique requirements for annotation and marking support. First, the non-linear nature of programs means it makes no sense to read from the beginning to the end of a program (as one does an essay). Second, as a part of the evaluation process most markers (teachers or teaching assistants) compile and execute the program: to do this they must work with a digital copy of the assignment. Hence electronic submission of programming assignments has become the norm. Digital drop-boxes have the added advantages of supporting diverse teaching modalities; in-class, distance education, e-learning etc.

Usable and affordable pen input devices, in particular tablet PCs, mean that we now have the hardware to support input of ink annotations directly onto the document surface. Here we present Penmarked, a software solution to this problem, and its comprehensive evaluation from the teachers, markers and students perspectives.

2 Motivation

Consider a TA 50 marking programming assignments. They need to check each program against the assignment requirements, provide feedback to the student, record the grade in the administration system and send the comments and grade to the student. The most difficult part of this task with current systems is providing feedback to the student, so it is often ignored or poorly done. Yet personalized feedback is highly valued by students.

The design focus of Penmarked is to provide an efficient and effective environment for annotating and marking of computer programming assignments. In this paper we start with the educational case for good student feedback mechanisms before reviewing existing work on ink annotation and assignment annotation tools. This is followed by a description of the design and implementation of Penmarked. Then the evaluation methodology and results are described. In the discussion
section we reflect on the efficacy of our approach and lessons learnt including the requirements for program marking compared with the more general needs of paperless environments. The conclusions summarize our project and suggest directions for future work.

For clarity we include a short glossary here: assessment, we use to mean a class set of student assignments; assignment, is an individual student’s work; marking schedule is a table of marking criteria and the mark allocated to each criterion, others may refer to this as a marking scheme or rubric (see Figure 2 for example).

### 3 Related Work

Many organizations are currently attempting to reduce or eliminate paper (Sellen and Harper 2002). Computer programs are, perhaps, the best example of a document type that has evolved to be best suited to digital environments. The folders of paper copies of program listings that existed in all programming shops have been replaced by on-line libraries as paper copies of large object oriented programs with their multiple classes and files make no sense. Programming integrated development environments (IDEs) and supporting tools are designed specifically to support program document management and other programming tasks. However they do not support assignment marking.

Educationalists agree that active learning, where the student is required to do something, is more effective that passive learning, where the student simply observes. In programming classes students are routinely set formative and summative programming tasks to actively engage them in learning correct programming techniques. To complete the feedback loop the work is reviewed and, if it is a summative assignment, a grade allocated. The reviewer’s comments and grade assist the student to reflect on what they have done right and wrong and thus gain a better understanding of the subject. Thus the teacher and students together complete the learning cycle (Figure 1).

![Kolb’s Learning Cycle](image)

**Figure 1 Kolb’s Learning Cycle (from Greenaway 2003)**

The marker’s red ink annotations have traditionally decorated students’ paper assignments. Ink annotations are rich and expressive due to their free-format, (Marshall 1997; Marshall, Price et al. 2001). Ink annotations also have a part to play in supporting active reading (Schilit, Golovchinsky et al. 1998). One of the purposes of making annotations is to help the reader engage more deeply in the material. These annotations also direct subsequent readers to points of interest (Shipman, Price et al. 2003). There is a well understood, yet informal, code for shared annotations such as ticks, crosses and enclosing loops with or without comments or marks attached and ideograms such as ☺. Studies comparing comprehension of annotated and unannotated documents suggest that annotation is an effective aid to learning (Shipman, Price et al. 2003). From these studies we can conclude that annotation is useful for the marker to support his/her active reading of the document, and to the student to direct him/her to parts of the assignment that the marker thought were noteworthy.

One of the first tools developed to explore digital ink annotation was Wang Freestyle (Francik 1995). It provides the user with simple free-form ink annotation over a static page. XLibris (Schilit, Golovchinsky et al. 1998) was developed to offer users an active reading experience, with a main goal of addressing the tangibility challenges of reading online documents. It provides users with an interface and features similar to that of paper. While some word processors support ink comments and change tracking these environments are clumsy for multi-file programs. None of these general tools are supportive of our marker with 50 programming assignments to review and grade.

There are many automatic program marking programs, these are good for marking algorithm type problems where there is one correct answer. However many programming assignments include elements of user interface design and are designed to challenge the student to choose between alternative ‘good solutions’, making them unsuitable for automated marking.

We have identified three related annotation or marking tools. Marktool (Heinrich, Wang et al. 2003; Heinrich and Lawn 2004) supports annotation of assignments by use of drag-and-drop shapes and text boxes. Gild (Myers, Hargreaves et al. 2004) provides marking functionality within the Eclipse IDE but does not support digital ink or text annotation. Alongside this project we have explored annotation inside IDEs (Priest and Plimmer 2006; Chen and Plimmer 2007; Chang, Chen et al. 2008), we were unable to build a functioning clear annotation layer in either visual studio or Eclipse. Our current solution is to copy the document to be annotated and placed it as a background to the annotation pane; however this is not a very robust solution.

Digital ink annotation is technically challenging. The basic requirement is to collect ink input, then display and hold it in the correct place on the related document while the document is repositioned or resized. If the underlying document is dynamic then support for reflowing and reshaping of digital ink as the underlying layout changes is required. Margin bars, circles and underlines, must stretch or shrink with layout changes through font resizing, zooming or varying device characteristics (Golovchinsky and Denoue 2002). Annotations should also reshape when underlying text splits over line breaks and page breaks (Ramachandran and Kashi 2003). In this application the students’ documents are assumed to be finished therefore ink reflow is not required. However, even without reflow support, digital ink annotation continues to be technically challenging because the ink must exist in a separate layer to the original document.
and standard interface components do not support this notion.

4 Penmarked Software

The goal of Penmarked is to fully support the marking and annotating of students’ assignments. It is written in C# for the Microsoft Tablet OS and relies on the inbuilt recognition engine for writing recognition. Here we describe its pertinent features, more technical descriptions are available elsewhere (Plimmer and Mason 2004; Mason and Plimmer 2005; Plimmer and Mason 2006). The software described here is the third prototype of Penmarked. The first (Plimmer and Mason 2004) solved the technical ink annotation challenges and supported basic collection and return of work. As a result of our usability testing (Mason and Plimmer 2005) a number of small, but important, interaction changes were made to the system. Feedback from users has resulted in additional functionality, such as the comment pane. The version reported in (Plimmer and Mason 2006) was used for the evaluation study reported here.

We use a scenario of a teacher with a class set of 50 programming assignments to mark to describe Penmarked. Before starting the teacher creates a new master marking schedule for the assessment. A wizard (Figure 2) is available from the edit menu to support its construction or alternatively the xml file can be edited directly. Another setup step is to decide which files are to be displayed in the annotation panel. Full flexibility based on file names and extensions with wild cards to include or exclude is in the options menu. This is particularly useful for some programming environments that include a number of management files that are not of interest to the teacher.

![Marking Schedule Construction Wizard](image)

**Figure 2 Marking Schedule Construction Wizard.**

The final pre-marking step is to load the students’ files. From the file menu the user can set up or edit an assessment (a class set of marking). The assessment data includes identification and date information, the marking schedule and folder locations of the student assignments.

The student assignments can then be opened for marking. Penmarked parses the location for the student files specified above looking for student data files. These files must contain the student identifiers and the names and types of the associated student’s assignment files. Such files are common in drop boxes or student management systems but each has its own format. Currently we support two formats a simple text file and xml file; however Penmarked provides add-in functionality to extend the file types and structures for these files. All the student assignments found in the specified directory structures are listed in the left pane shown in Figure 3. The teacher can then commence marking by tapping on a student identifier in the list pane on the left of Figure 3.

On the first opening of a student’s assignment Penmarked checks for submitted zip files. If one is found a subdirectory named with the students identifier is created below the location of the zip file and the contents of the zip file are placed into this subdirectory. An individual copy of the marking schedule is also created for the newly opened student assignment and displayed in the bottom window pane as shown in Figure 3. On subsequent opening the unzipping is not required and the student’s personal marking schedule is displayed including any marks already entered.

The annotation pane displays all the files that meet the filter specification, each in a separate tab. The marker can now peruse the assignment. In the annotation pane the marker can ink and erase freely over the assignment. The ink is placed on a transparent layer that lies above a rich textbox, thus the text is inaccessible to the marker. Making the text inaccessible was a design decision we made after talking with students and teachers, it protects the student’s work from unintentional changes by the marker. The implementation challenges with this were many and varied, most difficult was scrolling both layers synchronously as it is difficult to access the appropriate scrolling methods on the lower layer (this is now simpler with Microsoft Presentation Foundation).

If the marker wants to run the program they can directly access the folder containing the student’s files by clicking on the folder icon on the tool bar.

As marking progresses the teacher can enter marks against an item in the schedule by first selecting the item row with a tap or click and then either write the mark into the box in the right bottom corner of Figure 3 or enter the mark via a keyboard. If ink is entered in the writing box it is recognized as soon as another item in the schedule is selected or after a short time delay. The OS recognition engine using the number factoid restriction (which limits recognition results to digits and numeric symbols) is used for recognition. The data is validated against the minimum and maximum values. Valid data is saved into the schedule; the box flashes red if recognition fails to produce a valid result. In our various trials two users experienced problems with recognition errors: one who was more accustomed to writing Chinese formed his ‘5’ more like an ‘s’, a little training solved this problem; the other placed decimal points quite high between numerals ‘5.6’ this is interpreted as a subtract symbol.

If the teacher wants to add some general comments a small comments pane can be opened on the right-hand side of the main window. In addition to the three main
panes the icons initiate frequently used functions of open, save, export, ink and erase.

When marking of a student’s assignment is completed the associated student item can be ticked in the list. A right click on a student item in the list pane displays a menu of less frequent tasks. There is access to functionality such as adding a file to the student’s assignment or cleaning all the annotations from the assignment. There are also options to mark the item as ‘in progress’ or ‘recheck’, these options change the colour of the list item to blue and red respectively.

Figure 4 Main Penmarked interface

Once all the students’ assignments have been marked there is a number of post-marking task required to return the assignments to the students and export the marks. There is a batch process to returned annotated pdf copies of the assignments to students. The marker simply ensures all assignments to be returned are selected in the list and selects ‘return pdfs’ from the file menu. Penmarked generates a pdf for each student that contains their marking schedule, general comments and annotated assignment and emails it to the student. A pdf can be generated independently of emailing. Similarly the detailed and total marks can be exported to an xml file, from which they can be imported into most standard software applications and student management systems.

Thus the marking task is completed. We contend that for efficiency marking software must support the entire marking process: gathering up the assignments, supporting examining, annotating and grading each assignment; returning the work to students and filing the grades. All in a time effective and easy-to-use manner.

The tasks supported by Penmarked for marking an assessment are:

1. Set up a marking schedule
2. Set up an assessment task
3. Gather up assignments
4. Examination, annotation and grading of assignments
5. Return work to students
6. Export of grades

During development Penmarked has been usability tested (Mason and Plimmer 2005) and trialled on a range of assessments including .Net programs, Java programs and essays. To evaluate the efficacy of Penmarked we
conducted a large evaluation study. The details of the study and results are described below.

5 Method

Our goal with the evaluation study was to consider the efficacy of Penmarked from the point-of-view of teachers, markers and students. We conducted a comparative study across three assessments of a second year programming class, applying three different marking treatments to each assessment in a Latin squares arrangement. Quantitative data was gathered from assignment marking while qualitative data was garnered from students, markers and teachers.

The three marking treatments are: traditional paper based marking, on-line marking where marks are recorded in a database system, and Penmarked. We will refer to these treatments as paper, database and Penmarked. The class consisted of approximately 200 students so approximately 600 individual pieces of work were marked. Six teaching assistants (TAs) marked one sixth of the assignments for each assessment (~33). Each student had one piece of work marked via each treatment and each TA marked using each treatment. The student/TA allocations were changed with each assessment so that a TA only marked one piece of work from each student. Therefore the study has marking method as the independent variable and, it is a between-subjects study and each TA and student marked, or had marked, a different assessment for each treatment.

The hypotheses to be tested were

- that there would be a difference in time required to mark using the different methods – with a corresponding null hypothesis that there would be no time difference.
- that the range of grades was consistent across all treatments;
- that there would be a difference in the number of annotation or comments between the in treatments
- that there would be a difference in marker and student satisfaction between treatments.

The assessments varied in difficulty and length: the number of program classes (which equates to files) and lines of code in the model answers are, respectively 1/74, 3/198 and 5/435. As per our usual practice, for each assignment the markers were provided with a model answer, marking schedule and participated in two discussions about the marking (one before any marking and one after 3-4 assignments had been marked). Many students take considerably more lines of code to write a program than the model answers, sometimes twice as much. The paper copies of the assessments 1-3 were respectively approximately 2-3, 10-12, 15-25 pages. For the paper treatment the markers used the paper copy along with the digital copy of the assignment and the IDE for marking. For the database treatment the markers were interacting with the database forms, a digital copy of the assignment and the IDE. For treatment three, Penmarked, the markers use the digital copy of the assignment with Penmarked and the IDE.

Different information was returned to students for each treatment. For paper they received the annotated paper copy of the assignment, and a paper, freehand completed marking schedule. For the database treatment they were emailed a simple list of the marking schedule with their mark for each item and any text comments alongside the item that the marker had entered into the database. For the Penmarked treatment the student received an email with an annotated pdf attached. The first page of the pdf showed the completed marking schedule followed by the digital ink annotated assignment. In all cases the completed marking schedule was returned and the students’ total mark was available on the student management system.

We collected the following data for each assignment: assessment number, treatment, marker, marking time, number and types of annotations, grades, and the number of marking appeals/complaints. After the three assignments had been marked we surveyed marker’s opinions, and student’s opinions.

The marking times were recorded by each marker as they marked. TAs are paid by the hour for marking based on an estimate of the time required for marking. We agreed with the TAs on a fixed, generous, number of hours before the study commenced to negate time pressures on them for the task. An adapted annotation categorization system from (Marshall 1997) was used to categorize annotations as either tick or cross, comment, grade or other. Grades were taken from marked assignments. The TAs’ and teachers’ opinions were gathered through semi-structure interviews and student opinions were garnered from an on-line survey. Student complaints and appeals were recorded and matched back to the assessment number, treatment and marker.

6 Results

Table 1 summarizes the results of the quantitative data on marking times, grades and annotations. We ran a series of statistical tests against this data to identify significant variations. First we compared each set of grades using a one-way ANOVA, these showed no significant differences between markers or groups for each assessment (p .298, p.327, p .265).

Analysis of the marking times showed significant differences for all three assessments. Comparing individual markers there was one marker who was significantly slower through all three treatments. However, disregarding this summary statistics showed that paper marking was significantly slower than the other treatments for the first assessment (<p .001). With assessment 2 there was no significant time difference between the paper and database marking but a significant difference between these two and Penmarked (p .002). With the third assessment Tukey HSD test showed there are significant differences between each treatment (p values between .03 and .004). Individual differences between markers account for some of this difference, however it is clear that Penmarked was consistently faster for markers than either of the other treatments.

Similar tests were conducted on the total number of annotations and number of annotated assignments. The differences were significant in all cases (p < .01) except
the first assessment where there was no significant difference between the paper and Penmarked treatments. Notably for the first assessment while 100% of the assignments marked with the paper and Penmarked treatments had annotations less than 20% of the database treatment had any individual comments and indeed our analysis of these showed most of the comments added to the database were simply ‘not implemented’ for one marker, or ‘not there’ for the other. Similar simple comments were evident in the database for the second and third assessments. Our suspicion is that these were pasted in by the markers.

We had not anticipated looking at where the annotations were on the documents however, with assessment 3 when analysing the paper copies it became obvious that all the annotations were on the front or back page – absolutely none were placed in-situ. The Penmarked annotations displayed some similarities with some annotations placed at the top of a class rather that close to the procedure they were related to. Comparing the total number of annotations and assignments with annotations across the assignment we note a general decline in the annotations between the assessments. A number of reasons could contribute to this, for example the students are less likely to make syntactic or layout mistakes. Or it could be as the programs get bigger it is more difficult to identify the critical parts of the code for the ink annotation, particularly with in the paper treatment.

<table>
<thead>
<tr>
<th>Assessment</th>
<th>Treatment</th>
<th>number of assignment</th>
<th>Time</th>
<th>Grades</th>
<th>Annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>number of assignment</td>
<td>mean</td>
<td>std</td>
<td>mean</td>
</tr>
<tr>
<td>1</td>
<td>Paper</td>
<td>60</td>
<td>20.9</td>
<td>10.6</td>
<td>78.3</td>
</tr>
<tr>
<td></td>
<td>Database</td>
<td>58</td>
<td>10.3</td>
<td>3.2</td>
<td>82.2</td>
</tr>
<tr>
<td></td>
<td>Penmarked</td>
<td>58</td>
<td>9.3</td>
<td>4.4</td>
<td>82.3</td>
</tr>
<tr>
<td>2</td>
<td>Paper</td>
<td>56</td>
<td>22.3</td>
<td>8.3</td>
<td>71.9</td>
</tr>
<tr>
<td></td>
<td>Database</td>
<td>51</td>
<td>21.0</td>
<td>6.9</td>
<td>61.9</td>
</tr>
<tr>
<td></td>
<td>Penmarked</td>
<td>56</td>
<td>16.1</td>
<td>4.3</td>
<td>64.4</td>
</tr>
<tr>
<td>3</td>
<td>Paper</td>
<td>53</td>
<td>21.4</td>
<td>3.8</td>
<td>79.5</td>
</tr>
<tr>
<td></td>
<td>Database</td>
<td>58</td>
<td>26.7</td>
<td>12.6</td>
<td>78.1</td>
</tr>
<tr>
<td></td>
<td>Penmarked</td>
<td>52</td>
<td>18.6</td>
<td>4.7</td>
<td>82.9</td>
</tr>
</tbody>
</table>

Table 1 Summary of quantitative marking data

We asked the TAs to rank the treatments on preference, speed and accuracy. They all ranked Penmarked first, database second and paper third. Their comments supported this very strongly. They appreciated the work-flow support Penmarked afforded telling us that the start-up and close down time for each assignment was considerably less with Penmarked. They particularly commented on the one tap access to the source files from the folder icon. They also compared annotating in Penmarked favourably with annotating the paper; the eraser in Penmarked was the winner here! As they tended to mark at home or when they had gaps between their own class commitments they found carrying around a bundle of paper a nuisance. Four of the markers had used the database treatment when marking for the same course the previous semester and two of these had used another similar tool. Again they all expressed a preference for Penmarked, commenting that it was more natural and easier to comment directly on the assignments. Two express concerns about their spelling and would have like the ability to spell-check their handwritten comments.

The teachers also made complementary comments about Penmarked. While we had recorded grading appeals, they often answered ad hoc questions about marking. They found they were getting less questions from the annotated assignments. Another benefit of Penmarked, that we had not considered is that the teacher had a complete digital replica of what was returned to the student. They found this useful for producing copies of marked work for course review or external moderation. One also commented that she had, on a couple of occasions had students change a handwritten grade on their work and then claim it had been added/recorded incorrectly. Having a digital replica would stop this type of dishonestly.
<table>
<thead>
<tr>
<th>Question</th>
<th>Treatment</th>
<th>Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>I found the submission easy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paper</td>
<td>Agree</td>
<td>Neutral</td>
</tr>
<tr>
<td>Database</td>
<td>6</td>
<td>14</td>
</tr>
<tr>
<td>Penmarked</td>
<td>23</td>
<td>13</td>
</tr>
<tr>
<td>I found the return of work easy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paper</td>
<td>Agree</td>
<td>Neutral</td>
</tr>
<tr>
<td>Database</td>
<td>12</td>
<td>9</td>
</tr>
<tr>
<td>Penmarked</td>
<td>16</td>
<td>12</td>
</tr>
<tr>
<td>I understood where I had lost or gained marks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paper</td>
<td>Mostly</td>
<td>Some-times</td>
</tr>
<tr>
<td>Database</td>
<td>19</td>
<td>7</td>
</tr>
<tr>
<td>Penmarked</td>
<td>18</td>
<td>15</td>
</tr>
<tr>
<td>The feedback helped my learning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paper</td>
<td>A lot</td>
<td>A bit</td>
</tr>
<tr>
<td>Database</td>
<td>7</td>
<td>16</td>
</tr>
<tr>
<td>Penmarked</td>
<td>10</td>
<td>21</td>
</tr>
<tr>
<td>From the feedback I know how to correct my mistakes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paper</td>
<td>Completely</td>
<td>A bit</td>
</tr>
<tr>
<td>Database</td>
<td>8</td>
<td>15</td>
</tr>
<tr>
<td>Penmarked</td>
<td>7</td>
<td>21</td>
</tr>
<tr>
<td>I prefer to get my assignments marked this way</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paper</td>
<td>Completely</td>
<td>A bit</td>
</tr>
<tr>
<td>Database</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>Penmarked</td>
<td>17</td>
<td>13</td>
</tr>
</tbody>
</table>

Table 2 Student Survey

We also ran a voluntary on-line survey for the students. Only 39 of the 200 students responded, not enough from which to draw any firm conclusions. A summary of the questions and responses is shown in Table 2. The first part of the questionnaire was to find out what they actually did with the assignments. The first two questions on the convenience of the different methods we added because of their grumblings about the paper treatment. If this sample is representative of the class it is clear from their responses that they disliked having to hand-in and collect paper with over 30% of them not collecting the marked assignment. This was consistent with a large pile of uncollected paper copies at the end of the course. They received the other marking feedback as an email attachment. Notably almost all claim to have read the comments on the paper or pdf, but only about half read the database comments.

As the purpose of feedback is to aid learning, if they had not looked at the markers comments the feedback would not have contributed to their learning. The next set of questions was to elicit their opinion on the contribution to learning. Because of the way the data was collected we could not exclude answers from the respondents that had not collected their paper from this set of answers.

They said they understood where they had lost and gained marks better with feedback on the Penmarked treatment and considered that it had helped their learning more. However there was no difference between any treatments on their ability to correct mistakes.

<table>
<thead>
<tr>
<th>Assessment</th>
<th>Treatment</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paper</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Database</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Penmarked</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Table 3 Number of Complaints and Appeals

The final data we collected was the number of complaints and appeals for remarking. The numbers are small so no firm conclusions can be drawn from them. However it does reinforce the data from the student survey with them being more satisfied or better understanding the marks allocated when the assignment has been annotated.

The hypothesis that marking time would not increase with Penmarked was proved, in fact Penmarked was
shown to be the fastest marking treatment. Grades remained consistent across all treatments; we did not take any extraordinary measure to ensure this consistency so feel confident that this is a valid result. We found that Penmarked did encourage markers to comment more on the students work in all cases except when comparing Penmarked to the paper copies of the first assignment. The difference in this case is that the paper copies are only 1-2 pages. The data from the markers, students and teachers is qualitative and suggested increased satisfaction from all parties. This is supported by the decrease in complaints and appeals for the annotated assignments.

7 Discussion
Our initial goal with this project was to create a software environment for marking programs that would support red ink annotation in a paper like manner. At the same time we did not want to adversely affect the work processes around marking. From a practical perspective we did not want to increase the time required for marking assignments. Nor could the software skew the grades in any way. The primary goal was to increase feedback and learning for the students, the aims were to increase their learning by increasing understanding of how they had gained or lost marks and how they could perform better, while at the same time increase their satisfaction with the grades they received.

Penmarked has been developed over three iterations. The experience we have gained from repeated use of the system ourselves and by other teachers and through the more formal usability studies resulted in interesting changes to the system. The area that has undergone the most change is the student list. Initially it was a simple list, we then added check boxes and later a right-click menu and colour coding for incomplete assignments or those to be revisited. We conclude from this that the work-flow support that this list provides is essential to the success of Penmarked.

In contrast the only alteration made to the annotation pane is the addition of a ‘find’ function. Users have suggested syntax highlighting and a clipboard for comments. The suggestion for ‘find’ and syntax highlighting lead to our investigations into implementing similar functionality into an IDEs {Chang, 2008 #514}. To date we have had limited success with IDEs: Visual Studio 2010 may make this easier and is worthy of investigation. The clipboard for comments is, at first glance, an obvious extension. However there are significant technical and user challenges to pasting ink, it is not as easy to reflow into a space as text. Our suspicion is that it would take longer to find and place the clipboard comment than to rewrite it. We also worry that a clipboard would encourage lazy generic comments evident in the database treatment in the study above. As educationalists we see little value in this type of comment.

We undertook some informal usability tests on the marking schedule when it was first developed to set screen sizes and time delays. It has remained unchanged since then. Clearly there are many other types of marking scales used; A, B, C …; Likert scales, Excellent – Unsatisfactory; and so on. Supporting other scales may need some redesign of the interface but we would not anticipate any great difficulties.

The responses from students and reduced complaints suggest that annotating their work is worthwhile. We would have like to have seen their confidence to correct mistakes increase. This is an area that could do with further investigation.

The negative comments from the TAs and students about paper suggest to us that for this generation paper and program code are incompatible. Both parties were more comfortable with the electronic systems. It should be noted that this is a programming course; hence we would expect the students and TAs to be at ease with computers. Using this system with a liberal arts class may elicit a different response.

Reflecting on the project ourselves the most important lessons we have learnt have been about the requirements for supporting the entire activity. In our institute paper copies of programming assignments have not been used for many years. Yet we needed to go back to understand the role of paper documents in marking assignments (Marshall 2003) in order to provide the equivalent functionality in a computer system. At the same time it was essential that we provided an easy interface to the existing institutional systems that support student management.

Most of the functionality of Penmarked is available in other software tools, annotation is available in commercial packages such as Microsoft Word or Adobe Acrobat, marking databases are easy to construct and exist in many forms. The display of program code is better in IDEs than that provided in Penmarked. The essential benefit of Penmarked is the bringing together of these different functions in a manner specifically designed to support program marking.

8 Conclusions
Penmarked is a specific example of a wider problem: How to translate successful paper-based techniques to a paperless system without compromising best practice. We have demonstrated that Penmarked is more time efficient and produces better results than either a paper-based system or a marking database. At the same time student understanding and satisfaction increased. The success of this project, we believe, is due in main to our rigorous efforts to support the whole process of marking an assessment from setting up the marking schedule and collecting the assignments to the return of work and filing of grades.

There are areas of Penmarked that could be improved; in particular we would like to include syntax highlighting. An alternative approach is to implement the same functionality into a programming IDE. In an IDE annotation could also be used for code review and for programmers to keep notes for themselves and others. Developing the annotation functionality inside and IDE has proved to be technically difficult. We hope that as pen-based computing becomes more common place that more basic controls support annotation and transparent overlays.

Many organizations are attempting to ‘go paperless’. Our experiences with this project suggest that they will have a higher success rate if careful consideration is
given to all tasks related to the activity to be supported.

9 References


Presenting Query Aspects to Support Exploratory Search

Mingfang Wu, Andrew Turpin, Simon J. Puglisi, Falk Scholer and James A. Thom

School of Computer Science and Information Technology
RMIT University,
GPO Box 2476 Melbourne, Australia, 3001
Email: {mingfang.wu, andrew.turpin, simon.puglisi, falk.scholer, james.thom}@rmit.edu.au

Abstract

Successful information search requires a joint effort from both syntactic matching provided by current search engines and semantic matching performed by human users. Word-based syntactic matching schemes work well for tasks such as homepage finding or fact finding, but they are less effective in supporting exploratory search tasks such as learning and investigation. One way to overcome this limitation of syntactic matching is to capture the search journeys of other users with semantically related queries, and use them as a roadmap to guide exploratory search.

This paper presents our investigation on the utilization of query semantics derived from query logs, to 1) increase the diversity of a search result; and 2) devise new interfaces that display a search result to support exploratory search. We conducted a user study to evaluate our initial interface prototypes. The evaluation shows that, with the interface that explicitly supports their task, subjects acquire more knowledge and are more confident about their task completeness. The differences between subjects' preferences suggest that we may need to provide a range of interfaces that can not only support users' search tasks, but also suit their personal styles.

Keywords: Information Retrieval, Exploratory Search, Search Interface, Web Search, Task based Design.

1 Introduction

A successful information search requires a joint effort from both the syntactic matching provided by current search engines and the semantic matching performed by human users. Word-based syntactic matching schemes work well for tasks such as homepage finding or simple fact finding, but are limited for more complicated needs such as learning and investigating tasks (labeled as exploratory search tasks by Marchionini (2006)) or research searches (Guha et al. 2003). For example, if a high school student is learning about the concept of “acid rain” and is asked to write an informative and comprehensive essay on the topic, a query containing “acid” and “rain” may yield many relevant pages, however if all these pages are about the definition of “acid rain”, that would not satisfy the student. With current information retrieval systems (typically represented by Web search engines), the student has to explore and gather information about all possible aspects of “acid rain” by iterating through a process of query reformulation and search result browsing. In the end, the student may be still unable to obtain a broad overview of the issue. To provide better support for such learning or exploratory search tasks, an information retrieval system should go beyond word-based match and the expectation that information needs can be answered with a single query. It should proactively provide users with related information, and an intuitive interface that guides users’ search and browsing activities.

In the Web search context, aggregated search logs from many users provide a rich information source for mining aspectual semantic relations about a topic. They provide a link between different aspects of a topic that are semantically related to the original query, but where the documents describing the aspects do not necessarily contain the original query words. For example, we followed and extracted sequences of related queries that were sent before and after the query “acid rain” by nearly 30 users from a query log. As we can see from Figure 1, each query chain indicates a user’s search and learning journey about “acid rain” or a particular aspect of “acid rain”. While some of the aspects contain the keywords “acid rain”, many do not. Representative queries from each chain can be used to populate the search results for the query “acid rain”, allowing a new user with the same query to read-and-choose, rather than reformulate the same query or choose from suggested queries, and so make more complex decisions about their next search.

![Figure 1: Example of query chains of “acid rain”](image-url)
Two recent papers study extraction of semantically related queries from a query chain (Badelinski and Dumas 2006; Parikh and Sundaresan, 2008). The suggested queries, as provided by major Web search engines (such as Google, Yahoo! and LiveSearch/Bing), include most query aspects of a topic that one would reasonably expect. In our study presented here, we focus on the use of aspectual semantic relations to support a user’s interaction in exploratory search tasks. In particular, we investigated how to use aspectual semantic relation to: 1) increase the diversity of a search result; and 2) devise new interfaces that display a search result to support exploratory search. As we focus on the search result organization and presentation in this study, we follow the typical cycle of user interface design and evaluation. Firstly we analyzed the exploratory search task, and users’ search behavior when under taking such a task, through reviewing related work. Next, we designed interfaces based on this understanding, and then conducted a pilot study to evaluate and compare the candidate interfaces with a small number of users and topics. This paper will present this design process and what we have found and learnt through the pilot study.  

2 Related Work  

2.1 Exploratory Search  

An understanding of a user’s search context and search intent behind a query is essential to the design and development of effective search algorithms and search interfaces. Marchionini (2006) classified users’ search intents into three categories: lookup, learn and investigation. When users perform a lookup search activity, they know exactly the resource for which they are searching, and can express their needs through keyword queries. Examples of lookup activities include fact finding, known item search and question answering. Such types of lookup searches are supported reasonably well by current Web search engines. 

The other two search activities, learn and investigation, are labeled by Marchionini as exploratory search (alternately, Guha et al. 2003) labels them research searches). An exploratory search task involves learning and gathering enough information about a topic so that users can accumulate new knowledge about the topic through the information seeking process. There are two major difficulties associated with exploratory search tasks: 1) the topic domain is usually unknown to users, thus it is difficult for users to describe what they are looking for through a query; and 2) the answer space is usually multi-dimensional and users’ information needs cannot be satisfied with a single query. Thus, when users start such a search task, their initial query is usually vague and broad. They then refine their queries according to what they learn from previous search results, and aggregate relevant information on the way. Compared to the lookup search task, exploratory search requires users to perform higher level cognitive processing of each query search iteration: users must spend a significant amount of time browsing search results from each query, reading, interpreting, comparing, evaluating and synthesizing unfamiliar information, revisiting what they found, and aggregate what they learn. This culminates in the formulation of a new query to retrieve information that either confirms what they have learnt, or leads to new dimensions of the answer space. This is a berrypicking information seeking process as described by Bates (1989).

2.2 Search Interface  

A search interface is a communication channel between a user and an underlying search engine, hence an important part of an information retrieval system. Current search interfaces usually provide a query box and a ranked list of retrieved documents, regardless of a user’s search task or search intent. Significant effort has been invested in tailoring search and ranking algorithms to particular search tasks in the last decade, as evidenced by various TREC (Text REtrieval Conference) tracks such as home page finding, new topic detection and question answering (Voorhees and Harman 2005). However, search interfaces have remained largely unchanged over the same period. A query box and a ranked list of retrieved documents may be good enough for lookup search tasks, where users know what they are looking for and can recognize relevant documents from a few top ranked pages, but they are inadequate to support complex information seeking tasks in which users are required not only to recognize relevant documents, but also to make sense and use of these documents. 

An ideal interface should direct and focus users’ attention to their information search tasks and allow them to interact with information effortlessly. Designing a search interface to support a user’s information task is highly overlapped with design goals of human computer interaction (HCI). In fact, we can regard the design of a search interface as an example of HCI or interface design. Norman (1988) emphasized that when we design HCI, we ought to be asking what tasks people need to accomplish and what tools and technologies are most appropriate for those tasks. Task-based user interface design has been a key methodology of HCI design, where task analysis serves as the primary source for determining which tasks and sub-tasks can be well supported by a system, and how an interface should be structured (Wilson and Johnson 1996).

There have been successful examples of building task-based search interfaces that are specific to certain domains and users. Wu et al. (2004) demonstrated that when users were provided with an interface that explicitly supported their topic distillation task, their search performance significantly improved over the use of a generic interface that showed search results in a ranked list. Based on the understanding of users’ needs and tasks for a personal information management task, the Phlat system (Cuttrell et al. 2006) provides an intuitive interface that integrates search and browsing through a variety of associative and contextual cues. Apart from various querying features, Phlat allows users to manipulate search results according to features or tags that can be associated with documents. The Flamenco search interface (Yee et al. 2003) is an example of emerging faceted search systems, which uses hierarchical faceted metadata in a manner that allows users to both refine and expand the current query while maintaining a consistent representation of search results in the document collection’s structure. The interface also provides guided or structured browsing. While such an interface is effective for searching and browsing a well-structured collection with rich metadata, it would be very challenging to apply such an interface to the web search context, where web pages are unstructured and heterogeneous (Teevan et al. 2008).  

2.3 Evaluation  

Evaluation of retrieval systems has largely followed the Craufurd methodology (Cleverdon 1967, Sparck-Jones 1981), where a test collection includes a collection of documents; a set of queries or topics; and rel-
evance judgements (usually binary) about each document with respect to each query. A system is then fed the set of queries and scored according to its ability to retrieve relevant documents. This methodology is still widely used today in the TREC (Text REtrieval Conference) evaluation framework (Voorhees and Harman 2005) with increased size of test collections and more realistic search tasks. The advantages of this methodology include the ability to isolate and compare individual components or algorithms embedded within a retrieval system by manipulating algorithmic parameters. Moreover, the experiments are (usually) repeatable by others, and are much less resource intensive than live user studies of information retrieval systems. This system driven evaluation methodology have played an important role in the advancement of search engines.

However, recent papers question whether many of the reported advancements in search engine technology measured with such a methodology are real, or an artefact of the evaluation method (Armstrong et al. 2009a,b). Furthermore, the existence of an information retrieval system is to assist people to find relevant information that allows informed decision making, or to bridge their knowledge gap (Belkin 1980). Ingebersen and Jarvelin (2004) argue that evaluation of an information retrieval system should be enlarged to include users and their problem context. Thus an information retrieval system should also be evaluated with its potential users and with those users interactively seeking during the retrieval processes.

In 1995 (TREC 4), TREC initiated an interactive track. The goal of this track was two fold: to investigate searching as an interactive task by examining the process as well as the outcome; and to develop better methodologies for the evaluation of interactive information retrieval systems (IIR) (Over 2001). The evaluation method as formalized by this track has become a defacto standard followed by many researchers to evaluate IIR systems in a laboratory controlled environment (Dumais and Belkin 2005).

In the TREC IIR evaluation framework, subjects (recruited users) are given a number of topics and asked to perform searches on given topics. During their search process, subjects would make their own judgments on what information is relevant and what is not. Subjects’ interaction with systems under evaluation are logged, and their experiences of the systems are collected through questionnaires or think-aloud methods. The evaluation criteria are a mix of IR criteria (such as user performance) and HCI criteria (such as a system’s accessibility and usability, and user satisfaction).

The TREC IIR evaluation framework has its limitations. Influenced by the system-driven evaluation approach, TREC IIR evaluation focused on the comparison of users’ performance with different systems. The systems usually had part of their components specified by the system designer. Thus, the systems collected through questionnaires or think-aloud methods. The evaluation criteria are a mix of IR criteria (such as user performance) and HCI criteria (such as a system’s accessibility and usability, and user satisfaction).

The TREC IIR evaluation framework has its limitations. Influenced by the system-driven evaluation approach, TREC IIR evaluation focused on the comparison of users’ performance with different systems. The systems usually had part of their components specified by the system designer. Thus, the systems collected through questionnaires or think-aloud methods. The evaluation criteria are a mix of IR criteria (such as user performance) and HCI criteria (such as a system’s accessibility and usability, and user satisfaction).

The TREC IIR evaluation framework has its limitations. Influenced by the system-driven evaluation approach, TREC IIR evaluation focused on the comparison of users’ performance with different systems. The systems usually had part of their components specified by the system designer. Thus, the systems collected through questionnaires or think-aloud methods. The evaluation criteria are a mix of IR criteria (such as user performance) and HCI criteria (such as a system’s accessibility and usability, and user satisfaction).

3 Utilizing Query Semantics to Support Exploratory Search

As discussed in Section 2.1, when a user conducts an exploratory search, the problem domain investigated is usually new to the user. Users devote much of their search time examining and comparing search results; every piece of new information may lead them to pursue new directions and formulate new queries (Bates 1989). Thus to support exploratory task effectively, we need to develop a more intuitive interface that could: reveal dynamic, multiple relationships among retrieved documents; provide a better explanation of the relationship between retrieved documents and the query than query word highlighting: proactively collect document attributes (such as document type and creation date) and show them in an appropriate context; as well as to provide better feedback or query recommendations to users.

In this section, we describe two attempts to improve the exploratory search process. Both approaches use an interpretation of the semantics of an initial query as a resource, but one enriches search result presentation, while the other diversifies the coverage of the search result list. We first discuss how the semantics of a query are derived, and then describe our two approaches.

3.1 Semantic Relations

Semantics is the study of meaning, and the meaning is usually represented through a relationship between two or more word entries or concepts. There exist many types of semantic relations. For example, WordNet (Miller 1990) includes relations such as synonymy, antonymy, hyponymy and hyponymy between words. The KL-One knowledge representation system captures the subsumption or super-concept relation between concepts (Brachman and Schmolze 1985).

Semantic relations can be formally defined and captured in manually (or semi-manually) constructed systems such as WordNet and KL-One. It can also be extracted automatically from document corpora (Anyanwu et al. 2005). Manually constructed relations are generally more accurate than those that are automatically extracted, however they are more costly to build, and difficult to extend when new concepts arise.

In this study, we try to support the learning and knowledge acquisition task by utilizing concepts that are semantically related to a query topic. For example, the concept “acid rain” in a high school science subject, would not only involve a definition, but also the following related concepts: “causes of acid rain”; “the relationship between acid rain and global warming”; and “the relationship between acid rain and pollution”. We refer to these related concepts that are essential for understanding a target concept as aspects of the target concept.

Now the question is, given a topic as expressed by a query, how is the set of aspects derived? Query logs from Web search engines provide a rich information source to (partially) solve this problem in the context of Web search. Studies on query logs have shown that if a query was issued by many users, it is likely that the information need of those users differ (Te-}

Some researchers also suggested longitudinal studies, especially for those IIR systems that are designed to support more complicated search tasks than simple fact finding (Kelly et al. 2009).

There have been a number of studies attempting to extract semantic relations from query logs (Parikh et al. 2010).
A problem with the related search functionality provided by major search engines is that it only provides an overview of possible semantic dimensions of a search topic, but lacks a preview of each aspectual dimension. Such an overview typically does not provide the user with enough information to assist them to decide whether to explore a particular aspect further. As defined by Greene et al. (2000), an overview is constructed from, and presents, a collection of objects of interest, while a preview is extracted from, and acts as a surrogate for, a single object of interest. An appropriate representation and display of overview and preview can facilitate rapid elimination of aspects that are not of interest to a user. For example, the query biased summaries typically presented as part of a ranked list are previews of a webpage and influences a user’s decision whether to view the webpage or not (Tombros and Sanderson 1998, Wu et al. 2001, Turpin et al. 2009).

The new search interfaces we explore present a ranked list of previews for each aspect of the initial query in a two dimensional table. Figure 2 shows an example for the “acid rain” query. Each row of the table is an aspect, and previews of ranked documents for each aspect appear across the columns of the row. The preview in each cell is simply the query biased summary of the document generated by Microsoft LiveSearch using each aspect as a query. We expected that, by using this tabular aspect interface, a user can ascertain possible coverage of the topic vertically, while exploring individual aspects horizontally. Users can focus their attention on examining, understanding and relating the available information, instead of expending cognitive effort formulating search strategies.

3.3 Diversified Search Results

Instead of explicitly presenting semantically related aspects of a query topic in the tabular aspect interface, an alternate approach is to present these aspects implicitly through a diversified search results list. As most users are comfortable interacting with a ranked list interface (such as those used by the major Web search engine companies), we assume many users may prefer the simplicity of a list based approach.

Diversifying a search results list has been advocated by IR researchers for over a decade. A diversified list may satisfy users with ambiguous queries, or broad queries with multiple aspects (Carbonell and Goldstein 1998, Paramita et al. 2009). To support an exploratory search task, a diversified list should cover as many different aspects of the query topic as possible. Here we take a simple breadth strategy which involves two steps. Firstly, given a query, we collect its aspects through the “related search” suggestions of a major search engine and retrieve the top 20 webpages from the original query and each aspect query. To ensure accuracy and prevent topic drift, each query aspect must contain the original query words. Second, we rank the aggregated search results according to the number of aspects contained in a document, taking into account the number aspects that have not been covered by higher ranked documents. In other words, the top document has the most aspects, the second has the most aspects after the top document’s aspects have been removed from consideration, and so on.

Figure 3 shows the presentation of a diversified list that is ranked based on a document’s coverage and novelty. The top ranked pages usually cover more than one query aspect, thus the page has multiple previews: one for each aspect. There is (typically) not enough room on a single page to show all of the previews for all of the documents, and so each page is still represented by its title, a preview (randomly chosen amongst all aspects) and its URL. If the page has more than one aspect, a label “more summaries” will be placed next to the title, and if the user hovers her mouse over the label, a text box pops up and displays other previews for the same page.

It has been demonstrated that when users visit a webpage from a search results list, they may browse within the website, particularly if they feel the site can answer their information need (Pirolli and Card 1995). Hence we could consider presenting a list of websites, rather than documents, ranked by the site’s coverage and novelty. There are two possible advantages of treating a website as a unit for ranking. Firstly, we could find an information-rich site that is devoted to the query topic; a list of ranked websites would function like a resource page or topic distillation page (Craswell et al. 2003). Secondly, users could focus their attention on possibly relevant pages from a site instead of spending time browsing the site to locate these pages. They may not even be able to find them if a website is not structured appropriately.

Accordingly, we built an interface that ranks sites, rather than documents, with previews of multiple documents from the same site being presented to users. As the number of previews to list for each site can quite large, we opted for an in-line presentation of the lists, rather than the hover-based interface used for aspects of a single document. Figure 4 shows an example of the list of ranked sites, and then if a user selects “Click here for more matches from this site”, the summaries of pages are shown inline. Figure 5 shows the result if the first site is selected from Figure 4. The expanded area will disappear if the user clicks the label again.

Figure 3: Diversified List with in-line previews interface. Summaries showing the aspects of one page are displayed as the result of a mouse-hover action.
4 Evaluation

In the process of researching and designing an interface to support the exploratory search task, we followed design → evaluation → redesign cycle. This section introduces a discounted usability evaluation method (Nielsen 1989) that we adopted to evaluate our initial interface prototypes.

We recruited a small number of subjects and re-
with site-based ranking and in-line previews (Figure 4 and 5). We decided not to include the diversified list with page-based ranking and hovering preview lists, as initial user feedback indicated that while users preferred multiple snippets extracted from a page, they did not like the presentation style. We are exploring other possible presentation styles for lists of this nature.

For the two interfaces, we chose a similar interface from mainstream search engines as a baseline for comparison. For the tabular aspect interface, we chose the “related search” interface (Figure 6) from LiveSearch; for the site-based diversified list, we chose a conventional ranked list as shown in Figure 7.

We hand picked four queries from a Web search log that might suit our subjects. For each topic, we developed a task scenario that simulates a search task in the real world. The four topics (and a training topic) are shown in Table 1.

Table 1: The search topics.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acid Rain</td>
<td>Imagine you are doing a science project on acid rain; you need to find as many facts about acid rain as possible and understand how it impacts our environment, so you can write a thorough and well-researched report.</td>
</tr>
<tr>
<td>Scientology</td>
<td>Imagine that you hear this word from all sorts of media now and then and you were also asked by your friends what Scientology is about: is it a science, a religion or a cult? You set out to search for the truth about Scientology so you could understand what it is and therefore help you to form your own opinion.</td>
</tr>
<tr>
<td>Nanotechnology</td>
<td>Imagine that you are a science reporter of a leading newspaper, and you are asked by your boss to write a news story on nanotechnology so the public can be informed about this new technology. Think what the public would like to know about a new technology and what information you need to provide in such an essay. And now please go and find the information that you think it would be required.</td>
</tr>
<tr>
<td>Impressionism</td>
<td>Imagine that your friend is inviting you to visit an impressionism exhibition this weekend. You would like to know more about impressionism before the visit so you can get the most out of the visit and also impress your friends.</td>
</tr>
</tbody>
</table>

One of your close friends told you that he was recently diagnosed with Diabetes. You might hear about this disease before but you have only limited knowledge. Now you would like to learn more and get more information about Diabetes so that you can communicate better with your friends in the future.

<table>
<thead>
<tr>
<th>Training topic: Diabetes</th>
</tr>
</thead>
</table>

Table 2 shows a block of the experimental design. Each subject searched one topic per interface; the order of topics were fixed, while the four interfaces are rotated in each position. A complete design requires a group of four subjects.

We recruited eight subjects and divided them into two groups so that the block design was repeated twice. The four subjects from the first group were postgraduate students from our School of Computer Science, while the other four subjects were from various backgrounds: two of them were teenagers (both 16 years old) and another two were middle aged professionals.

Figures 2 and 4-7 show the four testing interfaces respectively. As shown in Figure 7, two buttons, “save” and “cancel”, are placed next to each retrieved document. Subjects can save a document if they think that this entry is useful, or cancel their previous saving if they change their mind. When subjects started a new topic, they were initially presented with the same set of pre-saved documents. If these initial candidate documents could not satisfy a subject’s need, they could formulate their own queries. Their queries were directed to LiveSearch through its API, and search results were displayed as a ranked list, as the related search topics were not included in the API.

During sessions, all significant events such as documents viewed and saved, and queries sent, were automatically logged and time-stamped.

4.2 Experimental Procedure

Subjects undertaking the experiment were instructed as follows. Firstly they read a plain English statement about the experiment and signed a consent form in compliance with our university’s Ethics Board. The
search task and the four testing interfaces were then explained and demonstrated to subjects. Subjects had a practice with each interface using the example topic, and were encouraged to ask any related questions. After the subjects acknowledged their understanding of the search task and search interfaces, they started their tasks, following the sequence of assigned topics and interfaces as per the block design.

For each topic, subjects filled in a pre-search questionnaire that gathered their familiarity with the search topic, they then commenced searching to gather information for the assigned task. When they finished interacting with the initial testing interface, but before continuing with their own search, they were asked to fill in an intermediate questionnaire that asked them how much they had learned about the topic and to describe their search experience. The same questions were asked again in a post-search questionnaire to those subjects who continued searching using their own search queries. After the subjects finished all four search topics, they filled in an exit questionnaire that asked them to compare the four interfaces.

Subjects had a maximum of 20 minutes to search on a topic. They could exit earlier if they felt that they found enough information for that topic.

4.3 Experimental Measurement

This experiment had an independent variable (the interface), which has four possible values: Conventional list, Related Search, Diversified List, and Tabular Aspect.

We devised two types of dependent variables to measure against the independent variable: subjective measure and objective measure.

For the objective measures, we count how many queries a subject sent, how many documents a subject read and saved, and, most importantly, the quality of their saved document set. In order to measure the quality of their saved set, the eight possible sets of saved documents for each topic (one per subject) were assessed for aspect coverage by the final four authors of this paper (who are experienced judges in information retrieval tasks). Note that none of these authors were involved in developing the topics, tasks, nor selecting the documents and lists for the experiments. During the assessment process, each set was rated on a 1-5 Likert scale: with 1 meaning “very narrow”, 2 “narrow”, 3 “somehow covered”, 4 “better covered” and 5 “most covered”.

This evaluation of the saved set is relative to the other subjects in the task. That is, if a particular topic is difficult, or the lists provided in the interfaces poor in some wider sense, then most subjects will have poor coverage of aspects in their sets, but some may still get a high score of 5 relative to the other subjects.

Our subjective measures include: subjects’ preference of each interface; their confidence of task completion; and the amount of knowledge acquired after they searched. The values of these variables were collected from questionnaires and are now described in detail.

Acquired knowledge: To measure how much knowledge a subject acquired after an interface was used, we designed the following questions for the pre-search, intermediate search and post-search questionnaires. We first asked subjects the following question in the pre-search questionnaire; the subjects were required to choose an indicative scale as answer.

How much do you think you know about the topic?

1. Not at all
2. A little bit
3. Somehow
4. A lot
5. Extremely knowledgeable

After the subjects searched with each interface, a similar question was then asked again in the intermediate questionnaire.

How much do you know about this topic now?

We could use the difference of two scores from above questions to measure how much a subject learnt. However, if a subject does not know a topic very well, they might not be able to truly answer the first question well. As one subject said: “I thought I knew about acid rain a lot, but after I searched and read, I found I only knew a little bit.” Hence we added the following question in the intermediate questionnaire.

Now you have learnt about this topic, please re-evaluate how much did you know about this topic before your search?

The difference between the value chosen in the intermediate questionnaire and the one from this re-evaluated question was used to measure the amount of acquired knowledge.

Confidence of task completion: This variable was measured by subjects’ response to the following question in the post-search questionnaire.

How certain are you that you have got enough information to fulfil your assigned task?

1. Not at all
2. A little bit
3. Somewhat
4. A lot
5. Extremely

Subjective preference: We asked subjects about their preference for the four interfaces in the exit questionnaire. Subjects were asked to say which they preferred out of the Conventional List interface and the Diversified List, and then which they preferred out of the Related Search interface and the Tabular Aspect interface. Subjects were also asked two open questions: 1) What did you like about each interface? and 2) What did you dislike about each interface?

5 Experimental Result

5.1 Objective Measures

Twenty minutes is not a lot time to carry out an exploratory or research type of search task. Most of our
Figure 8: Objective measures (mean of 8 subjects).

Figure 9: Aspect coverage of saved set. Each topic is the mean over 2 subjects. Average is the mean over all 8 subjects.

Figure 10: Acquired Knowledge. Each topic is the mean over 2 subjects. Average is the mean over all 8 subjects.

subjects spent twenty minutes in interacting with the initial answer lists or interfaces without doing much further search. Figure 8 indicates where the subjects, on average, spent their time. As we expected, the subjects read more pages with the Diversified List and the Tabular Aspect interfaces than the other two interfaces. The subjects who interacted with the Conventional List and the Related Search interfaces tend to issue their own queries (4 users issued queries for both of the former, compared with 2 and 1 for the latter pair, respectively). Between the two interfaces that explicitly showed query aspects, the subjects visited more aspect entries from the Tabular Aspect interface (4.4) than the Related Search interface (3.3).

Figure 9 shows the aspect coverage. The aspect coverage averaged over all four topics (final bars) is in an increasing order from Conventional List, Diversified List, Related Search, to Tabular Aspect interfaces. Topic by topic, the subjects with the aspectual interfaces (either explicit or implicit) performed better than the two list interfaces (except for Tabular on Topic 4).

Generally, the more pages a list contains, the more aspects it covers; except for those lists saved from the Diversified List interface. Although the subjects saved the most pages from the Diversified List interface, these pages did not have the most coverage. A possible reason could be that as most saved pages are from the same site, they tend to cover similar sets of aspects.

5.2 Subjective Measure

Acquired Knowledge: Figure 10 shows the different scores of the four interfaces topic by topic: subjects gave the highest score to the Conventional List for Topic 1, the Diversified List for Topic 3, and the Tabular Aspect for Topic 4. The Tabular Aspect and the Related Search interfaces are equivalent for Topic 2. Overall, as indicated in the group of the rightmost bars in Figure 10, subjects felt that they acquired more knowledge from the Tabular Aspect interface, followed by the Diversified, Related Search and Conventional List interfaces, although the difference was not statistically significant.

Confidence of task completeness: Subjects’ confidence of task completeness also varies from topic to topic, as shown in Figure 11. On average, subjects were the most confident of getting enough information using the Tabular Aspect interface, followed by the Related Search and the Diversified List interface. The Conventional List interface inspired the least confidence in completeness of the coverage of information gained.

It is interesting to note that although the Spearman Rank Correlation is low (and not significant) between the acquired knowledge as self-assessed by the subjects and the aspect coverage as assessed by some authors ($\rho = 0.29$), the self-confidence of task completeness is significantly correlated with the aspect coverage ($\rho = 0.60, p < 0.01$) and the acquired knowledge ($\rho = 0.54, p < 0.03$).

Subjective Preference: Subjects were asked to choose a preferred interface between the two list interfaces and the two aspect interfaces. Among eight subjects, five of them preferred the Diversified List interface over the Conventional List interface, and the same number of subjects preferred the Tabular Aspect interface over the Related Search interface. Four of the subjects are postgraduate students with a Computer Science and Information Technology background. It is interesting to note that two of them pre-
ferred the Conventional List over the Diversified list and three of them preferred the Related Search interface over the Tabular Aspect interface. Seven subjects replied to the question that asked them to name a preferred interface: five subjects named the Tabular Aspect interface (only one of them is the postgraduate), two named the Related Search interface, and one the Conventional List interface.

The subjects’ choice of preference indicates a trend that people from different backgrounds may prefer different interfaces: our postgraduate students preferred the Conventional List and the Related Search interfaces, they commented that there was too much text in the Tabular Aspect interface; while subjects from non-computing backgrounds showed the opposite preference. We conjecture that the postgraduates might have more knowledge on how a search engine works, and their search and browse behaviors are more analytical than the other group of subjects. The other four subjects who preferred the Tabular Aspect interface commented that all information is available from the interface, the two teenagers explicitly said that they could do less (search and click) with the Tabular Aspect interface. Their interaction behaviors from the list interfaces, click on a page and then read windows or tabs, which they then read one by one; unlike the postgraduates who tend to read the lists sequentially and followed this cycle: read page summaries from the list interfaces, click on a page and then read the page.

Our future work will include redesigning and refining the prototype interfaces and testing them with more users and users with wider backgrounds; and ultimately, when we get a reasonably accepted interface through laboratory controlled evaluation, we would like to conduct a longitudinal user study to test users’ true acceptance of the interface.

6 Discussion and Conclusion

We explored different interfaces that support learning and investigative search tasks. We conducted a user study to verify whether these interfaces could achieve the desired goals of improving the amount of information that users could discover about a topic in 20 minutes. The feedback from this pilot evaluation will guide our further improvement of the interfaces.

Our results show that, by either self-assessment or objective assessment, subjects acquired more knowledge and are more confident with the interface that explicitly supports their learning and information gathering tasks. There is a significant correlation between subjects’ confidence of task completeness and the acquired knowledge. However, subjects with different backgrounds preferred different answer presentation styles and showed different interaction behaviors. These findings indicate that we should combine the advantages of each interface and/or provide a range of presentation choices so that users could have some control on what should be included in the answer list and how the answer list should be presented.

While the number of users and topics is small, we feel that the observations made in the study are informative. In particular, the preference of postgraduate computing students for the “traditional” Conventional List interface is notable. Many user studies in the field of Information Retrieval make use of computing graduate students as subjects, and often the conjecture is made that the results would not carry over to other user populations. This study in part confirms that conjecture, emphasizing that extreme care must be taken when interpreting results of users studies on specific cohorts of participants.

Acknowledgements

We thank Microsoft Research for providing a search API to their LiveSearch, and for research funding for this study.

References


Paramita, M., Sanderson, M. & Clough, P. (2009), Developing a test collection to support diversity analysis, in *Proceedings of Redundancy, Diversity, and Interdependent Document Relevance workshop held at ACM SIGIR*.


Usability of Navigation Tools for Browsing Genetic Sequences

Paul Rutherford¹, Walt Abell¹, Clare Churcher¹, Alan McKinnon¹, John McCallum²

¹ Department of Applied Computing, Lincoln University, PO Box 84, Lincoln, New Zealand
{paul.rutherford, walter.abell, clare.churcher, alan.mckinnon} @lincoln.ac.nz
² Plant and Food Research, Private Bag 4704, Christchurch 8140, New Zealand
McCallumJ@crop.cri.nz

Abstract
Software to display DNA sequences is a crucial tool for bioinformatics research. This study examined techniques for navigating large DNA sequences via panning and zooming. This involved surveying the navigation facilities of current bioinformatics applications and performing a heuristic analysis on the most common interface controls found. Several prototypes for sequence navigation via panning and zooming were then developed and usability trials carried out, getting users to perform common sequence navigation tasks using the prototypes. The ‘Connected View’ design was found to be most usable for panning while the zooming results were less clear. The outcomes of this type of research can help improve bioinformatics applications so that will be more usable by the target research users.

Keywords: bioinformatics, DNA sequences, usability, user interface, navigation.

1 Introduction

The technology to display DNA sequences was developed in the mid-1970s and the volume of such data has been growing exponentially since then. Bioinformatics tools which apply computing and statistical techniques to such data are now commonly used. However, much of this software is developed or designed by scientists who typically have little formal training in user interface design issues, or by software developers who often have little understanding of the needs of researchers in the field. It is not uncommon for users of bioinformatics software to experience a steep learning curve and to be overwhelmed by the complexity of performing standard tasks.

The overall aim of this study was to evaluate different approaches for browsing DNA sequences on a computer to improve the usefulness of bioinformatics software. It

applied principles of user interface design, navigation and usability to applications that allow users to navigate sequences to look for particular features or attributes. The study looked at the type of browsing capabilities and controls provided by current bioinformatics applications and used these as the basis for the design of several prototypes. The efficacy and efficiency of the prototypes as well as user preferences were determined through a usability trial.

2 Background

DNA sequences are long strings of the letters A, C, G and T which represent the nucleotides (commonly called “bases”) Adenine, Cytosine, Guanine and Thymine. These letters are repeated in various combinations and can number into the thousands (or even millions) of characters in a single sequence. Clearly it is not possible to display this amount of information on a single screen. However, even sequences of a few hundred letters can still cause information overload for a user.

In addition, sequences are often annotated with a number of “features” which are segments of the DNA known to have a specific purpose. For example, Start and Stop Codons which mark the beginning and end of a subsequence and Exons which encode a protein product. Sequences have been displayed in various formats. A common display method is to show the sequence horizontally, with a ruler for the location of the bases and any annotations shown above and/or below the sequence. An example of this is shown in Figure 1 (Lorraine and Helt, 2002).

![Figure 1: A simple display of a DNA sequence](image1)

Figure 1: A simple display of a DNA sequence

While this provides the detailed information for a particular region of the sequence, it is often necessary to look at the annotations over a much larger region, thus requiring a less detailed view, often referred to as an overview. An example of this is shown in Figure 2. Note that the ruler and features are still visible but the base letters are not.

![Figure 2: A sequence overview display](image2)
2.1 Information Spaces

The display of a genetic sequence is an example of an information space (Benyon and Höök, 1997). As information spaces increasingly ‘go digital’, there are some intrinsic characteristics that impact on their navigation. These include the lack of a “stable Euclidean geometry” (Dahlbäck, 1998), relatively unconstrained navigation (Benyon and Höök, 1997) and “a lack of explicit or implicit information that [movement is] in the right direction” (Dahlbäck, 1998). These characteristics combined with the large amount of data that can be stored digitally can contribute to users ‘getting lost’ which has been identified as a major problem in information spaces (Dillon et al, 1990; Spence, 1999).

Three general navigation activities in information spaces were described by Benyon and Höök (1997). These are listed in Table 1 with an explanation of what each activity is trying to achieve and an example applying to genetic sequences.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Objective</th>
<th>Genetic Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploration</td>
<td>To see what objects are present and their relationships.</td>
<td>To investigate the number and order of features in a sequence.</td>
</tr>
<tr>
<td>Wayfinding</td>
<td>To browse to a specific location.</td>
<td>Find the location of the first base of the first exon in a sequence.</td>
</tr>
<tr>
<td>Identifying</td>
<td>To understand information about a set of features.</td>
<td>Find out how many exons there are between the start codon and position 2000.</td>
</tr>
<tr>
<td>objects</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Description of navigation activities

2.2 Navigation Aids

One of the issues in user interfaces for working with large information spaces is how to allow the user to navigate without losing track of where they are in the space. It may also be necessary to carry out comparisons between sections of the data that are quite far apart. Finally it is often necessary to be able to easily switch back and forth between a detailed view and an overview of the data.

Programs that deal with display and searching of genetic sequences suffer from the age old problem of how to show the appropriate level of detail while allowing the user to maintain the context from a larger area than can be accommodated on the screen. This problem has occurred in many application areas and various approaches have used such as distortion techniques (e.g. Fish Eye Lens and Distortion Wall) as well as ‘connected views’ for overview and detail.

Many of these techniques have been tested experimentally and some implemented in applications. As is often the case, the efficacy and efficiency of an approach varies depending on such factors as how well the feature is implemented, the sophistication of the end user, the type of task undertaken, and the specific application of the techniques involved.

On the other hand, standard office applications (e.g. word processors) and web applications offer somewhat standard approaches to navigation through large documents, i.e. scrolling, zooming, etc. It may be that some of these common approaches are suitable for browsing genetic sequences.

Where non-professional developers (in this case biological researchers) actually carry out application development (or play a significant role in the design), usability considerations may not be a top priority. Typically the types of users who develop bioinformatics applications are primarily interested in obtaining accurate and meaningful output (e.g. a clear diagram from part of a sequence). Features like user friendliness and appropriate interface controls may not be seen as directly contributing to the output and so not receive much attention (especially if software development is not officially part of a user’s job description).

3 Purpose of the Research

This study sought to understand how navigation of genetic sequences has been included into various bioinformatics applications and experiment with various ways of offering appropriate navigation features. To this end, the study was structured as follows:

- Cataloguing of the navigation features in current bioinformatics applications that provide genetic sequence browsing. This was followed by a heuristic evaluation of the user interface controls for browsing found in the applications.
- Development of several prototypes for sequence browsing that employ the most promising user interface controls identified in the heuristic evaluation.
- Performing a usability study on the prototypes developed to determine the efficacy, efficiency and user preference for type of control.

4 Bioinformatics Applications

There is a wide range of software available to support bioinformatics research, ranging from databases for lab management to 2D and 3D visualisation of data. For the purposes of this project, software was examined that allows some form of sequence browsing.

Altogether, 20 applications were examined including many in wide use within the Bioinformatics research community, e.g. BLAST (McGinnis & Madden, 2004) for comparing new sequences to a global database and Ensembl (Hubbard et al., 2005) for accessing data from the GenBank sequence database. The appendix contains a complete list of the applications surveyed.

Each application was examined to determine:

- the number of views provided, e.g. overview and detail
- the user interface controls provided for changing the views, e.g. panning or zooming
- the ‘connectedness’ of the views, i.e. did changing one of the views cause a change in the other views

For example, Ensembl provides several interconnected views at different levels of detail. The interface is very ‘space-intensive’, sometimes requiring multiple screens to view all the information. A (cut down) example of the display is shown in Figure 3.

Ensembl views may be panned and zoomed however these transitions require the display to be refreshed. Panning is provided through buttons that move a fixed distance in a particular direction. There are also buttons for zooming as well as a control to select the zoom level.
A heuristic evaluation of each of the designs identified for panning and zooming was then undertaken using the 10 usability principles defined by Nielsen (1994). Each design was examined for issues that conflicted with one or more of the usability principles. Each issue was rated as a “problem” (would definitely affect users) or a “warning” (could affect users but the effect could be minimised through minor redesign). For example, Table 2 shows the evaluation for the Zoom Control in Ensembl.

### Table 2: Issues for Ensembl Zoom Control

<table>
<thead>
<tr>
<th>Type</th>
<th>Principle</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problems</td>
<td>Internal model</td>
<td>No consistent way to interpret the scale; could associate small bars with more detail or interpret as showing less detail (i.e. overview).</td>
</tr>
<tr>
<td>User control</td>
<td></td>
<td>Only a limited set of levels available.</td>
</tr>
<tr>
<td>Recognition</td>
<td></td>
<td>Difficult to label buttons to indicate detail and overview. Design relies on recall and/or complex labelling.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Warnings</th>
<th>Standards</th>
<th>Non-standard design but familiarity with buttons may compensate.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recognition</td>
<td></td>
<td>Labelling important so users will recognise purpose of each button.</td>
</tr>
</tbody>
</table>

In addition to bioinformatics applications, the designs of a few common applications (e.g. Acrobat Reader and Google Maps) that provide panning and/or zooming were also evaluated. This was done to consider whether user interface designs from common software could be useful for sequence browsing.

A summary of the designs from current applications and the problems and warnings produced by the evaluation is shown in Tables 3 and 4 (ordered from least to most problems/warnings).

As can be seen from the panning list, the most common design uses scroll bars. This is probably due to perceived user familiarity with this common control. The use of this control is also fairly well understood as evidenced by the low number of issues in the heuristic evaluation.

### Table 3: Evaluation of panning designs

<table>
<thead>
<tr>
<th>Design</th>
<th>Configuration</th>
<th>Occurrences</th>
<th>Problems</th>
<th>Warnings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scroll bar</td>
<td>Horizontal</td>
<td>14</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Connected view</td>
<td>Overview displayed above detail</td>
<td>8</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Buttons</td>
<td>Two or four buttons, horizontal</td>
<td>4</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Hand tool</td>
<td>Drag view in either direction</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Circular map</td>
<td>Small circular overview</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>No panning</td>
<td></td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

### Table 4: Evaluation of zooming designs

<table>
<thead>
<tr>
<th>Design</th>
<th>Configuration</th>
<th>Occurrences</th>
<th>Problems</th>
<th>Warnings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slider</td>
<td>Horizontal or vertical</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Buttons</td>
<td>2-4 buttons to alter zoom or one button to toggle between overview and detail</td>
<td>6</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>On-view slider</td>
<td>Slider is superimposed on view</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Select level</td>
<td>Choose from several zoom levels</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Magnifying glass</td>
<td>Use mouse buttons to set zoom level</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Marquee Tool</td>
<td>Select region to zoom</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Dynamic Zoom</td>
<td>Drag mouse in 'zoom mode'</td>
<td>1</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>No zooming</td>
<td></td>
<td>8</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
The results for the zooming designs were less clear cut, especially as many of the applications did not provide any method to adjust the detail level. Of those that did, the most common control was buttons of varying types. It is interesting that sliders did not feature in more applications as these had relatively few issues and operate similarly to scroll bars.

5 The Prototypes

Prototypes that provided panning and zooming of sequences were developed in Flash. These were based on the top three designs of each type from the heuristic evaluation. Figure 4 shows a screenshot of the prototype with a Connected View control for panning.

The sequence display was created from screenshots of the detail window of the Artemis 7 application (Rutherford et al., 2000). For the zooming prototypes, the images were manipulated to provide various levels of detail and code was included to provide smooth transitions between levels. Details from the Artemis overview window were used to construct a display of sequence features used in the Connected View control.

The controls used in the panning designs were all oriented horizontally:

<table>
<thead>
<tr>
<th>Panning Buttons</th>
<th>Pan left or right at two different speeds or go to the start/end of the sequence.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scroll Bar</td>
<td>Pan by clicking the arrow keys or in the tray or by dragging the thumb.</td>
</tr>
<tr>
<td>Connected View</td>
<td>Pan by clicking the arrow keys or in the tray or by dragging the thumb. The tray shows a sequence overview with the main features highlighted.</td>
</tr>
</tbody>
</table>

The controls for zooming were vertically oriented but the prototypes also included a horizontal scroll bar for tasks that required panning.

<table>
<thead>
<tr>
<th>Zoom Buttons</th>
<th>Click + to zoom in, − to zoom out.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zoom Slider</td>
<td>Drag slider down to zoom in, up to zoom out. Can also click + and − buttons to zoom.</td>
</tr>
<tr>
<td>On-view Slider</td>
<td>Drag the slider to zoom in or out. Slider will follow mouse pointer. View will pan if mouse is moved to left or right edge of view.</td>
</tr>
</tbody>
</table>

6 Usability Trials

The prototypes were incorporated into an overall application for the trials. The application contained an introduction to the display and terminology used followed by sections presenting and testing each prototype design.

There were three different tasks for the user to perform with each design:

<table>
<thead>
<tr>
<th>Task</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find a Feature</td>
<td>Find the location (number) of the first base of the first exon.</td>
</tr>
<tr>
<td>Go to Location</td>
<td>Find the four bases on the sequence from location 2000.</td>
</tr>
<tr>
<td>Identify features</td>
<td>Find the number of features between the first exon and position 2000.</td>
</tr>
</tbody>
</table>

The same tasks were repeated for each prototype design but the locations and data were varied. Tasks were presented at the top of the screen (with an answer box to fill in) and the prototype showing the sequence and relevant controls was displayed below this. Pre-testing was carried out to refine the application and the terminology used.

Participants were recruited from biological research staff and students working at Lincoln University and the nearby Crown Research Institutes. The only pre-requisite was having had some prior experience of working with genetic sequences on a computer. Human Ethics Committee approval was obtained before participants were approached. A total of seven participants were involved.

At the start of each trial, the participant was briefed by the researcher reading from a usability script. The researcher then started the trial application and observed the participants as they worked on the tasks, making notes on a pre-printed observer sheet. The application also recorded the mouse actions and timings to a file and Camtasia was used to record the screen display and user interaction for further analysis. Each trial was scheduled to last for an hour.

7 Results and Discussion

There were several items which were evaluated for each prototype, some based on the data recorded by the application and some on observations and discussion with participants.

<table>
<thead>
<tr>
<th>Efficacy</th>
<th>Were users able to get the correct answers for tasks?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Efficiency</td>
<td>How much time/effort was required for each task?</td>
</tr>
<tr>
<td>Usage</td>
<td>How did users actually use the controls?</td>
</tr>
<tr>
<td>Preference</td>
<td>Which designs did users prefer after completing the trial?</td>
</tr>
</tbody>
</table>
As there were only seven participants in the study, a formal statistical analysis was not undertaken. Instead a descriptive approach was used to analyse the results.

### 7.1 Panning Designs

All tasks for all prototypes were completed and correct answers given by all participants (100% efficacy). There seemed to be little confusion about how to use each control, possibly due to the relative familiarity of the designs chosen.

To determine the efficiency of use, the amount of time that a participant worked on a task and the number of mouse actions used were compared. One participant was excluded from this analysis as they took significantly more time and mouse actions to complete tasks than the other participants. Figure 5 shows the average time and mouse actions for the other six participants.

![Figure 5: Average Number of Seconds and Clicks for Panning Tasks](image)

Note that for the Connected View, the Find Feature task could be completed by simply inspecting the tray (which displays an overview of the sequence features) without having to click the mouse. Hence the average number of mouse clicks is less than one in this case. It is possibly not surprising that the Connected View was the clear winner in both minimising the amount of time and number of mouse actions required to complete tasks.

Many participants commented on the benefit of the additional information provided by the overview display embedded in the Connected View control. One participant said “it’s good [because] you can see what you’re coming up to, or go straight to where you want to go”. Several participants suggested showing “location indicators” in the overview as these would have assisted in the Go to Position tasks. One participant also suggested the addition of ‘Go to Start’ and ‘Go to End’ buttons to the design.

The Panning Buttons were the least efficient approach while the Scroll Bar provided mixed results. To understand why these controls performed so poorly, further analysis of the actual usage of these controls was undertaken. Figure 6 shows a typical example of the use of the Panning Buttons in the Find Feature task.

As can be seen, the user began the task by immediately going to the start of the sequence (the view for each task started somewhere in the middle of the sequence). After pausing (possibly to reorient themselves), they panned right ‘fast’ (double arrowhead button) three times, overshooting the location for which they were searching. This required them to backtrack, using progressively shorter bursts of movement to ensure they did not overshoot again. Participants had mixed reactions to this design, some describing it as “good” while another called it “annoying”.

Figure 7 shows a typical example of the use of the Scroll Bar control for the Find Feature task. Here the user completed the task with one continuous drag action. First they moved to the start of the sequence in two motions. This was completed quite slowly; perhaps they were checking the features of the sequence as it scrolled by. After reaching the start of the sequence, they paused and then quickly panned right through the sequence, overshooting the feature and then backtracking. Despite some inefficiencies in usage, participants described this design as “responsive” and “easier to use” than the Panning Buttons.
Not surprisingly, participants overwhelmingly said they preferred the Connected View control for all tasks and overall (see Figure 8). One participant could not choose between the Scroll Bar and Connected View for the Go to Position task (both preferences have been included in the chart).

### 7.2 Zooming Designs

The results for the zooming designs were not so straightforward as for the panning ones. For one thing, participants were not restricted to only using zooming as a scroll bar was included in each design to allow panning (and the On-view Slider also performed panning). In addition, the tasks for the zooming section of the trial were the same as those tested in the panning section (but with different sequence locations). Participants were free to use zooming or not to complete the tasks. Figure 9 shows the percentage of the seven participants who did not use the supplied zooming control for each task. It is probably not surprising that zooming was least used in the Go to Position task as this involved finding a specific (numeric) location in the sequence.

![Figure 9: Percentage of participants NOT using zooming controls in the zooming tasks](image)

Seven answers to the task questions (a third) were incorrect in the zooming section (as opposed to none in the panning section). This is surprising especially given that the tasks were essentially the same in both sections. Most of the errors were minor, e.g. an obvious data entry error or miscounting of the number of features. However, some of the errors may have been due to the way the program displayed the sequence which caused some distortion of the text when zooming was used. There was also a bug in the code for the On-view Slider which caused its panning behaviour to be inconsistent when used with the panning scroll bar. This affected one participant’s responses but did not appear to impact on other participants.

To analyse the efficiency of use, the timing data was separated into tasks where only the panning scroll bar was used versus where the zooming control was used as well. Figure 10 shows the average number of seconds required to complete each task for both situations. As for the panning results, the times for the participant who took significantly longer have been excluded. Note that the times must be interpreted cautiously as some represent data from only one or two participants.

In almost all tasks, the efficiency of using the scroll bar alone was better or the same as also using the zooming control. Indeed only in the Identify Features task using the Zoom Slider (where all but one participant used the control) was use of the zooming control noticeably faster than panning alone.

The overall advantage of the ‘panning only’ approach may be explained by some participants commenting that they had not previously used software that provided zooming of sequence displays. Also, the behaviour of the prototype controls was not always what participants anticipated. For example, several said that they expected the centre of the zoomed image to be in the centre of the view but the prototype did not always do this accurately.

Those using the zoom controls did so in various ways but a typical approach for the Find Feature task was to zoom out, pan to find the feature, and then zoom in on the feature in one or two movements. Figure 11 illustrates an example of this approach.

Figure 12 shows the percentage of participants preferring each zoom control for the various tasks. These figures should also be treated cautiously because not all participants used the zoom control in every task. It can be seen that some designs were preferred by users who did not actually use them to do the task (but the controls were demonstrated and explained to each participant).

Some participants described the Zoom Slider as “more straightforward” than the On-view Slider. The one participant preferring Zoom Buttons overall said that if the Zoom Slider had been displayed horizontally rather than vertically, it would have been their equal preference.
8 Conclusions

This study evaluated a number of common designs in bioinformatics software to browse genetic sequences. Based on this, a set of prototypes to provide panning and zooming were constructed and a usability trial performed. The panning results were unambiguous with the Connected View being the most efficient as well as most preferred control. It would clearly be useful for developers of sequence browsers to consider some form of this control for navigation.

The zooming results were less clear cut but illustrated the tendency for users to use features with which they are already familiar. In this case, it meant a number of tasks were completed by panning using the scroll bar and with no use of the supplied zooming control. The use of zooming is much less prevalent in existing bioinformatics applications and this may account for its low use in the trial. In addition, the tasks required were relatively straightforward (and the same as those tested with the panning controls). There were also some glitches in the operation of the zooming controls which may have put some participants off. It would be instructive to design tasks that would more obviously benefit from zooming and repeat this section of the trial (with improved versions of the controls) to see what impact this has on users’ approaches.

The controls tested in this study could form the ‘building blocks’ of full sequence browsing software. Future work could look at how to add additional facilities for real life browsing tasks, e.g. to compare sets of features from different parts of a sequence. In addition,
there are often a number of parallel ‘tracks’ of features and annotations attached to a sequence. It would be useful to consider how to adapt the Connected View to be able to show a variety of features.

Finally these prototypes attempted to provide ‘smooth’ panning and zooming displays. This is different to the majority of current bioinformatics applications which tend to redisplay the whole screen, particularly when changing the level of detail displayed. If would be interesting to test whether smooth displays would better enable users to maintain context and orientation within a sequence.

Bioinformatics software is evolving (and the number of applications increasing) at a rapid rate. Since this study was carried out, newer versions of some of the applications evaluated have been released. In most cases, they have new features for particular sorts of analyses. In a few cases, the user interface has been improved by the addition of better labelling or more predictable behaviour. It is essential that usability issues are key design criteria for bioinformatics software if it is to be of maximum value to researchers who are increasingly reliant on it.

9 Appendix
Bioinformatics applications examined in this study.
1. APIC (Bisson & Garreau, 1995)
2. Apollo (Lewis et al., 2002)
3. Artemis (K. Rutherford et al., 2000)
4. BLAST (McGinnis & Madden, 2004)
5. ChARMView (Myers, Chen, & Troyanskaya, 2005)
6. DNAMAN (Woffelman, 2004)
7. Ensembl (Hubbard et al., 2002)
8. GAP (Bonfield, Smith, & Staden, 1995)
9. GeneViTo (Vernikos et al., 2003)
10. Genotator Browser (Harris, 1997)
12. MEGA (Kumar, Nei, Dudley, & Tamura, 2008)
13. NCBI Map Viewer (Wheeler et al., 2005)
15. Primer3 WWW Interface (Rozen & Skaletsky, 2000)
16. RegulonDB (Salgado et al., 2001)
18. Sequencer (Gene Codes Corporation, 2003)
19. SeqVista (Hu et al., 2003)
20. UCSC Browser (Karolchik et al., 2002)

10 References


Mohammad Daradkeh, Alan McKinnon, and Clare Churcher
PO Box 84 Lincoln University
Lincoln 7647 Canterbury
mohammad.daradkeh@lincolnuni.ac.nz, {mckinnon, churcher}@lincoln.ac.nz

Abstract
Models play an important part in the decision-making process. However, due to uncertainty in a model’s input variables, making decisions involves a degree of risk. We have developed two visualization prototypes for exploring the influence of uncertainty in the values of the input variables on the risk associated with the decision-making. The first prototype is the interactive tornado diagram, which is considered as an extension to the static tornado diagram. The second prototype is the Uncertainty Influence Explorer (UIExplorer). This paper presents and discusses the results of an experiment conducted to assess the efficacy of these prototypes and compare their ability to help people answer meaningful questions related to the risk associated with decision-making. The results show that participants using UIExplorer performed better in terms of accuracy and time taken to complete the questions. Also, they found it easier to use and had higher confidence in the decisions being made.

Keywords: Information visualization, Interaction Design, Decision-making process, Uncertainty, Risk, Sensitivity analysis.

1 Introduction
All decisions are intended to bring about some future benefit to someone or something, and involve choices (e.g. whether to buy a new machine, whether to implement design A or B, etc.) (Willows & Connell, 2003). Systematic approaches to decision-making usually involve models which give a quantitative estimate of the value for the decision-maker to base a decision on (Clemen & Reilly, 2001). For example, financial managers use net present value and internal rate of return for analysing investment alternatives (Dayananda, Harrison, Herbohn, & Rowland, 2002; Jovanovic, 1999). In another decision context such as water management, decision-makers use more complex models to rank multiple water management options or compare the frequency and extent of various flooding events (Hyde, Maier, & Colby, 2005; Xu & Tung, 2008; Xu, Tung, Li, & Niu, 2009).

There are different kinds of decisions (Harris, 1998); some decisions such as which company’s shares to buy, involve making a choice among alternatives while others such as whether or not to invest in a new business are more “yes/no” decisions. Whatever the type of decision, the decision-maker can never be certain of the values of the variables or parameters used in the model and there may also be errors or approximations in the model itself (Jovanovic, 1999). For this reason, descriptions of the decision-making process include a sensitivity analysis step once the “best” decision has been identified (Clemen & Reilly, 2001; Larichev & Moshkovitch, 1995; Roy & Vincke, 1981). This is evident in Figure 1 which shows a typical decision-making process (Clemen & Reilly, 2001).

Figure 1: A Decision Analysis Process flowchart (Clemen & Reilly, 2001).

Many researchers have emphasized the role of sensitivity analysis in decision-making (Clemen & Reilly, 2001; French, 1986; Triantaphyllou, 2000). After choosing the “best” alternative and following the completion of the decision analysis, sensitivity analysis should be carried out to investigate how uncertainty in the input variables and criteria weights (preferences) affects the values of decision criteria, as well as the final ranking...
of alternatives (Clemen & Reilly, 2001). Several sensitivity analysis methods have been proposed (Guillen, Trejos, & Canales, 1998; Hutton & Charles, 1988; Mareschal, 1988; Ringuet, 1997; Soofi, 1990; Triantaphyllou & Sanchez, 1997). These methods are carried out to investigate the relationship between changes in the criteria weights and the subsequent alteration that occurs in the ranking of alternatives (Hyde et al., 2005). Despite this, decision-making processes are often applied with little consideration given to uncertainty in the input variables and propagation of such uncertainties through the decision model (Basson & Petrie, 2007; Xu & Tung, 2008). We suggest this is because the way in which the sensitivity analysis step should be used as part of the overall decision-making is not at all clear.

We are investigating the use of visualization tools to enable the consideration of uncertainty to be an integral part of making the decision rather than treating it as an add-on step which does not have a clear role in the process. To date we have applied these ideas to "yes/no" decisions. Application to decisions involving multiple alternatives will be the subject of later work.

We have proposed three visualization tools to allow the consideration of uncertainty to be better integrated into the decision-making process for “yes/no” decisions. The first visualization we investigate is the well known static tornado diagram (Cooke & Van Noordwijk, 2000; Koller, 2005, 2007) which is commonly used to explore sensitivity in financial decision-making. For the second visualization we have added interactivity to the tornado diagram. The third visualization is a prototype of our Uncertainty Influence Explorer (UIExplorer). It has been developed to explicitly allow the decision-maker to explore how the risk of making an undesirable decision is affected by the uncertainty in the input variables as depicted by the process shown in Figure 2. The decision-maker specifies the risk criterion to be used (e.g. that the NPV is less than zero) and also the uncertainty range for each input parameter.

This paper describes an empirical evaluation and presents its results. We introduce an experimental design with two parts: a quantitative part that measures and compares the performance of participants during the experiment. Two variables of performance (accuracy of the answers and time to complete questions) for each visualization were measured. In the qualitative part, the participants answered a number of questions regarding their experience and satisfaction with these visualizations in the decision-making process.

This paper is structured as follows. Section 2 briefly describes the visualizations using a widely used financial model for investment decisions called Net Present Value (Dayananda et al., 2002; Jovanovic, 1999). Section 3 describes the evaluation. Section 4 presents and discusses the experiment results. Future work and conclusions are discussed in Section 5.

### 2 Description of the Visualization Tools

In this section, we describe the visualizations using a widely used financial model for investment decisions called Net Present Value (Dayananda et al., 2002; Jovanovic, 1999). NPV is a financial model used in capital budgeting to analyse the profitability of an investment or project. A positive NPV indicates that the investment is acceptable. If NPV is negative, the investment should properly be rejected. However, there are many estimations and subtle interactions between variables that have significant effects on the profit outcomes. A basic version of calculating NPV is given by equation 1:

\[
NPV = \sum_{t=0}^{\infty} \frac{C_t}{(1+r)^t}
\]

Where
- \(t\) is the time of the cash flow.
- \(N\) is the total time of the project.
- \(r\) is the discount rate (the rate of return that could be earned on an investment.)
- \(C_t\) is the net cash flow (inflow minus outflow) at time \(t\).

#### 2.1 Static Tornado Diagram

A tornado diagram is a pictorial representation of the contribution of each input variable to the output of the decision making model (Clemen & Reilly, 2001; Cooke & Van Noordwijk, 2000; Koller, 2005). It consists of stacked horizontal bars, each one associated with one input variable. Each horizontal bar represents the range of the output (NPV) as the corresponding variable is varied over its specified range while all other variables remain constant at their nominal values. The length of the bar indicates the variable's effect on the model output. The model output has a nominal value which is calculated for the nominal values of all the input variables and displayed as a vertical line on the diagram.

A typical diagram is shown in Figure 3. The left and right bar ends indicate the corresponding upper and lower bounds of NPV as the related variable is varied within its specified range while the other variables remain constant at their nominal values. The example in Figure 3 shows that, for the given values of the other parameters, the NPV is mostly influenced by varying the inflow, while the variation in the outflow has little effect on NPV. One of the main drawbacks of the tornado diagram is that it assumes all of the input variables are independent. Thus, it ignores the influence of the interaction between input variables that might have a significant effect on the output (Koller, 2005). In addition, it is a static representation of
the sensitivity and thus it doesn’t allow users to interactively explore and compare possible outputs under different scenarios.

![Tornado Diagram](https://via.placeholder.com/150)

**Figure 3: The static tornado diagram shows the sensitivity of NPV to the variation in each input variable while other variables are held constant (Source: own Figure).**

### 2.2 Interactive Tornado Diagram

This visualization prototype is an extension of the static tornado diagram where users can change the nominal values of the input variables interactively and so explore the effects of the interaction between input variables on the model’s output. As shown in Figure 4, this can be investigated by varying an input variable (with its scroll bar) and observing how the bars on the diagram for the other input variables change. For example, if the user scrolls the inflow scroll bar, he/she will notice that the length of the rate bar will change. This means that the uncertainty in the model’s output resulting from the variation in the rate variable is affected by the inflow value, as can be seen from Figures 4b and 4c. In this way, the prototype can help in exploring the interaction between input variables on the overall output uncertainty.

One drawback with the interactive tornado diagram is due to how the displayed information is calculated. The calculation used to display the horizontal bars depends on varying one input variable at a time while keeping the other inputs at constant values. The problem with this way of calculation is that, in most cases, the true influence on the output variable depends not only on the values of the input variables, but also on the interaction between input variables over all ranges. Thus, varying one input variable while keeping the others constant might not, reveal the true influence of any input variable. This led to the development of another prototype that rectifies this drawback.

### 2.3 Uncertainty Influence Explorer (UIExplorer)

The aim of this prototype is to visualize the risk due to uncertainty in the model’s input variables, and its sensitivity to the variation in these input variables. The UIExplorer uses colours to convey the risk magnitude (in this case the probability of making a loss). Figure 5 shows a screenshot of the prototype. Yellow means no risk (i.e., probability of making a loss =0) and dark orange represents the highest risk (i.e. probability of making a loss =1). The colour of a cell shows the risk associated with that value of the input variable. It is calculated by taking every possible value of all the other variables and calculating what proportion of these combinations will result in a loss. The user can retrieve the numerical value of the risk (i.e., the probability of a negative NPV) by pointing to any of the cells. For example if the inflow is $30000 (highlighted cell in Figure 5) then if we consider all other possible combinations of values for the other variables about 44% will result in a loss (probability 0.44).

By looking at the displayed range of colours that represent the risk magnitude, the user can quickly and easily see where the investment is potentially risky or where it is not. In addition, he/she can readily see the values of the input variables for which the investment will likely be profitable (i.e., low risk). For example, in Figure 5, if the inflow varies within the range [$33000, $35000], the user can have confidence that the NPV will be positive (i.e., there is no risk of making a loss if the other variables stay within the given ranges).

Clicking on a cell in the first grid fixes the value of that input variable. The chosen cell is highlighted, and the colours in the lower grid change to convey a new range of risks associated with the values of the other input variables. For example, the lower grid in Figure 6 shows the risk associated with the values of inflow, outflow, and rate based on fixing the Initial Investment at $90000 and allowing the other input variables to vary within their ranges. The new range of colours in the lower grid represents the recalculated risk which is calculated by fixing the initial investment at $90000 and taking every possible value of the other variables and calculating what proportion of these combinations will result in a loss. This is useful when the user wants to fix a value for a certain variable and explore the risk range associated with the other input variables.

The range of input variables can be modified by changing their minimum and maximum values and as a result, the range of colours in both grids will change to convey the new values of risk. For example, Figure 7 shows the effect of changing the range of Inflow, from ($25000, $35000) to ($20000, $30000). It can be seen from Figure 7 that the colours in the upper grid are darker which means that the risk is very high for almost all of the values of the input variables. However, this does not mean that the risk associated with all possible scenarios is very high. For example, when the user chooses the value $30000 from inflow (the highlighted cell), he/she can notice from the lower grid that there are some combinations of the other variables which have low or no risk. This is helpful when the decision maker wants to assess the risk of choosing different scenarios and allows him/her to answer many different “what-if” questions.
Figure 4: (a) Screenshot of interactive tornado diagram. (b) The influence of decreasing the Inflow value on the Rate bar length. (c) The influence of increasing the Inflow value on the Rate bar length.

Figure 5: A screenshot of the UIExplorer showing the ranges of risk.
In this version of UIExplorer, we display the information in a uniform grid. The use of a grid layout facilitates the presentation of uncertainty and the associated risk in an organised way. In addition, it makes it easier to see and follow the change in the risk magnitude across the rows, which in turn facilitates the discovery of trends and relationships between uncertainty and risk. In addition, all input variables are bounded by a known maximum and minimum and, for this case study, we have made the assumption that all values in between occur with equal likelihood. Therefore, they can be mapped onto equal-sized cells. This way, the user can run through or compare several scenarios with various values and easily determine the risk associated with each value or scenario.

In this version of UIExplorer there is a limit to the number of divisions. We have divided each input variable into eleven divisions although there is no specific reason for that number. However, for finer-grained analysis and representation the user can change the min and max values. This facilitates more detailed assessment over a small range, and consequently more precise and effective decision making. In the future versions we plan to make the UIExplorer more flexible and amenable to larger number of variables.

Colour was chosen for the purpose of presenting risk because it is widely used for risk visualisation and communication (Bostrom, Anselin, & Farris, 2008). Also it is an important visual attention guide that can highlight levels of risk (Wolfe & Horowitz, 2004). Several studies have addressed the influence of colours on risk perception and decision-making processes (Lipkus & Hollands, 1999; Rogers & Groop, 1981; Soldat & Sinclair, 2001; Wogalter, Conzola, & Smith-Jackson, 2002). A study by Wogalter et al. (Wogalter et al., 2002) supports the following hierarchy of colours to convey risk level: red/dark orange riskier than yellow, yellow riskier than green. Scaling based on lightness or brightness may be helpful in presenting risk information (Bostrom et al., 2008). Davis and Keller (Davis & Keller, 1997) asserted that using colour hue and colour value are the “best candidates” for presenting risk information using static methods. Brewer (Brewer, 2006) advises use of light-to-dark colour for low-to high values with a constant hue.

### Evaluation

We evaluated the visualizations by testing and observing how people used them in a controlled experiment. The aim was to determine whether answering questions using the three visualizations would differ with respect to accuracy, completion time, ease of use, and confidence in the decisions that have been made. Specifically, we try to address the following questions:

1. How easily and quickly can users assess the magnitude of risk of making a loss associated with a certain value or a range of values?
2. How easily and successfully can users answer a variety of “what-if” questions?
3. Can users successfully use the interactive options to assist the risk assessment process?
4. How confident are users in the decisions they make using these tools?

In the following sections, the method of the evaluation will be explained, and then the results will be discussed.

### 3.1 Experimental Design

For this experiment, the Net Present Value (NPV) model is used. The four input variables are Initial investment, cash inflow, cash outflow, and discount rate. The output is the net present value, calculated over a number of years. We put the participant in the situation of deciding whether or not to make an investment as follows:

“You are planning to make an investment and you need to make a decision based on the value of the NPV. You are uncertain about the exact values of the model’s input variables so there is a risk involved in your decision. You need to deal with this uncertainty and assess the risk of your decision. The risk here means the probability of making a loss.”

### 3.2 Participants

We recruited 10 participants from the Lincoln University community. All participants had some understanding of the use of financial models and information to analyse and interpret data relating to business activities. Five were undergraduate students; all of them have enrolled in the Financial Information for Business Paper (ACCT103). One MSc student, with research experience in developing models for financial forecasting using neural networks, also participated. The other four participants were PhD students in the Faculty of Commerce studying marketing, business management, finance, and accounting respectively. The latter participant was also working as a part time lecturer in accounting. Of the sample, 6 were male and 4 were female ranging in age from late teens to 40+. All of them agreed to spend 30 minutes with our experiment and receive a $20 voucher in compensation. The motivation behind recruiting students from Faculty of Commerce is their good knowledge and understanding of the basic NPV model.

### 3.3 Procedure

The method used in the experiment was as follows: The participants filled out an entrance questionnaire to determine their background experience. It asked the participants to rate their experience in three areas on a four-point scale (1:None, 2:Beginner, 3:Intermediate, 4:Advanced). Figure 8 shows the average rating of participants’ familiarity with financial modelling, decision making under uncertainty and risk assessment. Participants rated their familiarity with financial modelling between none and intermediate, but rated their familiarity with decision making under uncertainty and risk assessment between beginner and intermediate level. This suggests that the participants were well placed to evaluate the visualizations and would have little trouble understanding the scenario.

![Average profile of participants](image)

**Figure 8: Background of participants.**

The participants were given a brief introduction to each visualization and the experimental method and were then asked to answer a set of questions using each visualization. During this time they were observed and their performance was recorded. Upon completion of the questions, the participants were asked to complete an exit questionnaire. Participants took part in the experiment individually with an observer present to record the time taken to answer the questions.

### 3.4 Test Questions

The task of the participants in the experiment was to answer questions about the impact of input uncertainty on the risk associated with making a decision for the given scenario. The questions addressed tasks common to the decision making process and required finding facts and information to answer them correctly. The five questions below were repeated for each visualization:

**Q1.** For the displayed ranges, which variable do you think has the most effect on the risk of making a loss? (Purpose: to find the most influential variable on risk)

**Q2.** Approximately, for what range of cash inflow can you be assured that the NPV will stay > 0 i.e. there is no risk of making a loss? (Purpose: to find a range of values associated with a specified risk)

**Q3.** What do you think the risk of making a loss will be if the Discount rate becomes 10%? (Purpose: to determine the risk associated with a specified input value)

**Q4.** Given that Initial investment is fixed at $90000, approximately what is the minimum Cash Inflow that will ensure a positive NPV? i.e. probability of making a loss is zero. (Purpose: to find a value within a particular scenario)

**Q5.** If the Inflow is $30000, what is the range of rate values that will ensure no risk of making a loss? (Purpose: to determine a range of values resulting in a specified outcome).

In addition to the total time to complete questions using each visualization and the accuracy of answers overall, we looked at each question individually. Each participant was asked to rate the ease of use of each visualization to answer each question on a four-point scale (1-Very difficult, 2-Difficult, 3-Easy, 4-Very easy). Participants were also asked to explain why they found a
question easy or difficult to answer. After completing the questions for each visualization, participants were required to rate their confidence in the decisions they had made with these visualizations on a 5 point scale (where 1 meant Not Confident and 5 Highly Confident).

4 Evaluation Results

The results revealed considerable differences between the visualizations.

4.1 Accuracy

In terms of the accuracy of answers, the results showed that there is a noticeable difference between the three visualizations. Figure 9 shows the number of accurate answers for each question using each visualization. 7 out of 10 participants answered the question on finding the most influential variable on risk (Q1) correctly using the three visualizations. This is because this question doesn’t need interaction from participants to find the answer. For the questions on finding a value or range of values (Q2 and Q4), the static tornado diagram failed to give very accurate answers. This is because these questions could not be answered without interaction. The number of correct answers for Q2 and Q4 increased for the interactive tornado diagram. However, moving the sliders did not give high accuracy (5 out of 10 were incorrect for both Q2 and Q4). Using UIExplorer, participants gave more accurate answers for Q2 and Q4 than using static tornado or interactive tornado diagram.

For the questions on determining the risk associated with a certain value (Q3) or the risk associated with a range of values within a scenario (Q5), the results show that participants did better using UIExplorer than using both the static tornado and the interactive tornado. This suggests that both interactive tornado and UIExplorer helped in finding precise answers to the questions that are related to finding the risk (Q3 and Q5). On the other hand, interactive tornado failed to give high accuracy for the questions that are related to finding the values that affect the risk (Q2 and Q4), while UIExplorer succeed to give high accuracy for the same questions, as shown in Figure 9.

4.2 Time Taken

Participants took from 5 to 10 minutes to complete the five questions using each visualization. Figure 10 shows that the mean time taken to answer the questions was 7.3 minutes using static tornado, 6.2 minutes using interactive tornado, and 6 minutes using UIExplorer. The mean time was shorter for UIExplorer than interactive tornado perhaps because, as some participants expressed, moving the sliders in interactive tornado prototype to find the answer takes longer than observing the colour change in UIExplorer. Table 1 summarizes the results of the time taken by the participants to answer the questions using each visualization.

![Figure 9: Comparison between the three visualizations in terms of accuracy.](image)

![Figure 10: The average time taken by participants to complete the questions using each visualization.](image)

<table>
<thead>
<tr>
<th>visualization</th>
<th>average</th>
<th>sd</th>
<th>min</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static tornado</td>
<td>7.4</td>
<td>2.1187</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Interactive tornado</td>
<td>6.2</td>
<td>1.3166</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>UIExplorer</td>
<td>6</td>
<td>1.1547</td>
<td>5</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 1: Summary of the time taken Results.

4.3 Ease of Use

Figure 11 shows that there is a clear distinction between the three visualizations in terms of their ease of use. While it was difficult to find answers using static tornado, it becomes easier using interactive tornado and UIExplorer. For example, using the static tornado prototype, participants found it difficult to obtain answers to the questions related to finding the values of the input variables (Q2 and Q4) and the risk associated with these values (Q3 and Q5). However, using the interactive tornado prototype, participants found that the search for the answers became easier and even much easier when using UIExplorer.

Using the static tornado prototype, participants found that the question related to finding the most influential input variable on the risk (Q1) easier than the other questions. On the other hand, although the ease of use for the interactive tornado prototype and UIExplorer was rated between very easy and difficult, users found UIExplorer much easier than the interactive tornado. This was consistent with the feedback from the participants, who mostly expressed their satisfaction with the use of UIExplorer.
On average, participants found UIExplorer easier to use than the static tornado and interactive tornado as shown in Figure 12. Table 2 summarizes the results of the participants’ evaluation to the ease of use. It is noted here that two of the participants who answered Q2 and three participants who answered Q4 incorrectly assessed finding the answers as easy. This gives the impression that the interactive tornado prototype is misleading because some of the participants have been unable to identify the correct answers despite what they perceived as a reasonable ease of use.

Some of the participants have been unable to identify the correct answers despite what they perceived as a reasonable ease of use.

Figure 11: Comparison between the ease of use for each question in each visualization.

Figure 12: A comparison between the three visualizations in terms of ease of use.

Table 2: Summary of the ease of use results.

### 4.4 Level of Confidence

Results summarized in Table 3 show that there is a clear difference in the confidence of the participants for the decisions that can be made based on the three visualizations. Participants showed a low level of confidence in their decisions using the static tornado prototype (2). This level rose noticeably for the interactive tornado prototype (3.7) and rose to (4.3) for UIExplorer. One of the main features of UIExplorer, as most of the participants expressed, is that the risk associated with each value can be directly seen without the need for further interpretation. It should be noted that although some of the participants responded with incorrect answers, their level of confidence using the interactive tornado prototype was high. This again indicates that the interactive tornado prototype leads to misleading interpretations concerning the risk associated with the decision making.

Table 3: confidence level results.

### 4.5 Post-study Questionnaire

Eight of the participants found UIExplorer easier to use, more intuitive, and quicker to understand and learn. They understood the representation of risk by colours. On the other hand, two participants preferred the interactive tornado diagram because they found that moving the sliders to reach the required values was intuitive and lead to more understanding and easier assessment. This is supported by their correct answers when using the interactive tornado prototype.

We didn’t notice that participants faced a problem in distinguishing colours/gradients. Participants were able to relate between the degree of colour and the degree of risk. They were also able to understand the relationship between uncertainty and risk through gradations of colours. This is consistent with the results of the experiment which showed that the overall performance of participants was better using UIExplorer.

### 5 Conclusion and future work

This paper presents the results of an experiment conducted to compare the ability of three visualization prototypes to help people explore the influence of uncertainty in the input variables on the risk associated with the decision making. The first visualization is a static tornado diagram. The second visualization is an interactive tornado diagram, which is a modification of the static tornado diagram. The third visualization is the uncertainty influence explorer.

The results show that most of the participants preferred to use UIExplorer rather than the static or interactive tornado prototypes. The use of UIExplorer leads to more correct answers and a shorter time taken to find those answers. Participants found UIExplorer easier to use and they had greater confidence in their decisions compared with the static tornado and interactive tornado prototypes. This was consistent with the quantitative results. Participants’ feedback confirmed that further research is needed to improve the design of the UIExplorer so that the user can explore the risk associated with the decision making models at several levels of detail. It is important for the decision maker to be able to explore the risk associated with each value of each input variable, the risk associated with each input variable regardless of the
current value of that variable, and the risk associated with each scenario.

Although there were only 10 participants in this study and we did not completely get rid of learning effects and bias in the results, we believe that the results clearly indicate that our approach of including the risk of making an acceptable decision as an integral part of the decision-making process has significant merit. However, further evaluation is needed. More extensive user evaluation would include more participants and more tasks. In addition, we intend to develop UIExplorer further and also explore ways of extending the approach to decisions where there are a number of alternative options.

Acknowledgements
We would like to thank all participants without whom the study would not have been possible.

6 References


http://dx.doi.org/10.1016/S0098-3004(97)00012-5


iAnnotate: Exploring Multi-User Ink Annotation in Web Browsers

Beryl Plimmer, Samuel Hsiao-Heng Chang, Meghavi Doshi, Laura Laycock, Nilanthi Seneviratne

Department of Computer Science
University of Auckland

beryl@cs.auckland.ac.nz, {hcha155|mdos002|llay008|lsen008}@aucklanduni.ac.nz

Abstract

We present iAnnotate, a tool that provides multi-user digital ink annotation on standard web pages within a commercial browser. The annotation can be saved, retrieved and shared with others via a URL. In addition multiple users’ annotations can be displayed on the same page. We describe our design goals and the technical challenges. While realizing annotation on web documents is difficult because of the dynamic nature of the documents and the security constraints of web browsers, our user evaluation suggests that fully realized digital ink annotation tools would be very valuable.

Keywords: Digital ink, document annotation, web annotation

1 Introduction

Annotating a document with a pen helps people think and engage more deeply with the material (Wolfe 2000). Furthermore annotations can be used to communicate ideas about the document content with others (Marshall 1997). Increasingly documents are web-based. Yet, although the computer hardware to annotate documents (digital pen and touch interfaces) is freely available, there are few examples of ink annotation functionality in web browsers.

Digital ink annotation support is available in many desktop applications such as word processors. However web browsers do not offer built-in annotation functionality of any kind. In terms of retrieving and sharing information, web browsers are probably used more than desktop applications. The existence of annotation functionality, enhanced with storing and sharing, should further increase the convenience offered by the web.

Web documents and web browsers present particular challenges that make annotation difficult. First, web documents are usually dynamic. That is the content is changing quickly and web documents are designed to reflow within the available window space. Annotations derive meaning from their location, therefore robust location fixing of the annotation within the dynamic document is required. Second, the web browser is on the user’s machine and from this machine it interacts with data from unsecure sources. As such browsers present a security risk and must have excellent security. The security of browsers limits the extension points for add-ins and the ability to mix content between different servers. Finally, realizing digital ink annotation is technically quite difficult as text, pictures and annotations must lie over each other. This usually requires layering of the interface which can be problematic as accurate position information of elements on the different layers is not always available.

Several projects have investigated user annotation of web pages (e.g. Cadiz, Gupta et al. 2000; Chatti, Sodhi et al. 2006). However, none of these applications provide a complete solution. Some lack the flexibility to allow annotations to adapt to the changes of the underlying webpage; others do not save the ink or lack support for window scrolling.

Digital ink, created with a stylus directly onto the display, simulates real world annotation behaviour by enabling users to directly draw on web pages. The interaction is straightforward. However a number of complex problems are faced from an implementation perspective. In this project we investigate whether Silverlight (Microsoft Corporation 2008), a recently released browser plug-in with digital inking capabilities, provides sufficient functionality to support freehand annotation on any web page.

2 Related Work

Current implementations of annotation on the web can be divided into two types: text based annotation applications (Wilson ; Kahan, Koivunen et al. 2002; Bottoni, Civica et al. 2004) and ink based annotation applications (Ramachandran and Kashi 2003; Chatti, Sodhi et al. 2006). Text based annotation applications allow the user to annotate web pages using the mouse and keyboard whereas ink based annotation applications allow the user to utilize a stylus or touch screen (or mouse) to annotate a webpage with digital ink. Both approaches need to be able to capture and display the annotations, store and retrieve annotations, and finally allow for changes in the underlying webpage and size of the browser window.

An important aspect of annotation is deciding how to acquire the annotation from the user. In text based annotation systems, this entails the user typing the annotation using a keyboard. In digital ink annotation the user is able to annotate directly on top of the document, as if it were paper.

Text based annotation applications generally require the user to select text, right click the selected text or click a button to generate a text field to enter the annotation (Figure 1). This can be unintuitive and also breaks the thought process behind the annotation. This means that
the user’s concentration is on how to insert the annotation rather than the actual content of the annotation.

When the underlying web page is changed via insertion or deletion, the annotation application should recognize this and display accordingly. In the Avaya prototype (Ramachandran and Kashi 2003) and uAnnotate (Chatti, Sodhi et al. 2006) if the underlying content is modified, the annotation is not displayed. This is acceptable if the keywords for anchoring the annotation have been deleted. However if there is an insertion before the keywords the annotation application should still display the annotation. Annotea (Kahan, Koivunen et al. 2002) does not display orphaned annotations (annotations where the keywords have been deleted) on the web page however they will display it in a list view where all annotations associated with the page are displayed. Current approaches provide approximate reflow in common situations. The dynamic nature of web documents and browsers makes robust annotation reflow a difficult problem.

The transient nature of web pages means that the annotation needs to be anchored properly in order to allow annotation reflow. In order to generate the anchoring points Wang and Raghupathy (2007) describe methods in which the anchoring position can be determined by the context of the annotation. This means that the annotation anchoring position should be determined by the type, as well as the position of the annotation. Another approach to attaching annotations to the underlying document is described by Priest and Plimmer (2006). This method uses a linker, which is the first stroke of the annotation. The linker stroke can either be a circle stroke or a line stroke identifying the document position to which the annotation is attached.

In order to create an effective web annotation application, annotations need to be encapsulated and stored efficiently so that they may be reused at a later date. Annotations are usually encapsulated either using the resource description framework (RDF) or extensible mark-up language (XML) schema. The common annotation framework (Bargeron and Moscovich 2003) and the Avaya prototype (Ramachandran and Kashi 2003) both use a customized XML schema to encapsulate attributes of annotation such as the anchors for the annotation, the annotating document and the annotation content. Annotea (Kahan, Koivunen et al. 2002) uses the RDF framework to create classes of annotations that can give fine grained detail of annotation types. In either case, the annotation can be formed so that it is self-contained. This means that the annotation can be stored on a different server than the web pages. Thus it promotes portability of annotations, and also provides an approach to annotate read-only documents (Bargeron and Moscovich 2003).

Most web annotation applications prefer to encapsulate annotations via XML and save these files in servers (Kahan, Koivunen et al. 2002; Ramachandran and Kashi 2003; Bottoni, Civica et al. 2004; Chatti, Sodhi et al. 2006). There are number of advantages in storing annotations in this manner: all annotations are hosted in one place thus preventing users from loading annotations to the wrong web page. Annotea (Kahan, Koivunen et al. 2002) and MADCOW (Bottoni, Civica et al. 2004) take advantage of this, incorporating multiple servers in their architecture. However, a drawback of this approach is

---

**Figure 1: Annotating with Annozilla (Wilson)**

Digital ink annotation bypasses this aspect of annotating because the user is able to directly write on the webpage. This way the user can concentrate on the content of the annotation. Although this process can be achieved via a mouse it is most effective when using a stylus for input.

Displaying the annotations to the user is as important as acquisition of the annotation. In a text based annotation application this usually is achieved by displaying an icon for the annotation—the user double clicks the icon in order to view the annotation. Annotation applications that support digital ink can display the annotation more easily directly on top of the webpage. For example, uAnnotate (Chatti, Sodhi et al. 2006) uses a transparent Flash object so the user is able to view the created annotation as well as the content of the underlying web page.

However, in both cases it is possible to cover the content of the document with annotations. This becomes a problem in digital ink annotations, because the ink is displayed on top of the content. Therefore functionality to hide or filter annotations is required. Annotea (Kahan, Koivunen et al. 2002) allows the user to filter and hide annotations according to author, server or annotation type. uAnnotate (Chatti, Sodhi et al. 2006) provides a link which the user can click to clear all annotations on the web page.

Annotations derive meaning from their position so it is important to retain their relative position when the underlying document changes. Moving the ink to retain its meaning is called ink annotation reflow. Brush et al. (2001) in a study regarding the repositioning of annotations concluded that users prefer the annotation to be anchored onto specific key words. For example if the document is changed and only a small amount of the annotated text is remaining, the user expects the remaining text to be still annotated.

uAnnotate (Chatti, Sodhi et al. 2006) approaches the problem of repositioning when scrolling by making the ink overlay as large as the webpage. This means that when the user creates an annotation and scrolls down the page the annotation scrolls with the content. If the user resizes the window then the overlay as well as the content resizes to fit the window.
that each server uses an API to retrieve and store annotations. This means third parties cannot extend the implementation unless they are provided with this API.

As an alternative Denoue and Vignollet (2001) propose that annotations should be stored within the URL of the webpage. As well as being self-contained, it avoids the use of a server altogether. However this approach to storing would make ink annotation storage difficult due to browser limitations such as the character limit for the URL. Also, it does mean that users are unable to annotate on third party web pages due to restricted domain access.

Another option is to store the annotations locally. uAnnotate (Chatti, Sodhi et al. 2006) uses Flash’s Local Shared Objects to store information about its annotations. The main drawback of this approach is that annotations are restricted to one computer and one user.

If the user wishes to view annotations which were saved at a previous date, he or she could import this annotation onto the same webpage. For example in uAnnotate (Chatti, Sodhi et al. 2006) it is possible to export the created annotations into an XML file and reload this file to a web page at a later date. In this case the annotation application should check whether the URL of the annotating document and the URL of the web page are the same. If the annotation is saved onto a server, this server should prevent the user from loading an annotation which was not created for that particular web page.

Some of the constraints of existing digital ink annotation are that the annotation pane is restricted to the current window (Chatti, Sodhi et al. 2006), the annotation can be loaded onto any URL (Chatti, Sodhi et al. 2006), and saving is local only or manual from the user’s clipboard (Chatti, Sodhi et al. 2006). Reflow is either not supported (Chatti, Sodhi et al. 2006) or limited (Ramachandran and Kashi 2003). Annozilla (Wilson) supports multiple text annotations but only one can be viewed at a time: we are not aware of any digital ink annotation tools that can show multiple users’ annotations at the same time.

3 Design Goals

Our vision for this project is a web space where a document can be annotated with digital ink by any number of people. The annotations may have different purposes: for example they may be a teacher’s or student’s comments on course notes, a study group’s shared considerations of a text, or an extended family’s shared annotations on a family calendar. Potentially, they could contribute to social discourse where an initial document acts as the trigger for a discussion. Realizing this vision requires us to explore a number of technical and design issues.

Many-to-many annotations raise several research questions: how can we appropriately visualize, index, store, search and filter the annotation data? Imagine a web document that 10, 20, or 100 people have annotated. Simply displaying the annotations over the document is unlikely to be useful or usable. However seeing ‘hot spots’ (or ‘aggregated’ annotations), points in the document that a number of people have annotated, is likely to be useful to find foci of attention. Additionally filtering the annotations to see selected users’ (e.g. the boss/teacher) or groups’ (friends/colleagues) annotations or those on a particular theme or topic are also likely be useful. If the annotations are a conduit for social discussion then being able to retrieve the annotations with appropriate spatial and temporal information is also crucial. One can imagine converting annotations into a time series so that a viewer can replay the conversation (or the time series could simply be used as a mechanism to see individual annotations in a hotspot).

In this first stage of the project we explore the technical issues of realizing multiple, shareable digital ink annotations on any web page (not just specially designed pages that incorporate annotation) and basic usability considerations for digital ink web annotation.

4 Our Approach

As there are well-documented limitations to current approaches such as Flash, in this project we have investigated Silverlight. Microsoft Silverlight (2008) is a cross browser and cross platform plug-in which allows developers to create rich internet applications. Of particular interest for this project is that it exposes an API to the .NET Framework that has excellent support for digital ink.

From a software architecture perspective there are two parts to realizing the annotation capability. The first is the core annotation framework, which provides ‘on document’ functionality of inking, anchoring and reflowing ink, and general ink editing support. The second is the browser extension for the instantiation of the ink overlay, and saving and loading of ink.

4.1 First Prototype

Our first prototype (Figure 2) used Silverlight 2 beta 2, released in June 2008. The core annotation framework provides functionality for the editing, anchoring and reflow of the digital ink. The browser plug-in provides a toolbar with start and stop annotating buttons and functionality to inject Silverlight into a page’s HTML.

4.1.1 Core Annotation Framework

The core annotation framework (CAF) of iAnnotate was built using Silverlight, JavaScript, XML and ASMX web services. In order to capture the annotation the CAF uses a Silverlight object. When the page is loaded this object is overlaid on the webpage. The Silverlight object is transparent, except the buttons, which allow the user to interact with the application.

The Silverlight object contains our InkPresenter control, which renders digital ink strokes using the movement of the stylus. The InkPresenter can also create strokes programmatically thus it is used to display previously saved annotations. When the user begins an annotation the Silverlight object calls JavaScript methods to anchor the annotation to the underlying webpage.

To anchor annotations effectively the individual strokes must be grouped into annotations. Two approaches were considered. The first serializes the ink strokes into a XML string and sends this to the server where they can be recognized and grouped into different annotations. However, this method may take a considerable amount of time as the strokes need to make a round-trip to the server.
The second method uses a bounding box to group ink strokes into annotations. The bounding box implemented is similar to that used by Priest and Plimmer (2006).

Figure 2: Prototype one

This bounding box is first displayed to the user when they start an annotation. As seen in Figure 3 all strokes (red ink) within the bounding box (blue rectangle) are classified as one annotation. If the user starts an ink stroke within the bounding box and reaches towards any edge of the bounding box the bounding box will expand in that direction to allow the user to carry on annotating.

Figure 3: Bounding box in CAF

The algorithm for grouping ink strokes checks if the first point of a new stroke is inside the bounding box of the current annotation. If it is the stroke is added to that annotation. If not, a new Annotation object is created and a bounding box created to enclose it. This algorithm can potentially group two separate annotations together. For example if a user circles one word in a sentence and tries to underline another word which is just within the bounding box, then the bounding box would expand to allow this stroke to be in the same annotation.

Once the ink strokes are grouped into annotations, anchor locations on the underlying webpage are computed. The position of the first point of the first stroke is used to identify the closest HTML element. The standard JavaScript method ‘elementFromPoint’ returns an HTML element given a point. However it only returns the first element enclosing the point. Thus when the CAF is on a webpage it always returns the Silverlight object and not the content elements.

We developed a JavaScript algorithm to determine the closest HTML element. This algorithm traverses through the Document Object Model (DOM) tree to find the vertically closest HTML element in the document’s body section. The algorithm matches the vertical position of the annotation to the document because the distance between...
the annotation and the top edge of the document describes the section that the annotation belongs to most accurately.

The main limitation of this algorithm is that if there are a large number of elements on the webpage it will iterate through all of the elements. This may delay the rendering of the first stroke. Another constraint is that the annotation is anchored to an element; JavaScript does not allow fine-grained control over the text in a webpage. However, attaching the annotation to the element has the advantage of supporting anchoring annotations to images as well as text elements.

Annotation reflow happens on two occasions: when the browser window resizes and when the underlying content of the webpage is changed. The algorithm processes each annotation by first locating its anchoring element. If it is an element other than the document element it computes the x and y offset of the anchoring element from the top of the page. This offset is used to move the annotation. In the case when the anchoring element is the document element of the webpage the ink annotation is rendered at the original position. If the anchoring element is not found the annotation is an orphan and is not displayed.

Our reflow algorithm does not properly take account of the horizontal position of an annotation. If the browser window is resized then the annotations can potentially be on the wrong place. More sophisticated reflow is necessary for a completely satisfactory system.

4.1.2 Plug-in

Our goal with the plug in is to extend a browser with a toolbar to provide the necessary interaction buttons (‘Start Annotating’ etc.). Silverlight works best with Microsoft Internet Explorer (IE), so as a proof of concept we concentrated our efforts on IE. iAnnotate’s plug-in separates into two main sections: the Browser Helper Object, and the iAnnotate toolbar.

An Internet Explorer extension can be created by using the Browser Helper Object (BHO). BHO allows the developers to access Component Object Model (COM) components that load each time the browser starts up. The BHO implements the IObjectWithSite interface to establish a COM-based communication channel to obtain the browser’s events.

The iAnnotate toolbar of the extension inherits the BHO. The BHO adds the band on which the buttons can be placed. iAnnotate adds the “Start Annotating” and “Stop Annotating” button in the band of the toolbar displayed on the browser, as shown in Figure 2.

On ‘start annotating’ the plug-in injects a Silverlight object into the webpage. Our first attempt injected the Silverlight object into the browsing pane of the browser: the area in which the webpage is displayed. While we could add the Silverlight object, the events were not fully exposed. Thus we could not progress with this approach.

The second approach injected a Silverlight object into every webpage the user wishes to annotate. To achieve this, the Microsoft HTML Object Library was used. This “mshtml” reference allowed access to the webpage’s HTML through the Component Object Model (COM). This access was obtained by creating an HTML document using the HTMLDocument class and casting the Browser’s document into the HTMLDocument type. When injecting HTML using the mshtml “insertAdjacentHTML(string where, string HTML)” method, the injection of a reference to an external JavaScript file does not work. We attributed this to security. However injection of inline JavaScript was effective.

We converted all of the JavaScript code into one string and added it using the “execScript(string code, string language)” method. Thus a clean architecture could be obtained and the extension could successfully inject the Silverlight object into any webpage the user would wish to annotate. There were additional problems with interfacing between the browser and server for saving and retrieving annotations that we had not fully resolved before Silverlight 2.0 was released.

Using this version of iAnnotate we undertook an informal usability study, which is reported in the evaluation section.

4.2 Second Prototype

The upgrade to Silverlight 2.0 (release version) in November 2008 caused significant problems with our plug-in architecture. Unfortunately the release version of Silverlight 2 disallowed applications to be retrieved from outside the site’s domain. As a result, although we could still insert code into web pages and call Silverlight from the page, because the domain of that page is different from the annotation server it will not respond.

After exploring various avenues we decided the only viable approach was an iFrame design. Similar to the previous approach, we have layers. The top layer is the annotation panel made with Silverlight 2.0, which allows the creation of digital ink. The middle layer is the IFrame, which displays the content of some webpage ready to be annotated. These layers sit on a HTML page, which is located server side, which means the Silverlight can be loaded without security issues.

Annotations can be saved and retrieved from a simple MySQL database on the server. Each annotation is tagged with the user’s ID, URL and time data. This allows annotations to be shared and multiple annotations of the same page displayed.
The user experience with this implementation is different. The user must login to the annotation website so that their annotations can be identified. In order to annotate a page the user must first put the URL in the textbox at the top of the page and click ‘go’ to load the page (Figure 4). They can then ink, highlight and erase using the buttons at the bottom of the window. On saving each annotation is automatically allocated a unique key and a URL is generated for the key (annotation URL with the key appended) (Figure 5). The annotation can be retrieved using the key or the URL. Using this generated URL the web page and annotation can be retrieved without logging in. Multiple annotations for the one page can be loaded by entering the key or selecting the annotation codes from the list and clicking the ‘load’ button (Figure 6). Annotations of a different page can be loaded with the page from the open list (Figure 7). This is a two-step process; first the page is loaded from its original location (we do not copy the page content to our server) then the annotation is rendered on the page using the CAF.
The disadvantage of this approach is that the user has to copy the URL of a page they want to annotate into the field on the annotation website. The annotation website then reloads the webpage into the IFrame. This is a less natural interaction paradigm than we would like. Furthermore it is difficult to calculate the size of a webpage before it is loaded consequently we have created a fixed (large) IFrame that can accommodate most standard web pages (there are some dynamic technologies such as AJAX that do not work). This effectively disables normal window resizing functionality as the IFrame is of fixed width. One benefit from fixing the width is that ink reflow is much less of an issue. Another advantage of this approach is that the user does not need to install a plug-in so the functionality is more freely available.

5 Evaluation

We have conducted user evaluations on both versions of iAnnotate. The usability study on the first prototype evaluated three main aspects of the system: the user interface, the appeal of digital ink annotation in a web browser and the users’ reactions to annotation reflow. The usability of the second prototype focused on the sharing of an annotation.

5.1 Prototype One

The study was conducted with eight student participants on a Tablet PC with stylus input. The participants had used a stylus at least once and were familiar with how a Tablet PC works. Most of the participants digitally annotated paper documents at least “sometimes” but “not often”.

The study consisted of two main parts. The first part was to check the usability of creating annotations. The participants were provided with a paragraph in a wiki which had several spelling mistakes. They were asked to mark-up the corrections with ink annotations (Figure 8 top). Once this task was accomplished, the participants were asked to rate the system for its user interaction and functionality.

The second part was to examine the ink reflow. The study conductor edited the wiki in front of the participant so they could notice the changes. This included moving the paragraphs from one point to another. The webpage is then refreshed and the annotations are reloaded, with the ink reflowed (Figure 8 bottom). The participants were requested to rate the system again after seeing this new functionality and asked what they thought of the reflow of the annotation. We also used this data to measure the accuracy of the ink reflow.

The key results showed that the participants found iAnnotate very easy and intuitive to use (4.71 on a 5 point Likert scale). The average accuracy of iAnnotate’s reflow functionality is approximately 73%, ranging from 58.33% to 83.33%. This affected the appeal of annotation on the web, which dropped from 4.71 after the first task to 4.42. Some participants mentioned that if they knew how the anchoring algorithm worked, they would be happy to adapt their style of annotating to increase the accuracy of the reflow.

5.2 Prototype Two

The evaluation of the second prototype focused on its new functionality: saving, sending and loading annotations and displaying multiple people’s annotations on the one page. The first task was to annotate a web page as an instruction guide to the basics of website layout for an 8 year old (Figure 4). The second was collecting menu choices from friends for a shared meal (Figure 9); we had
pre-prepared four annotations of an online menu site for this.

![Image of iAnnotate](image)

**Figure 9: Multiple user's annotations of a web page**

The 10 participants were all students (aged 19-26); half were computer science/software engineering majors and the other half were from various other disciplines. The gender split was 50:50 and pen-computing experience varied from a lot (4), some (4), never (2). Each participant was given a short demonstration of the main functionality before they attempted the tasks.

Unlike the first prototype, this prototype exists within the browser pane. The users had to first enter the URL of the web page they wanted to annotate into the text box and press ‘load’. Two participants initially put the URL into the browser address bar rather than the iAnnotate textbox and some participants looked for functionality such as loading or saving annotations within the webpage.

The basic functionality of iAnnotate proved easy and, for the most part, intuitive for users (mean 3.6, median 4 on 5 point Likert scale). The save, load and browse buttons were easily located by the majority of users in the study. The participants, even those who had not used a Tablet PC before, all found annotating the webpage to be easy.

The method for saving is likely to cause problems. The user has no choice of what to call the annotation; a random ten character code is assigned when the user saves an annotation. Most of the participants could not recall a code they had saved only minutes before and had to look it up. One participant commented “good but annotation codes were inconvenient”. Another suggested thumbnails as an alternative.

A few people interpreted the ‘Load’ functionality to be equivalent to the ‘Open’ functionality. ‘Load’ opens annotations that have been created on the web page currently open and it places them on top of existing annotations. ‘Open’ opens any annotation created by the user and its associated webpage in a new window. There needs to be a clearer distinction between ‘Open’ and ‘Load’.

We experienced a technical problem during the study with some larger annotations not always loading completely. The program gave no indication of failure. We believe that this was because of time-outs occurring during the round-trip to the server.

A small inconsistency identified was that while a user can input text into the URL box and press Enter to load the corresponding URL, this functionality is not duplicated in the behaviour of the Load tool. When a user inputs an annotation code into the Load box and presses enter, the page is cleared of any previously saved annotations. This may cause loss of data when the page is refreshed which would cause more work for the user and unnecessary frustration.

Most of the usability issues uncovered are not associated with the technical challenges of web annotation and are easy to correct.

The overall impressions of the tool were positive, with most users appreciating the potential usefulness of the product while understanding that this system still needs more development. Feedback included comments such as “fun tool to use” and “good application and has a lot of promise and use”.

6 Discussion

In this project our goal was to provide a tool to facilitate digital ink annotation ‘anywhere’ on ‘any’ webpage. To this end we have investigated the functionality provided by Microsoft Silverlight as it is a new browser add-in that supports digital inking.

We encountered numerous technical issues, which are not dissimilar to issues previously reported when trying to extend existing proprietary tools (Dietrich, Hosking et al. 2007; Chang, Chen et al. 2008). Nevertheless we had some success with the two prototypes. The first prototype supported inking and basic ink reflow. While the second added saving, loading and display of multiple users’ annotations.

The users in both evaluations studies enjoyed the experience and felt that such functionality would be useful. The experience was less compelling with the second prototype as it was a less natural interaction experience with the users having to paste the URL of the page they want to annotate into the annotation site before they could ink, save, send, etc. To partially address this problem we have coded a small JavaScript hyperlink that can reside in the user’s favourites that will take the user directly to the annotation server and display the current web page.

While Silverlight offers appropriate base classes for digital ink support, the manner of integration into IE is insufficient for our purposes. The way forward in this respect may be an open source browser so that the base code can be modified as required. It may also be better to use independent inking resources as it is quite probable that proprietary ones have other extensibility issues.

Our goal of being able to annotate any website cannot be fully realized with the current approach. Sites where the content is very dynamic (scrolling adverts for example) are likely to be different on a minute-by-minute basis. Also, there are other web technologies (such as Flash) that can result in multiple pages having the same URL. In this case our recorded URL may not take the user to the appropriate page. To retain information in these cases a copy of the page must be captured – this has obvious consequences for server space etc.
Digital ink reflow remains a difficult problem that has not yet been fully solved. The DOM model of HTML makes this more challenging with these documents. However we expect that as more digital ink annotation research is conducted more intelligent ink reflow approaches will be developed.

7 Conclusions
In this project we have explored general digital ink annotation support in web browsers. Our technology choice was Silverlight and IE. We developed two prototypes, both of which had promising features, but neither of which was a complete solution. To make more progress from a technical perspective requires a change in the browser API or the use of an open source browser. In spite of the technical challenges our usability studies suggest that digital ink annotation of web documents is something that would be useful.

8 Acknowledgements
We thank Microsoft Research Asia for funding this project.

9 References
Life-Sketch - A Framework for Sketch-Based Modelling and Animation of 3D Objects

Rong Yang Burkhard C. Wünsche

Graphics Group, Department of Computer Science
University of Auckland,
Private Bag 92019, Auckland 1142, New Zealand,
Email: wizard.yang@gmail.com, burkhard@cs.auckland.ac.nz

Abstract

The design and animation of digital 3D models is an essential task for many applications in science, engineering, education, medicine and arts. In many instances only an approximate representation is required and a simple and intuitive modelling and animation process, suitable for untrained users, is more important than realism and extensive features. Sketch-based modelling has been shown to be a suitable interface because the underlying pen-and-paper metaphor is intuitive and effective.

In this paper we present LifeSketch, a framework for sketched-based modelling and animation. Three-dimensional models are created with a variation of the popular “Teddy” algorithm. The models are analysed and skeletons with joints are extracted fully automatically. The surface mesh is bound to the curved skeletons using skinning techniques and the resulting model can be animated using skeletal animation methods.

The results of our evaluation and user study suggest that modelling and animation tasks are considerable more efficient than with traditional tools. The learning curve is very flat and a half page document was sufficient to familiarise users with the tools functionality. Users were satisfied with the automatically extracted joints, but some users struggled selecting the appropriate rotation axes and angles for animating the resulting 3D objects. A more intuitive, preferable automatic or sketch-based approach for animations is needed. Overall users were satisfied with the modelling capabilities of the tool, found most of its functionality natural and intuitive, and they enjoyed using it.

Keywords: sketch-based modelling, skeletal animation, skinning, human-computer interfaces

1 Introduction

The design and animation of 3D computer models is essential for many applications in science, engineering, education, medicine and arts. In many cases rough prototypes of a model are sufficient. For example, in the early production stages of professional animations storyboards are used to translate the story into images and organize scenes (Hart 1999). Improvements in computer technology have led to digital storyboards and sketch-based design tools (Landay & Myers 2001) and simple sketch-like interfaces for human character animation (Mao et al. 2006). In medical imaging and scientific computing rough prototypes (frequently termed default models) are used for segmentation, feature recognition and object tracking (Cootes et al. 1995, Lepetit & Fua 2005). In such applications the topology of the model is more important than its exact geometry. Approximate models are also useful for demonstrating basic concepts, e.g., in education.

Creating models and animations with sketches is particularly attractive since it encourages creativity (Gross & Do 1996) and enables user to concentrate on the overall problems rather than details (Wong 1992). The past decade has seen a tremendous increase in the design and use of sketch-based interfaces.

In this paper we present LifeSketch, a prototype of a sketch-based modelling and animation system. The modelling process is based on Igarashi et al.’s famous “Teddy” system. The main contribution of our work is an animation system which automatically detects movable parts and enables skeletal animation based on curved bones. A subsequent user study demonstrates that the system is sufficient for creating rough 3D models and the automatic joint detection is intuitive.

Section 2 reviews previous work on sketch-based modelling and animation systems. Section 3 presents the design of our system including the modelling, animation and rendering steps. Section 4 discusses the user study we use to evaluate our system and section 5 summarizes result of the user study and our own evaluation. We conclude the paper in section 6 and discuss important future work.

2 Literature Review

2.1 Sketched-Based Modelling Systems

Our analysis of successful sketch-based modelling systems for 3D objects showed that virtually all applications use the following three steps:

1. Features characterising the 3D object to be modelled are sketched in 2D using as few strokes as possible

2. The sketched 2D strokes are mapped to 3D shapes according to application specific constraints, which reflect assumptions about the shape of the 3D object to be modelled

3. Ambiguities are resolved and more detailed features added by using modifier strokes. Frequently these strokes are directly applied to the 3D shape resulting from the previous step

These steps do not include 2D sketch processing and recognition, which is a non-trivial task. Igarashi et al. (1997) evaluate each stroke input for potential
geometric relations such as connections, alignments, horizontal and vertical strokes, and symmetry. Interactive beautification is performed after identifying the most suitable geometry relation. Sezgin et al. (2001) eliminate noise from free-hand drawings by combining average based filtering and scale space filtering. The method uses curvature information and pen speed data in order to differentiate between shape features of a curve and unintended wriggles.

Different approaches have been suggested to classify sketch-based modelling algorithms. Eggli et al. (1997) categorize sketch-based modeling systems by the different types of drawing the user wants to create: technical, symbol and free-form. McCord et al. additionally differentiate free-form shapes into blobby and domain specific shapes (McCord et al. 2008). Olsen et al. (2009) present an excellent survey of sketch-based systems and classify them according to the type of modeling operation considered, i.e., creating full models from sketches, augment existing models with sketches, and deform existing models using sketch input.

We suggest to classify sketch-based modelling techniques according to the sketched primitives used to characterise the desired 3D model: silhouettes, contours, cross-sections and skeletons. In addition, 3D objects can be created by deforming more basic shapes using sketch-input.

This classification takes into account that the most natural description of objects depends not on their type (e.g., man-made vs. natural), but on their structure and the required level of abstraction. This in turn can vary for different users and different applications. In many instances several of these concepts must be combined. For example, the stem and branches of a flower can be represented by a skeleton and its leaves by silhouettes and shape modifying strokes (Ijiri et al. 2005). Note that the type of captured primitives together with assumptions made about the shape of the desired 3D object determines the algorithms used to reconstruct the 3D surface from the sketch input.

### 2.1.1 Silhouette-Based Methods

The arguably most popular class of sketch-based 3D modelling techniques uses sketch input to represent the silhouette (outline) of a 3D object. The outline, usually referred to as contour, is expanded to a 3D object by making the assumption that the object is “blobby”, i.e., the cross section of each component of the sketched contour is circular.

The arguably best known system in this class is Igarashi et al.’s Teddy application (Igarashi et al. 1999). A 3D object is computed by sampling the contour (outline), triangulating the sample points, computing a skeleton from the mid-points of all internal edges of the triangles, and then fitting circular cross-sections around the skeleton. Additional functionalities for cutting and combining objects allow the creation of complex, inflated (blobby) shapes. Various modifications have been suggested, e.g., for smoothing the resulting 3D surface (Igarashi & Hughes 2003), smoothing the underlying skeleton (Levet & Granier 2007), or for modifying the shape by sketching contours or by making the assumption that the object is “blobby”, i.e., the cross section of each component of the sketched contour is circular.

Karpenko et al. use implicit surfaces to “inflate” contours to 3D bodies. As a result different sketched components can be easily blended together (Karpenko et al. 2002). Similar ideas are employed in ShapeShop (Schmidt et al. 2006) and MIBlob which uses implicit surfaces to inflate contours traced in medical images (de Araújo et al. 2004). Other authors have shown that complex 3D objects can be edited using stylus strokes that retrace an object’s silhouette (Cheutet et al. 2005, Hua & Qin 2003). The modification of a models silhouette subsequently rescales it so that it maps itself to the new silhouette.

Two interesting application of silhouette-based algorithms are garment and tree modelling. For tree modeling the user sketches the outline of the crown of the tree and the algorithm computes a fitting branching structure based on existing templates and a probabilistic distribution (Chen et al. 2008). Garment shapes can be modelled by sketching their outline and the algorithm automatically fits them to the body shape (Turquin et al. 2007).

### 2.1.2 Contour-Based Methods

Contour-based methods are related to silhouette-based techniques. We use the popular definition that silhouettes represent the outline of shape (e.g., the shape of its shadow), but contours include all visible lines and divisions of a shape, e.g., discontinuities in the surface gradient. Karpenko & Hughes (2006) extend the closed shape outline of “Teddy” with cusps, T-junctions, and overlapping sketch lines in order to represent local details. Nealen et al. (2007) allow the user to draw arbitrary silhouette lines and use free form deformations to adapt the underlying 3D shape. Gain et al. (2001) enable users to model 3D terrains by drawing the silhouette, spine and bounding curves of both extruding (hills and mountains) and embedding landforms (river courses and canyons).

A popular application of contour-based methods is the sketching of technical drawings and 3D CAD models. Computer designed items are often characterized by a blocky shape, flat or arced surfaces, sharp or evenly rounded edges and corners, many parallel and orthogonal edges and faces, and symmetrical features. These features can be captured using silhouettes which can then be interpret using application specific constraints, e.g., that surfaces of CAD objects frequently form 90 degree angles. Examples include “SKETCH” (Zeleznik et al. 1996), where complex objects are constructed using a combination of sketched 3D primitives. “Quick-sketch” allowed users to sketch 3D solid objects and B-spline surfaces (Eggli et al. 1997). A subsequent system, “3D Sketch”, creates an edge graph from a user sketch and matches it with existing topologies. After identifying planar faces and determining a view point the edge graph can be projected into a 3D model (Mitani et al. 2002). Recently active contour models were proposed to allow both curve creation and modification from totally arbitrary view points (Kara & Shimada 2007). The depth coordinates are computed by minimizing the spatial deviation from the original target curve.

### 2.1.3 Skeleton-Based Methods

The third group of sketch-based modelling techniques are skeleton-based. These methods can be considered the dual to the silhouette-based techniques which frequently use the sketched outline to compute a skeleton (as done in “Teddy”).

Skeleton-based techniques are most popular for modelling complex fibrous and branching structures. Ijiri et al. (2005) use sketch input to model the stem and branches of flowers. The 3D shape of a stem is computed by solving a differential equation such that the curvature and appearance of the resulting 3D shape is identical to the 2D sketch. User input can be reduced by combining the technique with L-systems. The L-systems reproduction rules simulate a growth pattern which can result in arbitrary complex shapes and the sketch input controls the overall appearance and the depth of the recursion. The techniques has been used for tree modelling (Ijiri et al. 1999).
of constraints reflecting assumptions about the intentions of the user. For example, the system assumes that all objects are rigid. If a bone has different lengths in different key frames then this indicates a rotation. When the bone is longest it is assumed to be parallel to the image plane. Mao et al. (2007) use a similar approach. Users can sketch skeletons and body outlines at key frame poses. The body contour is used to estimate the fat distribution which influences the animation.

There are several papers about animating sketched objects. In all of these applications the animation itself does not use sketch-input. Igarashi et al. (2005b) use spatial key framing where key frames are not determined by points in the temporal domain, but by key poses of the 3D object. A different approach is used for the “As-Rigid-As-Possible Shape Manipulation” (Igarashi et al. 2005a). The user can animate a shape by selecting arbitrary points within it and moving them. This is achieved by triangulating the 2D shape and computing a configuration containing the moved points (triangles) such that distortions of all other triangles are minimised. The system is very intuitive and easy to use when employing a multi-touch interface. However, it does not extend to 3D shapes and the animations would be difficult to control using sketch-input (mouse, tablet and pen).

3 System Design

We want to have a system which is easy to use and intuitive, but at the same time flexible and extendable so that it can be integrated into more complex modelling environments. Our analysis of the literature shows that the Teddy algorithm is one of the most popular modelling tools, that many improvements exist, and that it has been well tested with even a commercial game based on it. For animation purposes we would like to use skeletal animation since it is widely used, supported by many graphics APIs and graphics engines, and because of the possibility to use existing motion capture and animation data. Baran and Popović demonstrated that it is possible to animate a wide variety of shapes with the same skeleton (Baran & Popović 2007).

Our system can be divided into five modules which are explained subsequently:

1. Model and spine generation
2. Skeleton generation - bones and joints
3. Skeletal animation and skimming setup
4. Rendering
5. Animation

3.1 Model and Spine Generation

The sketch-based model is created using the Teddy algorithm (Igarashi et al. 1999). The user sketches a 2D outline of the shape which is sampled. The sample point are triangulated using a Constrained Delauney Triangulation and classified according to the number of internal edges. Triangles with one internal edge are termed terminal triangles, triangles with two internal edges split triangles, and triangles with three internal triangles (i.e., no edge on the contour) are called junction triangles.

A skeleton, the so-called chordal axis, is defined by connecting all the mid points of internal edges and centroids of junction triangles. Small side branches of the skeleton are eliminated by using a pruning operation (Igarashi et al. 1999). Starting from the terminal triangle, triangles are merged until the merged
Figure 1: Left image: A sleeve triangle indicated by light grey lines. Second image from the left: The triangle fan after performing a pruning operation. The green dot represents an elevated spine node (ESNs), the yellow dots are vertices on the sketched contour and the red dots are vertices on the quarter circles connecting contour vertices to ESNs. The two images on the right show how these vertices are triangulated by connecting the mid points of the quarter circles and by subdividing the resulting quadrilaterals.

Figure 2: Left image: A junction triangle with a pruned edge resulting in a triangle fan (grey lines). The green dots represent elevated spine nodes (ESNs), the yellow dots are vertices on the sketched contour and the red dots are vertices on the quarter ovals connecting contour vertices to ESNs. Second image from the left: After elevating the spine nodes the 3D vertices corresponding to the triangle fan are triangulated using the same algorithm as for a pruned sleeve triangle. Two images on the right: The remaining part of the junction triangle can be divided into four spherical triangles by connecting the ESNs of the pruned edge to the other ESNs and the contour point opposite to it. The four spherical triangles can be triangulated by connecting the vertices along the quarter circle in a zig-zag pattern similar as for a normal junction triangle.

A 3D shape is formed by elevating all nodes of the chordal axis orthogonally to the sketch plane. The height is equal to the distance of a node to the next sample point on the sketched contour. Note that the nodes are either the mid-points of internal edges or the centroids of junction triangles, i.e., the distance is easily computed. The sample points on the contour and the elevated nodes are then connected by quarter circles which are sampled and triangulated to create a surface mesh.

Surface construction for pruned sections of the chordal axis requires special considerations which are not explained in the Teddy paper and subsequent papers. Two examples are indicated in figure 1 and 2. The complete details are given in (Yang 2009).

3.2 Skeleton Generation - Bones and Joints

In order to animate the object we have to define a skeleton which is a hierarchical structure consisting of bones and joints. The chordal axis is an ideal candidate for this since it lies approximately in the centre of the sketched contour and it has a branched structure. We define a spine tree by first finding the largest junction triangle. The largest junction triangle usually represents the widest component of the final shape. For example, for a human shape that would be the body. We therefore make the centre of this triangle the root of the skeleton. We then traverse the chordal axis graph in pre-order starting with this node. The resulting spine tree has the following properties:

- All centres of junction triangles are called branch nodes and are internal nodes of the spine tree. The root of the tree is the centre of the largest junction triangle. If no branch node (i.e., no junction triangle) exist then the resulting object is rigid.
- All branch nodes, which are directly connected to a branch node detected earlier in the pre-order traversal of the tree, are children of that node. The branch nodes are connected by sections of the chordal axis.
- The leaves of the tree are spine nodes belonging to terminal triangles. The leaves are the children of the branch node to which they are directly connected by sections of the chordal axis.

The tree represents a hierarchical skeleton, but the branch nodes (centres of the junction triangles) do not represent suitable joints of the skeleton as illustrated in the images on the left of figure 3 and 4.

Marr & Nishihara (1978) noted that the concave parts of a silhouette define the subparts of an object. We observed that the edges of junction triangles isolate the subparts of the sketched shape and hence define candidate joints. In a previous paper we presented an algorithm for selecting folding axes of a 2D sketched contour by merging edges of junction triangles and determining “bendable” sections (McCord et al. 2008). The results approximate how a piece of paper of the sketched shape can be bend. Applying this algorithm to 3D shapes obtained by the Teddy algorithm does not lead to satisfactory results as demonstrated in the images on the right of figure 3 and 4. Whereas the folding axis in the middle of the torso of figure 3 is still acceptable, the one separating the right shoulder from body is unintuitive. The same problem occurs for the folding axis separating groups of two and three fingers in figure 4.

Figure 3: The chordal axis of a sketched contour of a doll (left) and folding axes constructed from it (bold lines) using a paper metaphor (right).
Figure 4: The chordal axis of a sketched contour of a doll (left) and the folding axes constructed from it (bold lines) using a paper metaphor (right).

the edge nearest to the parent spine node is a joint. The algorithm is illustrated in figure 5. The five red dots in the image on the bottom right are the final joints of the skeleton.

### 3.3 Skeletal Animation and Skinning Setup

An object is animated by moving its bones around joints. In order to get a smooth deformation of the surface mesh the mesh vertices must be associated with bone movements. We use the popular Linear Blend Skinning algorithm which is also frequently called Skeleton Subspace Deformation (SSD). The algorithm is unpublished in the literature but an excellent description is found in (Lewis et al. 2000). A linear blend skin is created by beginning with a static model of the character. We use the 3D model and hierarchical skeleton (spinetree) explained in the previous subsections. Note that the bones of the skeleton are the sections of the spine tree, which connect two joints or a joint and a leaf. Hence the bones are usually curved. We now define for each bone a curvilinear coordinate system by parameterising the corresponding curved section of the spine tree. At each point of the bone two orthogonal vectors are created by the normal vector of the sketch plane and cross product of the normal and tangent of the bone at that point. Using this coordinate system we can now compute the coordinates of a mesh vertex with respect to the bone. This is done by first projecting it onto the sketch plane and then finding the closest point to it on the curved bone.

If vertices are bound to only one bone then linear blend skinning results in gaps or overlaps in the surface. This is avoided by binding a vertex to several bones. We do this by determining for each vertex its distance to a joint and computing appropriate vertex weights. If a vertex has an equal distance to two bones then its weights are 0.5 for each bone. In our case distance is defined by the distance of a sample point to a junction triangle in the original 2D sketch. Since all 3D vertices and the skeleton result from this triangulation it is an appropriate and easy method to determine vertex weights.

In order to animate the mesh vertices we need to rotate bones around their parent joints (i.e., the joint connecting the bone to its parent in the hierarchical skeleton). Let $v^k_d$ be the coordinates of a vertex in a dress pose with respect to bone $k$. The position $v$ of the deformed vertex is computed by:

$$v = \sum_{i=0}^{n} w_i M_i L_i^{-1} L_v v^i_d$$  \hspace{1cm} (1)

where $w_i$ is the weight of the vertex with respect to bone $i$, $L_v$ is the matrix transforming the vertex $v$ from its surface representation to the world coordinate system, $L_i^{-1}$ transforms the vertex from the world coordinate system into the static $i$-th coordinate frame (dress pose), and $M_i$ expresses the motion of the $i$-th coordinate frame. The effect of this transformation is that we can express a rotation around an axis of the coordinate frame of the parent joint as rotation around the $x$-, $y$- or $z$-axes. Note that since we have a hierarchical skeleton the matrix $L_i$ is in fact the product of the matrices representing each bone in the coordinate system of its parent bone.

### 3.4 Rendering

Two rendering algorithms were implemented: Gouraud shading is the most popular algorithm for polygon rendering and provides a smooth shaded representation of the surface. However, the surfaces produced by the Teddy algorithm frequently suffers from shading artifacts caused by irregularities in the triangulation. This can be remedied using a mesh smoothing algorithm (Igarashi & Hughes 2003), which we plan to implement in future. For the
current model we found that using a toon shader gives visually acceptable results. We implemented the approach presented in (Villar 2007–2009). The cartoon-like appearance corresponds well with the intended usage, i.e., creating rough prototypes of 3D objects and scenes. Figure 6 illustrates the different effects achieved with these rendering algorithms.

Figure 6: Two sketch-based models rendered using Gouraud shading (left) and toon shading (right).

3.5 Animation

An efficient animation of sketched objects can be achieved in three ways: The first possibility is an automatic animation, i.e., without requiring any user inputs. This can be achieved using physically-based modelling, but usually requires some domain knowledge. For example, when we detect that an object does not have joints it can be considered rigid and animated using a physics library such as ODE (Smith 2007). This way the object could collide with other rigid objects or tumble down a (sketched) slope.

If the object does have joints it can still be animated automatically, e.g., using an evolved locomotion controller (Sims 1994, Sanders et al. 2003). In this case naturally placed joints are essential and appropriate constraints (hinge joint, ball joint, etc.) must be known. A sketched object could also be animated by rigging it with an existing animated skeleton using the algorithm presented in (Baran & Popović 2007)

Finally a sketched object can be animated efficiently using sketch input as explained in subsection 2.2. In this case the joints must be intuitively placed and the user must be able to understand how a desired shape can be achieved by rotating a section of the shape around a joint.

In order to test the user’s understanding of joint locations and rotations we implemented a very simple key-based interactive animation tool. In this instance keyboard input is preferable over sketch input since it avoids ambiguous responses due to confusion about how to select a joint or about how sketch a desired behaviour.

After the user has drawn a closed contour our system immediately creates the corresponding 3D model and indicates movable parts by thick dotted blue lines as illustrated in figure 7. The user can rotate movable parts using four keys: The first key provides a toggling through all possible rotation axes. The currently active joint is indicated by a yellow dotted line. A second key is used for toggling through the three possible rotation axes represented by the coordinate system of the parent bone at the joint. The last two keys are used for rotating the selected component forwards and backwards around the selected rotation axis.

4 User Study

We performed a user study in order to evaluate the effectiveness of our “LifeSketch” framework. The participants had to do five tasks of various difficulty. The tasks required increasingly complex interactions and had an increasing demand on 3D perception and mental modeling. Our main objectives were to determine whether users are able to model simple shapes, under which circumstances the modelling is most successful, whether the automatically detected joints are natural and intuitive, and whether users can use the joints correctly in order to deform a simple shape.

4.1 Hypotheses

Based on our experience with teaching students and based on results from cognitive science we formulated the following hypotheses which formed the bases for our experiments:

- It is easy to create a 3D model from an existing model if the view plane is the sketch plane. In this case the 3D model’s contour is identical to the sketch contour and the user only has to copy the contour.
- It is harder to create an appropriate contour sketch for a 3D model shown from a different perspective, since in this case the user has to create a mental model of the object and rotate it to find the correct sketch contour.
- It is hardest to create a sketch for a shape which is only described in words (i.e., without visual template).
- Animating a 3D model is relatively easy if the shape’s components move inside the image plane, i.e., the rotation axis is orthogonal to the image plane.
- It is more difficult to animate an object if rotations around axes in all three dimensions are required.

4.2 Tasks

In order to test the hypotheses above, we designed the following five experiments:

1. The user is shown the 3D model of a hand in figure 8 “Task 1” and is asked to copy it. Note that the view direction is orthogonal to the sketch plane, i.e., the user only needs to draw the contour of the 3D model.
2. The user is shown the 3D model of a doll in figure 8 “Task 2” and is asked to copy it. Note that the two views do not contain the contour, i.e., the user must reconstruct it mentally.
3. The user is asked to create a 3D model of a duck. No images are shown and no instructions are given.

4. The user is shown the images of the doll model in figure 8 “Task 4” and is asked to copy the poses by moving the model’s legs and arms. Note that the rotations are within the sketch (view) plane.

5. The user is shown the images of the hand model in figure 8 “Task 5” and is asked to copy the pose by moving the hand’s fingers. Note that this requires rotations around several axes (for the thumb) and that the fingers move out of the image plane.

5 Results

5.1 Efficiency

We first evaluated the efficiency and correctness of the algorithm. More than a dozen simple models were constructed and a selection of them is shown in figure 7. The most complex model, the lobster, took about 20-30 seconds to sketch. The contour has 243 sample points. The resulting 3D model has 2087 vertices, 4170 triangles, and took 0.493 seconds to compute on a machine with E7200 dual 2.53GHz CPUs with 4GB RAM and NVIDIA GeForce 9600GT graphics card with 512MB memory. The subsequent animations were all performed in real-time with no noticeable delays - the exact frame rate was not measured.

5.2 Correctness

All models we created were plausible with no major artifacts such as holes or non-manifold surfaces. Figure 7 demonstrates that most detected joints are meaningful. In particular note that the egg (stone) has no joints even though its triangulation contains junction triangles. In none of our examples were “natural” joints missing. In Figure 7 (a) it could be argued that the doll figure should also have an elbow joint, but when interviewing the participants in the subsequent user study, none of them commented on this. The algorithm produces some unexpected extra joints, e.g., the mouth and the left elbow of the lobster, the teeth of the saw, and a joint below the wrist of the hand model. The extra joint of the hand demonstrates a problem with Teddy’s pruning algorithm which we plan to fix together with the problem of having to many extra joints. The extra joints could be avoided by enforcing a stricter size criteria for branches of the spine tree. However, it is not clear what size is most appropriate. We are planning to conduct an analysis of natural and man-made “blobby” objects in order to optimize the automatic joint selection. Note that some users might expect the eyes of the lobster to be movable.

5.3 User Study Results

We conducted a user study with 11 participants (8 male, 3 female). Eight of them were Computer Science students and six of them had previous experience with modelling tools. The users were asked to perform the five tasks described in subsection 4.2. A time limit of 5 minutes/task was set. The users were observed and unusual actions were recorded. Users were provided with a half page long summary of the functionality of the tool, e.g., the keys for performing rotations as explained in subsection 3.5. However, users were not told what functionality to use for a modeling task. The time for each task was measured. The user experience was evaluated by assessing the modeling results and by asking the user’s level of agreement with the following statements:

1. The resulting 3D shape looks like the shape I wanted to model.
2. The resulting 3D shape looks like what I expected after sketching the 2D contour.
3. The rotation axes (joints) were at the positions where I wanted them (This question was asked only for task 4 and 5).
4. The tool is easy to use.
5. The tool is fun to use.

The response were recorded using a seven-level Likert scale (“strongly disagree” (-3) to “strongly agree” (3)) and are shown in table 1.

5.3.1 Task 1

Five users (two non-computer science students) felt that it is difficult to create a model by drawing a contour using a single stroke. Two of them suggested that the tool should have a function which can automatically connect multiple strokes. Two participants suggested that using a pen tablet as input device would be better. We agree that when drawing long curves it is quite cumbersome to keep the mouse.
Table 1: Users’ level of agreement (from -3 to 3) with the five statements about the five tasks in subsection 4.2. $\bar{X}$ indicates the average response and $\sigma$ the standard deviation. Below each task the average time for completion is given.

<table>
<thead>
<tr>
<th>Task</th>
<th>Shape wanted</th>
<th>Shape expected</th>
<th>Rotation wanted</th>
<th>Easy</th>
<th>Fun</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.55</td>
<td>1.73</td>
<td>n/a</td>
<td>1.73</td>
<td>2.18</td>
</tr>
<tr>
<td>$T = 29s$</td>
<td>1.04</td>
<td>1.94</td>
<td>n/a</td>
<td>1.92</td>
<td>0.97</td>
</tr>
<tr>
<td>2</td>
<td>2.00</td>
<td>2.00</td>
<td>n/a</td>
<td>1.90</td>
<td>2.18</td>
</tr>
<tr>
<td>$T = 36s$</td>
<td>0.83</td>
<td>0.84</td>
<td>n/a</td>
<td>0.70</td>
<td>0.60</td>
</tr>
<tr>
<td>3</td>
<td>0.63</td>
<td>1.45</td>
<td>n/a</td>
<td>1.73</td>
<td>2.09</td>
</tr>
<tr>
<td>$T = 63s$</td>
<td>0.80</td>
<td>1.37</td>
<td>n/a</td>
<td>1.42</td>
<td>1.14</td>
</tr>
<tr>
<td>4</td>
<td>1.91</td>
<td>2.00</td>
<td>2.27</td>
<td>1.82</td>
<td>2.55</td>
</tr>
<tr>
<td>$T = 94s$</td>
<td>0.19</td>
<td>0.45</td>
<td>0.63</td>
<td>0.81</td>
<td>0.66</td>
</tr>
<tr>
<td>5</td>
<td>1.48</td>
<td>1.44</td>
<td>1.64</td>
<td>1.85</td>
<td>1.34</td>
</tr>
</tbody>
</table>

5.3.5 Task 5

For the fifth task users had to draw a hand model and rotate its fingers in order to create the shape in figure 8 “Task 5”. Table 1 shows that this task had the lowest level of satisfaction, especially among non-computer science students, and the largest variation in user responses. The subsequent interviews showed that joint positions were perceived as intuitive but six participants thought that the rotation of the mesh around joints is not intuitive. Five people felt frustrated when they were rotating the fingers and one participant could not finish the animation within the time limit of 5 minutes.

The main problem was that users had difficulty finding and recognizing the correct rotation axis. This was partially due to their representation as unshaded lines, but also due to that fact that none of the three coordinate axes of the thumb joint corresponded to the direction in which the thumb had to be rotated. Users had to rotate around two different rotation axes which seemed to create considerable confusion, especially for inexperienced users. Another surprising observation was that many users did not rotate the model with the trackball, which would have helped with the perception of the three different rotations possible (the users were made aware of this functionality in the beginning).

The results suggest that any implementation of sketch-based animations should not use rotation axes or at the very least allow arbitrary rotations. In the subsequent interviews users suggested that a more natural way to define rotations would be to indicate a motion path and goal configuration with a curved arrow. In this case the system must use the current configuration and the context to map the arrow into a feasible 3D motion (i.e., use inverse kinematics). More user testing is necessary to verify this hypothesis.

6 Conclusion and Future Work

Sketch-based modelling and animation is an exciting technology with a wide range of applications. We have reviewed the current state of the literature and suggested a novel classification of sketch-based modelling systems which we believe is useful for developing more general modelling frameworks.

We have presented a prototype of our own framework, called LifeSketch. The system is based on the “Teddy” modelling system, but uses novel algorithms to automatically extract a skeleton for skeletal animation. This makes it possible to integrate physical-based animation systems, map motion templates or develop evolutionary algorithms in order to achieve
an automatic animation of sketched objects. We have also presented some previously unpublished details for generating a surface mesh for various configurations resulting from pruning the skeleton.

Our user study demonstrated that the resulting models and joint positions are perceived as natural and intuitive. In general users enjoyed using the tool and were able to complete simple modelling and animation tasks in a short time. The main disadvantage is that the current user interface for specifying joint rotations is clumsy and unintuitive.

We are currently working on a sketch-based interface for animating sketched objects. Arrows drawn by the user are associated with bones and the arrow shape together with the orientation and position of the corresponding bone and joint are used to estimate a rotation axis and angle. We would also like to implement some of the various extensions of the Teddy algorithm in order to increase modelling power and visual attractiveness of the resulting shapes. Finally we are interested in the automatic animation of sketched objects. So far we have a prototype of a physical animation system based on ODE (Smith 2007).

References


Hart, J. (1999), The Art of Storyboard: Storyboarding for Film, TV, and Animation, Focal Press.


Design and Impressions of a Multi-User Tabletop Interaction Device

Andrew Cunningham, Ben Close, Bruce Thomas and Peter Hutterer
Wearable Computer Lab
School of Computer and Information Science
University of South Australia
{andrew.cunningham, ben.close, bruce.thomas, peter.hutterer}@unisa.edu.au

Abstract
TableMouse is a cursor manipulation device designed specifically for multiple users interacting on large tabletop surface. TableMouse tracks position, height, orientation, button state, and unique identification. It is designed using infrared light emitting diodes and computer vision to perform device tracking and identification. This paper explores the functional design of such a device. Insights into the inherent features enabled by this functionality – out of arms reach interaction, collaborative interaction – are described. The architecture, vision analysis process, and issues to consider are described. Finally two example applications utilising the TableMouse are described.

Keywords: Input Devices, Tabletop Computing, CSCW.

1 Introduction
Horizontal display surfaces, tabletop displays, are a logical solution to a number of computing applications such as military (P. Hutterer, Close, & Thomas, 2006), photographic sharing (Apted, Kay, & Quigley, 2006), planning (Reitmayr, Eade, & Drummond, 2005), and entertainment (Patten, Recht, & Ishii, 2002). Proper interaction techniques and technologies are crucial for the effectiveness of these applications. Traditional desktop input technologies have been utilised with limited success to provide tabletop interaction mechanisms. Tabletop displays are suited toward absolute positioning devices which operate directly upon the display surface, while traditional input devices such as the mouse are indirect may require a separate surface to be used. Furthermore, many tabletop applications run in collaborative environments where multiple users work closely together and any interaction device should take this into consideration.

The TableMouse (Figure 1) is a solution developed for multi-user tabletop interaction. The TableMouse is a visually tracked device, which is orientation independent, supports multiple devices for multiple users, is uniquely identified, and is tracked in three-dimensional space.

We have previously described an experiment comparing the accuracy of the TableMouse against a traditional mouse and an eBeam pen device in a Fitts Law experiment run on a tabletop computer (Cunningham, Close, Thomas & Hutterer, 2009). The result of this experiment show the TableMouse as a suitable tabletop pointing device, especially in difficult tasks where the target is small or distant.

This paper explores the unique features enabled by the design of the TableMouse. Out of reach interaction and territoriality in multi-user environments is discussed in depth. The paper describes the client-server architecture of the TableMouse system, the computer vision algorithms used and technical issues addressed.

2 BACKGROUND
As prevalent technologies such as the traditional mouse do not address the requirements of tabletop display interaction, past and current research has looked at applying a variety of solutions. Notable of these are touchscreen input, pointing based devices and tangible interaction devices.

2.1 Touchscreen Input Devices
Touchscreen technologies are an obvious choice for interaction devices for tabletop displays. Buxton et al. (1985) notes the most critical requirement of touchscreens is that “the user is not required to point with some manually held device such as a stylus or puck.” They note a number of properties that differentiate touchscreens from other mouse-like devices: 1) the number and type of events they can transmit, 2) touchscreens may support multiple points of interaction, and 3) a touchscreen device may be divided into an assortment of independent virtual devices. A severe limitation they note is the inability to signal while pointing to a touchscreen. Albinsson and Zhai (2003) note that touchscreens have two special limitations: first, the screen may become obscured by the user’s finger, hand, or arm; and second, the finger is a very low “resolution” pointing device which may adversely affect compatibility with legacy applications. Esenther and
Ryall (2006) note that the low resolution of finger input can make standard GUI widgets unusable as their size is usually optimised for the more accurate mouse pointer.

Precise software-based selection techniques for touchscreens have been investigated starting with the take-off techniques (Potter, Weldon et al. 1988). Albinsson and Zhai investigated a number of possible high precision touchscreen interaction techniques which require the user to perform gross placement of the cursor near the point of selection through the initial touch, and then a second different form of interaction for the fine grain cursor positioning. Multi-touch displays allow for the extremely accurate selection on the display surface. Benko et al. (Benko, Wilson et al. 2006) investigated the following four different forms of dual finger selection: Dual Finger Offset, Dual Finger Stretch, Dual Finger X-Menu, and Dual Finger Slider. They found dual finger selections increased the precision and accuracy in small target selection tasks, and in particular the increasing of the target size overcame the problem of fingertip occlusion.

DiamondTouch (Dietz and Leigh 2001) is a multi-user touch input device designed to work in collaboration with a front-projection display device. The current version of the device allows for multiple touch points, and the ability to differentiate between four different users. Wu and Balakrishnan (Wu and Balakrishnan 2003) investigated the use of multiple finger and whole hand gestures with a DiamondTouch input device. These forms of interaction not only support traditional pointing, but complex command entry with concurrent direct manipulation. While DiamondTouch provides consistent unique identification, the cost of the system can be prohibitively expensive. Frustrated total internal reflection (FTIR) has been shown to provide a low-cost solution to multi-touch interaction on a tabletop display (Deutscher, Hoskinson et al. 2005 ), however it relies upon computer tracking to provide unique identification of touch locations and therefore cannot consistently uniquely identify touch locations when the user removes their hand from the table’s surface.

2.2 Tabletop Pointing Devices

There are a number of other input technologies (commercial and prototype) that have been explored for the use with tabletop displays. Guimbretière and Winograd (Guimbretière and Winograd 2000) utilised ultrasonic EFI e-Beam pen devices to support the FlowMenu system. To support the marking menu functions an extra button had to be added to the pen device. Laser pointers (Olsen and Nielsen 2001) have been utilized as cursor control devices for large displays, where mouse button presses have been supported through various means (dwell time, strobing the laser, and an external wireless mouse button on the laser).

TractorBeam (Parker, Mandryk et al.) combines the natural interaction of a pen device for graphical objects within arms reach with virtual laser pointing for further objects on tabletop displays. TractorBeam is a 4DOF device (x, y, z, azimuth and elevation) utilising a top-projected display and a Polhemus Fastrak for 6-DOF tracking. TractorBeam only supports a single user and, due of the limitations of the tracking technology, the pen device must be tethered, reducing user mobility. Furthermore, metal objects in the presence of the tracker may interfere with accuracy.

The Sensetable (Patten, Ishii et al. 2001) electromagnetically tracks the positions and orientations of up to ten wireless objects on a tabletop display surface. The Sensetable employs the Wacom IntuosTM sensing tablets that support 32-bit identification numbers for each mouse on the tablet. The authors state this form of technology is superior to vision based tracking as it is not susceptible to occlusion or changes in lighting conditions. A novel feature of the Intuous tablets is that tracked objects have state that can be modified by attaching physical dials and modifiers. Each tablet only supports two tracked objects natively; the Sensetable multiplexes the tracking by randomly switching the sensing coils in the tracked devices on and off. This multiplexing limits the number and performance of devices on one table; as the number of devices increases, the latency of tracking increases. A second limitation is only two devices may be moved at one time. The Sensetable was extended with the AudioPad (Patten, Recht et al. 2002), which allows for detection of rotation of the puck and can track up to nine pucks simultaneously to an accuracy of 4mm.

2.3 Device Identification

Fiducial markers have been used to visually track objects in the physical world (Kato and Billinghurst 1999; Rekimoto and Ayatsuka 2000). The fiducial marker allows 6DOF tracking and unique identification of the individual markers. Rekimoto and Ayatsuka (Rekimoto and Ayatsuka 2000) developed CyberCode, a 2D-barcode technology as a visual tagging system. They developed their own version of tabletop phicons (physical icons), which are tangible objects that provide a means of interaction with a computer. CyberCode’s phicons track translation and orientation and are uniquely identified but do not have the notion of mouse buttons. An I/O Bulb (Underkoffler and Ishii 1998) configuration has been used tracked physical objects coded with colour dots on a table surface. This coding scheme allowed for the position, orientation, and identification of the object to be tracked. The identification is determined by the relative position of the centre dot on the object.

2.4 PuckControl
The PuckControl (Takatsuka, Lowe et al. 2006) employs infrared (IR) Light Emitting Diodes (LED) as visual landmarks for a 2D translation and button activation vision based tracking system. The motivation for this device was to fill a gap between the current tabletop input devices, such as the traditional mouse, pen-based, and hand gestures. IR LEDs were employed as ambient IR levels are usually quite low in an indoor setting. Four IR LEDs are positioned on the bottom of the puck device arranged in a circle configuration. Devices are operated on top of a back-projected screen, which allows the IR LED positions to be captured by an infrared camera placed underneath the screen. Two LEDs allow for a cursor position to be calculated. The activation of one of the LEDs signals a button press from that respective mouse button thus providing the advantage of being a wireless/radio-free solution. A major benefit of PuckControl is its scalability, as many pucks can be easily supported on the tabletop surface simultaneously.

### 3 TableMouse

The TableMouse is a visually tracked device based on a unique arrangement of up to 9 infrared LEDs, providing position, elevation, orientation, button state and unique identification. We take a computer vision approach as it allows devices to be operated untethered, is not affected by interference and can handle a large number of simultaneously tracked devices.

We imagined two different form factors for the TableMouse; a puck-like device to act as a tangible prop, and a mouse-like device to interact with a traditional graphical user interface system. Four configurations have been prototyped (Figure 2), three of which have the puck form factor, while the last is based on a modified traditional mouse. The exact feature set available to each configuration is defined by the number of active LEDs on the device. A single LED configuration (Figure 2a) conveys only two-dimensional position. A 4 LED configuration (Figure 2b) conveys two-dimensional position, height and orientation, while a nine LED configuration (Figure 2c) has the complete feature set.

The mouse-like TableMouse (Figure 2d) is our most successful configuration from a usability and acceptance point of view. It is a 4LED configuration, consisting of the casing of a two-button traditional mouse with custom circuitry. The custom circuit is mounted upon an insulated steel plate bracket which is attached to the front of the mouse, and positions the custom board and LEDs 30mm above buttons of the mouse. Two of the LEDs are used to indicate device state, corresponding to the mouse button clicks, and the remaining two LEDs, which are always illuminated, are used in combination with the two button LEDs to track position and orientation of the table mouse. Initially, nine LEDs were used in this device; however, visual tracking became error-prone due to the size of the board and the close proximity of the LEDs. The identification LEDs were disabled to overcome this problem.

The nine-LED TableMouse requires a larger circuit board (60mm x 60mm) to be created (Figure 2c). The extra five LEDs are used to indicate a unique binary pattern for identification of devices.

### 3.1 Functionality

The TableMouse is designed to be both a graphical user interface (GUI) pointing device and a tangible interaction device supporting a wide range of tabletop configurations and software. The software layer of the TableMouse consists of 1) a server backend performing the image processing and messaging of TableMouse events to registered client applications and 2) an optional client framework.

The TableMouse supports most tabletop configurations and legacy applications currently available. At its most basic configuration, TableMouse can be used as a single operating system cursor device. This provides the TableMouse the ability to seamlessly interact with legacy applications while maintaining orientation independence. Under Linux with MPX (described below) TableMouse can support multiple pointing devices at a system level and therefore integrates multiple cursor support for legacy applications. Finally, custom applications that support multiple cursors with 4DOF information can use the TableMouse client framework to subscribe and interpret TableMouse server events into unique device information, making use of the full capabilities of the TableMouse.
3.1 Orientation independence
As an absolute positioning device, the TableMouse is orientation independent from the display, letting users operate the device from any edge of the screen.

3.1.2 4 Degrees of Freedom
Being a visually tracked device allows the TableMouse to perform 4DOF tracking \((x, y, z, \text{ and } \theta)\) in real time. Orientation \((\theta)\) is calculated in 1D around the camera’s view direction.

3.1.3 TableMouse as a pointing device
The TableMouse can operate as a normal Microsoft Windows, X Windows or MacOS X mouse device. To perform this, a client application called Squeak listens for TableMouse events and propagates them back to the underlying operating system as native system cursor and button calls. This allows the TableMouse to be used with all legacy applications that are not natively aware of the TableMouse.

Specialised applications can use the TableMouse client application framework to subscribe to the same events as Squeak to determine the orientation and height of the device controlling the system cursor. This allows a user to seamlessly work with legacy applications, where the TableMouse would operate as a traditional 2DOF mouse, and TableMouse aware applications, where the device would have the full 4DOF support. Such functionality is essential for a tabletop device as this reduces cognitive context switching that would otherwise occur when changing from legacy applications to custom applications. Having two different physical input devices for each set of applications would require the user to switch devices.

3.1.4 Unique Identification
Each TableMouse may be uniquely identified through a binary pattern of the fixed IR LEDs.

3.1.5 Multiple pointer support
The TableMouse has support for up to 16 devices operating as 16 independent X Windows system cursors. To achieve this, TableMouse integrates with MPX (Hutterer and Thomas 2007), a windowing system that natively supports Single Display Groupware features.

3.2 Features
We have identified and explored several features that emerge from the functionality of the TableMouse.

3.2.1 Device Height
Although the TableMouse physically resembles a traditional mouse, it isn’t restricted to the tabletop surface. From our experiments, however, we have found that people instinctively use it on the tabletop surface, most likely due to its familiar traditional mouse form factor.

Given this default behaviour, we experimented with using the device height for secondary functionality. We felt that secondary functionality usually reflects advancement in the user’s knowledge of the system, and this in-turn is reflected by the advanced usage of the TableMouse. As an initial test, we use device elevation to simulate the scroll wheel functionality of a traditional mouse. When the table mouse was elevated above the tabletop and the right-button held down, a scroll event would be sent to the application current application, allowing users to scroll long documents or pan through an image, for example. This scroll event, being the delta \((S_x, S_y)\) from the current position of the active application’s viewport, is defined as follows:

\[
(S_x, S_y) = (D_x \cdot D_h \cdot k), (D_y \cdot D_h \cdot k)
\]

Where \(D_x, D_y\) are the delta of the horizontal and vertical TableMouse movements, respectively, \(D_h\) is the height of the device from the table’s surface and \(k\) is some scaling constant.

This allows a user to fluidly scroll, overcoming the TableMouse’s inability to have any continuous or analogue buttons such as a scroll wheel.

3.2.2 Device Orientation
To improve usability when operated from an orientation that is not aligned with the system orientation, the TableMouse positions the system cursor in-front of and orientated with the particular device (Figure 3). This feature is also crucial to ensure the user can aim with the cursor as it renders it always visible.

3.2.3 Out of Arms Reach Interaction
While we are primarily interested in investigating an absolute positioning device, being able to interact with parts of the display which are out-of-reach, similar to the pointing feature of TractorBeam (Parker, Mandryk et al. 2005), is desirable. As a by-product of the top-down camera configuration, the TableMouse can operate as an absolute-positing device or as a scaled positioning-device, where absolute movements of the device are...
scaled on to display. When operated exclusively on the tabletop surface there is a one-to-one mapping between the cursor position and the position of the device on the display. As the device is raised above the table’s surface however, the cross section of the capturing camera’s view frustum is reduced and movements of the device are scaled relative to the device’s height from the table.

This property allows the user to manipulate cursors in a larger display area than their arm’s reach is capable of, as depicted in Figure 4. The black rectangles in the figure indicate positions of TableMouse devices. By raising the device from the tabletop surface the user is able to interact with the far end of the display thus reducing the user fatigue that can occur when they are required to physically reach across the table.

Comparatively, go-go interaction techniques (Poupyrev, 1996) can confuse a user. These techniques provide a linear mapping up to an arbitrary point, beyond which a non-linear mapping is used. Confusion occurs since predicting from which point a non-linear mapping occurs and the function used is difficult without any cue. The TableMouse relies upon the implicit projective nature of the device to perform out-of-reach interaction. This is intuitive as it replicates the interaction a shadow would have if the top mounted camera were a light source.

Through user study feedback, we have found this unique out-of-arms-reach interaction to be a beneficial attribute of the technology. One user commented positively that in certain cases the device was less strenuous than the e-Beam. This was especially obvious in cases where the user was not tall enough to comfortably reach the end of the table. Based on this feedback we have implemented a “scaling” property into the TableMouse software, allowing the out-of-arms reach to be exaggerated on a per-user basis.

Out of arms reach interaction is of interest to the territories (Stacey, Sheelagh et al. 2004) that occur naturally in collocated collaboration. Users collaborating on a tabletop have tendencies to partition the available space into personal, group and shared territories. Personal territories are established closest to the user for ergonomic reasons, under which case it would be preferential to use the TableMouse exclusively on the tabletop to stabilize the hand for better accuracy. Group territories represent shared space between all users and tend to be located in a position optimal for all those involved. In this case, the out-of-reach interaction would be preferential as the user could reach further into the group territory without having to relocate. This also serves as a social cue to indicate when interaction with a group territory is occurring. Collisions in group territories may be avoided since users can “hover” the TableMouse above the other users, an aspect which just is not possible with many touchscreen and pen-based interactions. Furthermore, since the elevation of the device is tracked, when a user positions their TableMouse to cover another TableMouse, a client application could employ an artificial “veto” function – the social cue of covering up another user’s device may be recognized and enforced by the software.

3.3 Vision Analysis

The image processing side of the server is built upon the open source OpenCV image processing library. Images are captured from an IEEE1394 camera at 30 frames-per-second (FPS) at a resolution of 1024x768, in 8bit mono. The camera has an IR bypass filter installed over the lens allowing the infrared near and medium light (750nm – 2000nm) to pass through to the camera.

We chose a binary pattern of fixed IR LEDs to represent the unique position and identification of each TableMouse. We rejected a blinking binary pattern for identification due to latency in determining the identification pattern. As Moore et al. (Moore, Want, Harrison, Gujar, & Fishkin, 1999) note, the theoretical throughput for detecting a pattern is one half the frame-rate of the digital camera observing the pattern. Given that we are trying to use common components (such as a 30 FPS camera), this option just isn’t sufficient.

3.3.1 Development Issues

Before discussing the vision analysis process, it is worth mentioning several issues that should be noted to anyone developing similar technology to the TableMouse.

**Issue 1.** LED divergence. The LEDs used have a light distribution that is restricted to a low divergence. Low divergence becomes a problem when the beam of light produced by the LED does not enter the camera field-of-view (FOV) because the LED is angled away from the FOV of the camera. This can be addressed by increasing the intensity of the LED, however this creates a flood of light when the LED directly points into the camera. These are diometric issues; increase the LED brightness too much and it will flood the camera image when it is incident with the camera, don’t increase the brightness and it will not be visible when angled away from the camera.

**Issue 2.** Rapid movement causes LEDs captured by the camera can appear smeared. In the worst case, the LEDs will appear to blend into each other.

**Issue 3.** Accuracy is a major objective of the TableMouse. In our experimental setup, where the
camera is mounted 1.5 meters above the table, we capture more than the full screen surface in the camera image. This means that some of the image resolution is beyond screen space, effectively reducing the number of useful image pixels. From where our camera is mounted, approximately 720 x 534 of the 1024 x 768 image is used by the screen surface; almost a 30% reduction.

Several solutions were employed to address these issues in the vision analysis process (Figure 5), which is described in the following sections.

### 3.3.2 Camera Image Undistortion

Initially in the vision analysis, an image is captured from the camera and processed through the Camera Coordinate Transformer, which applies a homography to the image to correct for the radial and tangential distortion of the camera as well as mapping coordinates in the image into 2D screen space.

### 3.3.3 Particle Recognizer

The corrected image is passed to the Particle Recognizer, which captures “particles” in the image. In this case, particles are LEDs; however any sort of distinguishable point may be considered a particle, hence the generalised term of “particles”. This allows us to adapt the TableMouse vision analysis to alternate technologies to IR LEDs and cameras.

The Particle Recognizer begins by thresholding the image to remove any low-level noise. Through experimentation, we settled on an adaptive thresholding, which performs a threshold on the pixel $\forall(x, y)$ against a Gaussian averaged n-by-n region around the pixel. We chose this method as this produced the best results to address Issue 1, which causes the LEDs near the periphery of the camera to appear far less lit than those in the centre.

After thresholding, the particle recognizer identifies all contours within the image. To compute the centre of mass of the detected contours, which is required to determine the exact device location, we use the contour moments. This decision was made following experimentation to increase the pixel accuracy of the TableMouse (Issue 3).

To compute the centre, we find the zeroth and first moments:

$$
\bar{x} = \frac{\sum_{i=0}^{n} I(x, y)x}{\sum_{i=0}^{n} I(x, y)}, \quad \bar{y} = \frac{\sum_{i=0}^{n} I(x, y)y}{\sum_{i=0}^{n} I(x, y)}
$$

Where $\bar{x}$ and $\bar{y}$ are the computed sub-pixel centre of the contour.

### 3.3.4 Device Recognizer

These particles are passed to the Device Recognizer. During initial development, device recognition occurred through graph matching, which attempts to match an ideal template graph to a set of input graphs. In this case, the graphs were specified as particles being vertices of the graph, and the edges being physical relationships between the vertices (e.g., vertex N should twice as far from vertex M as vertex O). This choice was essential in allowing us to rapidly prototype the physical layout of the TableMouse without requiring expensive code rewrites, as the graphs could be specified in an easily rewritable configuration file.

Eventually, an optimal device layout was determined (Figure 6) based on tracking accuracy in most common circumstances. At this stage the generalized graph matching algorithm was replaced with an optimized discreet syntactic pattern recogniser (Schalkoff 1992), that attempts to match relative distances between sets of particles with a set of known constraints. These constraints are split into required structural constraints and secondary state and identification constraints. Required structural constraints identify the particles that must be present on the device, otherwise a device’s position and orientation cannot be determined. These constraints and related margins of error provide optimal tracking under normal operation without resulting in erroneous devices being identified, as specified in Table 1. These constraints allow for some margin of error as the camera introduces noise into the data. Furthermore, if the user were to tilt the device, the angles and relative distances would vary from the ideal state, which the margin of error to compensate for.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
<th>Margin of error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A and B are the closest particles to each other</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>C is closer to A than to B</td>
<td>N/A</td>
</tr>
<tr>
<td>3</td>
<td>$</td>
<td>AC</td>
</tr>
<tr>
<td>4</td>
<td>$</td>
<td>BC</td>
</tr>
<tr>
<td>5</td>
<td>$\angle BAC \approx 72.075\pi$ rad</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Table 1: Constraints

The computer vision algorithms were chosen to be computationally inexpensive, as we wanted to rapidly prototype a variety of device designs and functionality without relying on any hardware vision processing. The final algorithm returns a list of devices $P$, where each device is defined as:

$$
P_1 = (x_i, y_i, z_i, \theta, \Xi_i, \epsilon_i)
$$

Where $x_i$ and $y_i$ are the coordinates of the device, $z_i$ is the height, $\theta$ is the orientation, $\Xi_i$ is the set of button state and $\epsilon_i$ is the unique identification. Besides the coordinates, any of these values may be null, indicating that the device does not support that particular feature.

Given this definition of the device, the following parameters are used in the vision analysis process:

![Figure 6: Device Layout](Image)
\[ V := \{v_0, v_1 \ldots v_n\}, \] where each \( v_i \) is an identified particle of the form \((x, y)\)  
\[ V' := \text{set of particles in } V \text{ that are recognised as forming a device, initially } \emptyset \]  
\[ D := (d_{ij})_{n \times n} \text{ is a } n \times n \text{ symmetric matrix, where } d_{ij} \text{ is equal to the distance between the particles } v_i \text{ and } v'_j (|v_i - v'_j|) \]  
\[ E := (e_{ij})_{n \times n} \text{ is an } n \times n \text{ matrix where for any row in } E, e_{i1 \ldots n} = \text{argsort}(d_{i1 \ldots n}) \text{ and argsort is a function that returns the indices of its sorted argument values} \]  
\[ P := \{p_0, p_1 \ldots p_n\}, \text{ set of identified devices, initially } \emptyset, m \text{ is the number of identified devices} \]  
\[ Q := \text{set of particle secondary constraints specified as tuples of the form } (\beta, dx, dc, ax, ac), \text{ where } \beta \text{ represents which particle, either state or unique identification constraint, this constraint specifies (one of } D \ldots i), \text{ dx represents the ratio } |AB|/|A|, \text{ dc represents the acceptable distance error, ax represents the angle } BA\beta \text{ and ac is the acceptable angle error} \]  

Pre-computing \( D \) and \( E \) allows the classification to be run as a linear search through the set \( V \), as shown here:

**Algorithm 1:** doclassify

```plaintext```
for \( i \) from 0 to \(|V|\)  
  comment: only process particles that do not already form another device  
  if \( v \notin V' \):  
    result = classify\((i, D, V)\)  
    if result ≠ Nothing  
      \( V' \leftarrow V' + \text{ result} \)
```

The classify function performs constraint matching against the particles in \( V \) in an attempt to determine if and \( v_i \) corresponds to LED A of a unique device. The function can be divided into two parts. The first part attempts to match the required structural constraints as a series of assertions. If one assertion fails, the function terminates returning nothing:

**Algorithm 2:** classify\((i, D, V)\)

```plaintext```
output: Nothing if \( v_i \) is not LED A of a device, else  
comment: copy the indices of the 3 closest particles to \( V_i \) into \( a, b \) and \( c \), and copy the coordinates of the 3 closest particles in \( A, B \) and \( C \)  
\( a, b, c \leftarrow D_{i0}, D_{i1}, D_{i2} \)  
\( A, B, C \leftarrow V_{i0}, V_{i1}, V_{i2} \)  
comment: C must be closer to A than to B  
if \(|AC| > |BC|\)  
  then return (Nothing)  
comment: |AC| must be within 1.5 * |AB|  
if \(|AC| \approx |AB| * 1.5 \) or \(|BC| \approx |AB| * 2.5:\)  
  then return (Nothing)  
comment: The angle \( \angle BAC \) must be within -2.8°
```

if \( \angle BAC \approx -2.8° \pm 0.5° \)  
then return (Nothing)  
comment: We have recognised a device
```

At this stage, classify has recognized a device via the required structural constraints, with its centre located at \( A \) and an orientation determined by \( AB \).

The second part of the classify function uses secondary state and identification constraints to recognise the identification and state particles of the device. Since \( A, B \) and \( C \) are known, the identification and state LEDs of the device can be recognised by a linear search through the next 6 adjacent LEDs to \( A \). So for any \( \eta \in D_{i1 \ldots 7} \), this search will attempt to match the characteristics specified in the set of tuples \( Q \) against the ratio \( |AB|/|A\eta| \) and \( \angle B\eta \). This determines if any of the LEDs match the state particle \( D \) or identification particles \( |E F G H I| \). Any recognised identification LEDs are placed into a bit set which corresponds to the unique identity of the device. The function finally returns identified device, which is appended to \( P \).

### 3.3.5 Device Tracking

The identified devices, referred to as “device samples”, are passed to the **Device Tracker**. The purpose of the Device Tracker is to assign a consistent unique identification to successive samples (known as assigning a device track to a device sample). This unique identification may either be persistent, which is based upon the identification LEDs on the device, or transient, for devices without identification LEDs. In the persistent case, the tracker is used to enforce or correct the syntactic classification. In the transient case, the tracker assigns an arbitrary unique identification that exists only as long as the device is present on the interaction surface.

The Device Tracker is currently implemented with the transportation algorithm, which was chosen for it is low processor usage while still providing stable tracking. The algorithm attempts to minimize the cost of assigning a previously identified device (referred to as a track) and devices recognized in the current frame (referred to as samples). The cost is determined by a cost function, which applies a set of heuristics to a track and a sample, and may be described as:

\[
\text{COST} = |S_{\text{pos}} - T_{\text{pos}} + T_v| + \frac{\Gamma(T)}{360}
\]

Where \( S_{\text{pos}} \) and \( T_{\text{pos}} \) are the position of the sample and the track respectively, \( T_v \) is the velocity of the track as

![Server Architecture](image)

**Figure 7:** Server Architecture
computed from previous matching tracks. \( \Gamma \) is a function that returns the number of milliseconds that have elapsed since the last observation of its argument. This cost function is minimized across all tracks and samples to perform tracking.

4 Server-Client Architecture

TableMouse is designed as TCP/IP client-server framework. The server performs the image processing and vision analysis and distributes the identified devices as events to subscribed clients. While this approach does introduce latency into the system, we chose a client-server architecture as this is the most flexible in terms of writing client applications as most languages have a TCP/IP stack – currently there are TableMouse client stacks for C++, Java, and Python.

Basic TableMouse system interaction can occur by running the client program, Squeak. Squeak manipulates the standard operating system mouse pointer using the position and state of the first device returned by the TableMouse server.

Specialised applications can use the TableMouse client application framework to subscribe to the same events as Squeak to determine the orientation and height of the device controlling the system cursor. This architecture is illustrated in Figure 7. This allows a user to seamlessly work with legacy applications, where the TableMouse would operate as a traditional 2DOF mouse, and TableMouse aware applications, where the device would have the full 4DOF support. Such functionality is essential for a tabletop device as reduces the effect of the cognitive context switch that would otherwise occur when switching from legacy applications to custom applications while using the TableMouse.

4.1 Applications

We developed two notable applications that take advantage of the TableMouse; CheeseDraw and MalaMinya.

CheeseDraw is a vector-based drawing application that demonstrates the rich interactions possible enabled by the TableMouse. CheeseDraw leverages the 4DOF nature of the TableMouse to create, manipulate and delete graphical objects on a canvas.

CheeseDraw appears as a standard GUI application with a minimal interface (Figure 8). The left side of the window contains controls for the user to select the type of vector shape they will create, including rectangles, ellipses and photographs. The process of creating vector shapes resembles most drawing applications, requiring the user to click and drag regions on the canvas with the TableMouse to specify position and size of the new shape.

Once created, the user may manipulate the shapes using the full capabilities of the TableMouse. Clicking and dragging a shape will translate the position of the shape around the canvas. One-degree orientation of the TableMouse is directly reflected in the dragged shape so that the orientation of the TableMouse affects the orientation of the shape in a one-to-one mapping (Figure 9). Raising the TableMouse from the table surface while dragging a shape causes the shape to scale respective to the elevation of the device from the table. This unique interaction allows a user to alter three attributes of a shape (position, orientation and scale) of a shape in one continuous motion of the TableMouse.

Our multi-user drawing tool MalaMinya (not pictured) utilises SDG functionality to provide a drawing canvas that can be operated by use to eight users simultaneously. Figure 5 shows MalaMinya begin operated by three different users. The users have unique icons assigned to their cursors and can draw lines, delete with an eraser or wipe the whole canvas. Initiating a tool only activates the tool for the activating device. Toolbars are aligned around the table for close proximity to the user’s physical position, and each user has their own toolbar. The user’s toolbar is identified by the user’s unique cursor icon. This is indicated on the left side of the user’s tool bar. The toolbars are limited to a single user each using MPX’s floor-control mechanism, whereas the various colour buttons are accessible for anyone. The available colours are spread around the drawing canvas and each user can pick a colour at any time. Doing so only changes colours for the user’s device. MPX allows MalaMinya to be used simultaneously with legacy applications such as a web browser or office applications.

5 Discussion

We have received extensive subjective feedback from a previous user study (Cunningham et al. 2009). Consistently, users have commented that the TableMouse feels “smooth” and “fluid”, despite the fact that the device is limited to the camera’s frame-rate of 30fps. We believe this perceived smoothness is due to the TableMouse being unrestricted to any surface, unlike a traditional mouse where lifting it from its surface will stop the tracking, causing a discontinuity between the
cursor and the physical device. This theory is enforced by user comments repeatedly mentioning that they appreciated being able to lift the device from the table while still controlling the cursor.

Users have found it disconcerting that under rapid movement the cursor would trail behind the mouse. This is a technical limitation between the time of moving the mouse and processing the captured frames from the camera to move the cursor. However, we feel that this issue is acceptable given the benefits of the approach we have taken.

The TableMouse requires more physical prototyping. The most successful design, based on the traditional mouse, carries with it an incorrect intuition of how the device should be used; it can still be utilised above the tabletop. This would be addressed by a more unique form factor and instruction.

6 Conclusion and Future work

We have detailed the design and implementation of a multi-user tabletop interaction device. We have discussed the features that are enabled by this implementation. The concepts of out-of-reach interaction and territoriality, upon further exploration, may prove to be boons to multi-user collaboration. We also discussed the issues and solutions we came upon when implementing the TableMouse and its computer vision algorithms, which would prove useful to anyone researching a similar technology.

There are a number of future research directions we would like to pursue. Firstly, we plan to write more client applications to leverage the concepts of territoriality and out of reach interaction. To improve the feel of the device, we will look at incorporating some physical features of a pen-like device to encourage it to be lifted above the tabletop surface and we will experiment with multiple camera tracking to provide full 6DOF tracking of the TableMouse, as we envision tilt gestures to be a useful interaction technique.

7 References


Graph Drawing Aesthetics in User-Sketched Graph Layouts

Helen C. Purchase  
Department of Computing Science  
University of Glasgow  
Glasgow, G12 8QQ, UK  
hcp@dcs.gla.ac.uk

Beryl Plimmer, Rosemary Baker,  
Christopher Pilcher  
Department of Computer Science  
University of Auckland  
Auckland 1142, New Zealand, UK  
beryl@cs.auckland.ac.nz  
{rbak026,cpi1016}@aucklanduni.ac.nz

Abstract
Empirical work on appropriate layout aesthetics for graph drawing algorithms has concentrated on the interpretation of existing graph drawings. A more recent experiment has considered layout aesthetics from the point of view of users moving nodes in an existing graph drawing so as to create a desirable layout. The project reported here extends this research further, by asking participants to use sketching software to draw graphs based on adjacency lists, and to then lay them out – removing any bias caused by an initial configuration. We find, in common with many other studies, that removing edge crossings is the most significant aesthetic, but also discover that aligning nodes and edges to an underlying grid is important, especially to male participants who have Computer Science experience. We observe that the aesthetics favoured by participants during creation of a graph drawing are often not evident in the final product.

Keywords: Graph drawing creation, graph aesthetics, sketching.

1 Introduction
Typically the aesthetic criteria for graph layout algorithms have been based on the intuition of the algorithm designers. Some empirical work has been done in an attempt to verify the usefulness of these criteria in reading and understanding graphs (as measured by graph tasks, often shortest-path tasks) (Purchase, 1997). More recently, work has been done on the process of graph comprehension, using eye-tracking data (Huang, 2007). These empirical studies have all related to the reading and understanding of graphs, rather than their creation. A recent study by van Ham and Rogowitz (2008) considered the creation of graph layouts, looking at how people prefer to layout graphs when they have the opportunity to move nodes and edges around. This is a very different empirical task to that of reading and interpreting a graph, and Van Ham and Rogowitz investigated, in particular, the depiction of clusters, the presence of edge crossings, edge length distribution and orientation.

The work presented here builds on the work of van Ham and Rogowitz (2008), but differs in two important ways. First, the participants drew the graph from scratch, rather than simply moving nodes in a pre-drawn graph – this removed any layout bias present in the original drawing. Second, they used a sketching tool with a stylus – this allowed for the physical action of creating the graph to be done as easily as if it were on paper, reducing any cognitive distance between the participant’s desired drawing and what is presented. Using the interface of a formal graph drawing tool is a less natural way of drawing a graph than the free-form hand movements made possible by a sketching tool.

This paper describes previous graph drawing empirical work, discusses the work of van Ham and Rogowitz and its conclusions, outlines the graph sketching experiment, and presents its results. Three aspects of the human graph drawing process are discussed: the product, the process and the preferences.

2 Related Work
The many graph layout algorithms that have been devised over several decades (Battista et. al., 1998) have typically been designed in accordance with the intuitions of the algorithm designers. Over the years, a set of assumed ‘graph drawing aesthetics’ has emerged, defining the criteria by which the ‘goodness’ of the graph drawing produced by a layout algorithm can be assessed (Coleman and Stott Parker, 1996, Purchase et. al., 1995). Such aesthetics include, for example, a minimum number of edge crossings, as few edge bends as possible, a display of symmetric sub-structures, and large angles between edges incident at a node. Graph layout algorithms therefore tend to be valued for the extent to which their output graph drawings conform to these aesthetic criteria.

Graph drawings should also be assessed according to the extent to which they assist human comprehension of the relational information represented in the graph. Some empirical work has been done to this end, investigating whether the aesthetic criteria used by algorithm designers do indeed assist with comprehension. Findings include the overwhelming evidence for the reduction of edge crossings (Huang et. al., 2006, Ware et. al., 2002, Purchase, 1997), some evidence for the reduction of bends and depiction of symmetry (Purchase, 1997), placement of important nodes at the top of the graph (Huang et. al., 2007) and large angles between incident edges (Huang 2007). All these studies have been conducted by asking participants to answer graph-based
questions on a variety of presented graph drawings, each carefully controlled for the aesthetic criteria.

A recent publication by van Ham and Rogowitz (2008) has taken a different empirical approach to determining the best graph layout for human use. They asked participants to manually adjust the layout of existing graph drawings: “rearrange the nodes in the network in a way that you think best reflects their interconnections.” They used four graphs of 16 nodes, each with two clusters: these clusters were separated by one, two, three and four edges respectively. These were both presented in a circular and a spring layout (Gansner et. al., 2005), giving a total of eight starting diagrams, which were presented in random order. Users of the Many Eyes visualisation service (Viegas et. al., 2007) were invited to take part in the experiment, by adapting the layout of the drawings. They collected 73 unique drawings, which they visually analysed according to the number of edge crossings and evidence of clustering, as well as other computational measures such as edge length distribution and cluster distance. They found that most participants separated the two node clusters, the human drawings contained 60% fewer edge crossings than the automatically produced drawings, and that humans did not value uniform edge length as much as the spring algorithm did.

Van Ham and Rogowitz (2008) acknowledge the limitations of their work. In particular, using a web-based experiment means that they have no information on their participants, apart from the fact that they had ‘some sophistication in data visualisation.’ In addition, the fact that the graphs were presented with an initial layout may have biased the resultant drawings, and the names used as node labels may have had an effect (for example, when participants attempt to avoid label overlaps).

The experiment reported here improves on van Ham and Rogowitz’ (2008) methodology in several important ways:

• the experiment was conducted face-to-face, so demographic information about the participants is available and we know that all participants did all drawings;
• the participants drew the graphs from scratch, so they were not biased by any initial layout;
• a sketching tool was used, so the physical drawing process was unhindered by a clumsy editing process and participants could draw curved or bent lines if they wished;
• we collected video data, so we were able to analyse both the process and product of creation;
• we discussed layout preferences with the participants in a post-experiment interview, so we were able to find out more about their thoughts on the process;
• our node labels were simple letters, enclosed within the node boundary.

Our more comprehensive and face-to-face methodology and our choice of equipment resulted in a smaller version of the experiment with 17 participants. Including the task of graph creation as well as layout meant that we used four graphs (two practise graphs and two experimental graphs), so as to make the duration of the experiment acceptable to participants. The screen size of the tablet PC sketching tool limited the graphs to 10 nodes each, with the clusters joined by one and two edges respectively.

Our experiment has produced extensive and rich data, in terms of product, process and preferences. The results can inform the design of automatic graph drawing algorithms by highlighting those features that users consider important when they are unconstrained in their own drawing of graphs and are not subject to any layout bias.

3 Graph Sketching Experiment

3.1 Equipment

A graph-drawing sketch tool, SketchNode (Reid et. al., 2007, Figure 1) was used on a tablet PC. This tool allows nodes and edges to be drawn with a stylus on the tablet screen, laid flat, thus allowing the same hand-movements as pen-and-paper, giving a more natural interaction than using an editing tool. Unlike pen-and-paper, however, the SketchNode interface allows nodes (or groups of nodes) to be selected and relocated (with corresponding movement of attached edges), and nodes and edges to be erased. It thus has the advantages of pen-and-paper, as well as the advantages of a graph drawing editing tool.

![Figure 1: The SketchNode system](image)

3.2 Task

Participants were given an adjacency list of edges (Figure 2) and asked to draw the graph in SketchNode using the stylus, with the instruction to *Please draw this graph as best as you can so to make it “easy to understand”.* They were deliberately not given any further instruction as to what “easy to understand” means. In particular, they were not primed with any information about common graph layout aesthetics, for example, minimising edge crossings, use of straight lines etc. They were given as long as they liked to draw and adjust the layout of the graphs.

<table>
<thead>
<tr>
<th>Graph A</th>
<th>(A,D) (A,C) (B,D) (C,D) (B,C) (B,E) (C,E) (E,J) (F,G) (J,F) (F,J) (G,I) (J,H) (I,H)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph B</td>
<td>(J,F) (J,I) (G,I) (H,I) (G,H) (F,H) (G,J) (F,A) (F,G) (G,E) (A,E) (D,E) (D,C) (D,B) (C,B) (A,B) (A,C) (B,E)</td>
</tr>
</tbody>
</table>

Figure 2: The adjacency lists for the two experimental graphs
3.3 Graphs
We designed two experimental graphs: graph A had 10 nodes and 14 edges; graph B had 10 nodes and 18 edges. We were unable to use graphs as big as those used by Van Ham and Rogowitz (2008), as we were required to keep the duration of the experiment to a reasonable time, and the screen size of the tablet PC was limited. However, these graphs were still designed with the aims of Van Ham and Rogowitz (2008) in mind, as they both had identifiable clusters: graph A had two clusters separated by one cut edge; graph B had two clusters separated by two cut edges: Figure 3 shows these two graphs as drawn by participant 4, clearly showing the two clusters.

![Figure 3: 4A and 4B: The drawings of graphs A and B by participant 4, using the SketchNode system](image)

3.4 Experimental Process
After reading the information sheet and signing the consent form, participants answered a pre-experiment questionnaire which asked for some demographic information, including information about the extent of their experience with maths, theoretical computer science, graphs, and pen-based technology.

The participants were then given a demonstration of the SketchNode system, including all the interface features: node and edge creation using the stylus, selecting and moving nodes and edges, selecting and moving sub-graphs, labelling nodes, erasing, undoing and redoing actions, zooming and scrolling. They were also shown how to represent a simple four-element adjacency list as a four-node and four edge graph drawing. Participants were given ample chance to ask questions about the SketchNode system and the process of representing an adjacency list as a graph drawing at this stage.

Besides the two experimental graphs, A and B, two practise graphs were defined, P1 (n=5,e=6) and P2 (n=8,e=8), and these were presented first. Participants were not aware that these were practise graphs – their use ensured that the participants were comfortable with the task and with the system before they drew the two experimental graphs that we were interested in. Exactly the same instructions were given to the participants for the practise graphs as for the two experimental graphs which followed: Please draw this graph as best as you can so to make it ‘easy to understand’.

The two experimental graphs A and B, were then presented to the participants, with the edges listed in a different random order for each participant. Each experiment was conducted individually, with only the experimenter and participant present. So as to control for any possible ordering effects, seven participants were given A before B, while other ten were given B before A.¹

3.5 Participants
The data from 17 participants (numbered 3-19) was collected for analysis. Participants were friends, family and classmates of the student experimenters, and were students and non-students, of both genders. Only some of the student participants were studying Computer Science (Table 1).

<table>
<thead>
<tr>
<th>Gender</th>
<th>7F, 10M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current occupation</td>
<td>10 students, 7 non-students. None of the non-students were younger than 21.</td>
</tr>
<tr>
<td>Age</td>
<td>max 49, min 20, mode 22, median 24, mean 27.65</td>
</tr>
<tr>
<td>Education</td>
<td>13 have some university-level education, 4 have not</td>
</tr>
<tr>
<td>Experience of graphs</td>
<td>8 yes, 9 no</td>
</tr>
<tr>
<td>Experience of university maths or computer science</td>
<td>8 yes, 9 no (matching the ‘experience of graphs’ data)</td>
</tr>
<tr>
<td>Pen-based technology experience</td>
<td>4 none, 9 minimal, 3 moderate, 1 extensive.</td>
</tr>
</tbody>
</table>

Table 1: Demographic information about the participants

3.6 Data Collection
All interactions with the tablet were recorded using Morae software (2009), producing screen casts of all the participants’ interactions with the SketchNode tool, as well as a corresponding audio track. The time taken for the drawing of graphs A and B was recorded.

At the end of the experiment, the participants were asked “Why did you arrange the graphs in the way you did?” in a recorded interview.

4 Results
4.1 The product: what do the graph drawings look like?
Of the 34 drawings produced (Figure 4), three were incorrect. 15A had an additional edge (G,H), 19A represented the node H twice, and 16B was missing the (D,C) edge: this had been drawn by the participant at the start, but had been lost in the later editing process. As the focus of the experiment was on how participants represented graphs (and not on whether they drew the

¹ These numbers (7 and 10) are not half of the total number of the participants, as the data from two participants (participants number 1 and 2) who were given A before B has been removed from the analysis - in both these cases the experiment was affected by unexpected interruption or data collection errors.
graphs correctly or not), these graphs were not removed from the analysis.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td><img src="image" alt="Graph 3A" /></td>
<td><img src="image" alt="Graph 3B" /></td>
</tr>
<tr>
<td>4</td>
<td><img src="image" alt="Graph 4A" /></td>
<td><img src="image" alt="Graph 4B" /></td>
</tr>
<tr>
<td>5</td>
<td><img src="image" alt="Graph 5A" /></td>
<td><img src="image" alt="Graph 5B" /></td>
</tr>
<tr>
<td>6</td>
<td><img src="image" alt="Graph 6A" /></td>
<td><img src="image" alt="Graph 6B" /></td>
</tr>
<tr>
<td>7</td>
<td><img src="image" alt="Graph 7A" /></td>
<td><img src="image" alt="Graph 7B" /></td>
</tr>
<tr>
<td>8</td>
<td><img src="image" alt="Graph 8A" /></td>
<td><img src="image" alt="Graph 8B" /></td>
</tr>
<tr>
<td>9</td>
<td><img src="image" alt="Graph 9A" /></td>
<td><img src="image" alt="Graph 9B" /></td>
</tr>
<tr>
<td>10</td>
<td><img src="image" alt="Graph 10A" /></td>
<td><img src="image" alt="Graph 10B" /></td>
</tr>
<tr>
<td>11</td>
<td><img src="image" alt="Graph 11A" /></td>
<td><img src="image" alt="Graph 11B" /></td>
</tr>
<tr>
<td>12</td>
<td><img src="image" alt="Graph 12A" /></td>
<td><img src="image" alt="Graph 12B" /></td>
</tr>
<tr>
<td>13</td>
<td><img src="image" alt="Graph 13A" /></td>
<td><img src="image" alt="Graph 13B" /></td>
</tr>
<tr>
<td>14</td>
<td><img src="image" alt="Graph 14A" /></td>
<td><img src="image" alt="Graph 14B" /></td>
</tr>
<tr>
<td>15</td>
<td><img src="image" alt="Graph 15A" /></td>
<td><img src="image" alt="Graph 15B" /></td>
</tr>
<tr>
<td>16</td>
<td><img src="image" alt="Graph 16A" /></td>
<td><img src="image" alt="Graph 16B" /></td>
</tr>
<tr>
<td>17</td>
<td><img src="image" alt="Graph 17A" /></td>
<td><img src="image" alt="Graph 17B" /></td>
</tr>
<tr>
<td>18</td>
<td><img src="image" alt="Graph 18A" /></td>
<td><img src="image" alt="Graph 18B" /></td>
</tr>
<tr>
<td>19</td>
<td><img src="image" alt="Graph 19A" /></td>
<td><img src="image" alt="Graph 19B" /></td>
</tr>
</tbody>
</table>

**Figure 4: The graph drawings**

The graph drawings were visually analysed for the following layout features (Table 2).\(^2\)

- **Number of edge crossings.** A crossing is defined as any point outside of the node boundaries where one or more edges cross.
- **Representation of clusters.** This feature is true if a straight line can be drawn through the graph drawing so that the two pre-defined clusters in the graph can be visually separated.
- **Number of hulls.** A hull is defined as a cluster that is bounded by edges. This feature is either 1 or 2, and is not applicable if the ‘Representation of clusters’ feature is false.
- **Number of straight lines.** In a sketching system, the lines are unlikely to be geometrically straight, so a visual assessment of straightness was made.

\(^2\) Our notational convention is that version of graph A drawn by participant 3 is called 3A, and the version of graph B drawn by participant 8 is referred to as 8B.
• **Number of vertical or horizontal edges.** In both cases, a visual assessment was made as to whether the edge was intended to be horizontal or visual.

All analysis was done by visual observation, rather than computationally. A detailed computational analysis of these drawings (such as that done by van Ham and Rogowitz (2008)) is beyond the scope of this paper.

The drawings revealed that even though some graphs were drawn with several edge crossings (for example, 11B and 8B), most drawings had been drawn so as to minimise crossings: 16 drawings had zero crossings, while 4 had one crossing and 4 had two. It is clear that this was the most important layout aesthetic used by the participants, and this result concurs with other empirical studies (e.g. Purchase, 1997) and with van Ham and Rogowitz’ (2008) findings.

Most participants recognised the presence of the two clusters, and separated them appropriately. Van Ham and Rogowitz (2008) explicitly asked their participants to layout the graphs so that they best ...reflect[ed] their interconnections; in our experiment we were not so explicit in our instructions regarding the connectedness of the graph – despite this, our participants still performed well when it came to grouping clustered nodes together. In addition, most of them surrounded their clusters with edges, forming clear hulls.

<table>
<thead>
<tr>
<th></th>
<th>GA</th>
<th>GB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Edge crossings</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>1.35</td>
<td>3</td>
</tr>
<tr>
<td>Median</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Max</td>
<td>2</td>
<td>22</td>
</tr>
<tr>
<td>Number of graphs with zero crossings</td>
<td>11</td>
<td>6</td>
</tr>
<tr>
<td><strong>Clusters and Hulls</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of graph drawings with two clusters</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>Percentage of visible clusters represented as hulls</td>
<td>70%</td>
<td>100%</td>
</tr>
<tr>
<td><strong>Vertical/ horizontal edges</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean percentage edges horizontal or vertical</td>
<td>38%</td>
<td>29%</td>
</tr>
<tr>
<td>Number of graphs clearly drawn with a grid arrangement in mind</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td><strong>Straight lines</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean percentage of edges straight</td>
<td>89%</td>
<td>87%</td>
</tr>
<tr>
<td>Number of graphs drawings with all lines straight</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>Number of graph drawings with no edges straight</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2: Features of the 17 drawings of graph A and 17 drawings of Graph B

Participants were reluctant to produce final drawings with curved lines: although only 18 of the 34 graphs comprised only straight lines, the overall percentage of straight lines is high, and curved lines are typically used at the edge of the drawing to avoid edges crossing nodes or other edges. SketchNode straightens associated curved edges when nodes are moved, but this affected the nature of the final drawings for only two participants (4 and 14). In both these cases the participants drew the whole graph first before moving any nodes, indicating that they were unconcerned with the initial shape of the graph (including the curved edges) as they knew that they were going to subsequently adapt the whole drawing to make the layout more acceptable.

Approximately one third of all the edges were aligned along the horizontal or vertical axes, and ten of the 34 drawings were clearly drawn with a grid-like formation in mind. Van Ham and Rogowitz (2008) do not analyse their drawings with respect to horizontal or vertical edges, and their published examples show little evidence of this feature. It is possible that their starting configurations of circular and spring layouts, neither of which favour the presentation of edges along grid lines, may have meant that their participants were biased away from this feature.

### 4.2 The process: how were the graphs drawn?

Analysing the screen cast videos proved very revealing, in that it showed that the final product seldom represented the layout strategy used by the participant when drawing the graph, and that aesthetic criteria emphasised by participants early in the creation process were often compromised as the graph grew in size.

The videos of the creation process were analysed for the following features (Table 3):

- The order and timing of drawing the nodes.
- If and when nodes or sub-graphs were relocated.
- Use of straight or curved lines.
- The variation in the length of edges.
- Alignment to a horizontal/vertical grid.
- Evidence of participants analysing the adjacency list and planning ahead before drawing.

As expected, Graph B took significantly longer to draw than Graph A, as it had more edges. However the time taken to draw the graphs varied considerably between participants (Figure 5). Analysis of the screen casts revealed that there was seldom a clear break between the process of creating the drawing (i.e., representing all the information in the adjacency list) and the process of laying the graph drawing out so as to make it “easy to understand.” In most cases, node positioning decisions were made during the creation process. This means that no timing data could explicitly be associated with the process of graph layout. This is unlike the research of Van Ham and Rogowitz, (2008) who, because graph creation was not part of the task that they set their participants, have clear data on the time taken for the graph layout process and are therefore able to make layout time comparisons between different graphs and different initial layouts.
<table>
<thead>
<tr>
<th>Strategy</th>
<th>Number of participants (out of 17)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drawing nodes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Draw all nodes first</td>
<td>6</td>
<td>Two participants changed their strategy; for their first graph, they drew nodes where convenient, for their second graphs, they drew all the nodes first.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Draw nodes where</td>
<td>11</td>
<td>One participant used a mix of the strategies</td>
</tr>
<tr>
<td>convenient for the</td>
<td></td>
<td></td>
</tr>
<tr>
<td>placement of the</td>
<td></td>
<td></td>
</tr>
<tr>
<td>next edge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Moving nodes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move nodes during</td>
<td>10</td>
<td>One participant changed strategy: in the first graph, he moved nodes and sub-graphs during creation; for the second graph, he didn’t.</td>
</tr>
<tr>
<td>creation of the graph</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move sub-graphs</td>
<td>9</td>
<td>One participant moved nodes during creation, but not sub-graphs</td>
</tr>
<tr>
<td>during creation of</td>
<td></td>
<td></td>
</tr>
<tr>
<td>the graph</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lines</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mainly straight lines</td>
<td>14</td>
<td>This process feature is derived independently of the number of straight lines in the final product, as SketchNode straightens curved edges when nodes are moved.</td>
</tr>
<tr>
<td>Only use curved lines</td>
<td>10</td>
<td>These participants favoured straight lines, and only introduced curved lines when necessary.</td>
</tr>
<tr>
<td>to avoid crossings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use similar length</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>lines during creation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alignment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Favour horizontal</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>and vertical edges</td>
<td></td>
<td></td>
</tr>
<tr>
<td>during creation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyse</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plan ahead</td>
<td>5</td>
<td>This was evidenced by excessive pauses, or by self-reporting</td>
</tr>
</tbody>
</table>

Table 3: Features of the graph drawing process, showing how many participants adopted the different strategies

Figure 5: Box plots showing the median and range (in seconds) of the time taken for the participants to draw the graphs.

Figure 6 shows snapshots of the creation of three of the drawings, each demonstrating a different strategy.

12A has all the nodes drawn first, and then moved around as necessary, so as to allow for easy insertion of edges; 13B has the nodes drawn wherever convenient, avoiding crossings at the start, introducing crossings as the graph becomes more complex, and then relocating nodes at the end so as to remove crossings; 17A has nodes placed where convenient, and never relocated, with curved edges used so as to avoid edge crossings.

There were clearly two different strategies in the creation of the graph drawing: drawing all the nodes first and then moving them as necessary when creating edges, or drawing nodes where convenient. The latter was favoured. Participants tended to place the next node they came across in the adjacency list near to the node it needed to be connected to so as to avoid edge crossings.

There were also two different strategies when it came to producing the layout of the graph: movement of nodes during creation of the graph, and movement after the whole graph was complete. The former was favoured. There were even a few participants who did not move the nodes at all once they had been placed conveniently next to their first adjoining node (e.g. participants 7, 8 and 10), even if this meant introducing edge crossing or curved edges.

The use of these different strategies, and the fact that only 8 participants moved nodes after the graph was created, suggest a reluctance on the part of some participants to move nodes once placed, and a possible misunderstanding of the task (which was to arrange the final graph so as to make it easy to understand). It appeared that some participants may have focussed on ease of process (i.e. ease of creating the graph) rather than on the form of the final product. While our experiment removed the layout bias of van Dam and Rogowitz’ (2008) work by asking participants to draw the graph
from scratch, asking them to effectively do two tasks (creation and layout) makes it more difficult (and in some cases impossible) to focus our analysis on the process of layout.

Analysing the participants’ drawing and layout processes allowed us to see where layout aesthetics had been favoured during the creation process, but abandoned later when the graph became more complex. These aesthetics are therefore not evident in the final product, even though there were considered important. Most noticeable of these was the tendency for participants to place their nodes as if on the intersections of an underlying unit grid, and to draw edges vertically and horizontally. So while only 10 of the resultant graph drawings had evidence of a grid-like formation (representing 6 participants), a further four participants attempted to use horizontal and vertical lines, but abandoned this feature as more edges were added.

Similarly, most participants attempted to use only straight edge, and the curved edges tended to be the very last edges to be drawn. Most participants also attempted to use edges of similar length: a feature that is evident in very few of the resultant drawings.

Five students performed some pre-drawing analysis and planned ahead. Planning ahead is, of course, unnecessary, as any nodes could be moved to a more appropriate position at any time. This again suggests a reluctance of the participants to move nodes and change the overall layout of the drawing.

The appendix shows snapshots of the creation of two versions of graph A (3A and 3B) and two versions of graph B (15B and 16B) both of which have similar final products.

4.3 The preferences: what did the participants think?

In the post-experiment interview, the participants were encouraged to articulate their strategy in drawing the graphs, both in terms of the product and the process. No specific features of graphs were suggested to the participants in this interview, and they were encouraged to describe the features in their own words.

All but four of the participants emphasised the need to avoid edge crossings; three of these said that avoiding crosses was not important, one of whom said that edge crossings are acceptable if the cross is at right angles. Two of those who said that edge crossing were not important specifically said that avoiding edges crossing nodes was more important.

Four participants mentioned a preference for horizontal or vertical alignment of edges, with one also favouring 45 and 60 degree angles. Five participants said that straight lines were preferred, while one specifically said that they were not important. Other features mentioned were maintaining a similar distance between nodes (2), spreading the nodes out (2), putting the nodes in groups (clusters) (2) and symmetry (1).

With respect to the graph creation process, four participants specifically mentioned their use of the node with highest degree: two participants placed it centrally, one participant placed it at the top (while mentioning a desire to draw the graphs in a tree-structure), and another participant just made sure that the node with the highest degree was drawn first.

Two participants said that they preferred to place the nodes in alphabetical order at the beginning of the drawing process.

When asked why they drew the graphs the way that they did, most participants used phrases like ‘most logical’ (3), ‘easier’ (4), and ‘neater’ (4). Two participants specifically said that they looked ahead, making room for future edges that would be added.

### 4.4 Demographic effects

There are insufficient data points for a complete statistical analysis, but we can make some informal observations on product, process and preferences when considered with respect to the important demographic properties of Computer Science/graph experience, and gender, both of which have appropriate splits to allow for reasonable analysis (Table 4).

The data suggest that the male participants favoured straight lines, horizontal and vertical lines and a grid formation (both during the process and in the final product) more than the female participants. There was no obvious gender difference in strategy, edge crossings or clusters.

<table>
<thead>
<tr>
<th>Gender</th>
<th>CS/graph experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>M (n=8, 16 graphs)</td>
<td>Yes (n=9, 18 graphs)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product</th>
<th>F (n=7, 14 graphs)</th>
<th>M (n=10, 20 graphs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of crosses (mean)</td>
<td>2.21</td>
<td>2.15</td>
</tr>
<tr>
<td>Percentage lines straight</td>
<td>76%</td>
<td>96%</td>
</tr>
<tr>
<td>Percentage lines horizontal/vertical</td>
<td>23%</td>
<td>41%</td>
</tr>
<tr>
<td>Number of drawings aligned to a grid</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>Number of clusters (mean)</td>
<td>1.1</td>
<td>1.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Process (number of participants, out of 17)</th>
<th>F</th>
<th>M</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drawing all nodes first</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Applying layout after creation</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Aligning to a grid during creation</td>
<td>2</td>
<td>8</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>Planning in advance</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Mean time for both graphs</td>
<td>11m36s</td>
<td>11m35s</td>
<td>13m7s</td>
<td>10m32s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Preferences (number of participants stating preferences, out of 17)</th>
<th>F</th>
<th>M</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>No crosses</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>Grid configuration</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Straight lines</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Special use of node with highest degree</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 4: The graph drawing product, process and preferences with respect to the demographics**

Computer Science and graph experience seems to have made a difference in several factors: reduction in the number of edge crossings, increased percentage of straight edges and number of clusters, as well as a higher instance of horizontal and vertical edge and grid-like graph drawings. However, it is interesting to note that three of the participants who did not have Computer
Science experience aligned nodes and edges to an underlying grid while creating the drawings, even if their final graph drawings do not show evidence of this feature being favoured. Participants with Computer Science and graph experience tended to explicitly choose a strategy (drawing nodes first or laying out all nodes at the end), while the others were more likely to pause to plan their actions during the creation process. Only participants with Computer Science and graph experience recognised the importance of the node with highest degree. Non-Computer Science participants tended to perform the graph drawing tasks more quickly.

5 Discussion

5.1 Layout aesthetics and the graph drawing process

With respect to the layout aesthetics of the resultant graph drawing, it is not surprising that the minimisation of edge crossings has again been revealed as most important. The data presented here gives more weight to the principle of fixing edges and nodes to an underlying unit grid than has previously been shown to be the case in empirical studies of graph drawing comprehension (Purchase, 1997). This latter point is surprising, suggesting that users would like to see their graphs fixed to a grid, even if doing so does not necessarily assist in improving their performance in graph reading tasks.

Only one of the published graph drawings from the research of van Ham and Rogowitz (2008) shows any indication that the participants were interested in horizontal or vertical edges: we suggest that this is due to the layout bias (circular or spring) in the initial drawing presented to the participants.

When considering the creation strategies, two particular methodology issues stood out. First, a final graph drawing often does not reveal the aesthetic principles that the participants have tried to adhere to during the drawing process, and much can be missed by concentrating on the product rather than the process. Second, researchers in the area of Graph Drawing tend to make a clear distinction between the process of creating a graph and the algorithm for laying it out. This does not appear to be the case for our participants: the action of presenting relational information in as best a way as possible using a graph drawing encompasses both the initial visual representation of the nodes as well as their relocation. The fact that we cannot separate the process of creation from the process of layout in our data is a confounding factor when comparing our timing data of that of van Ham and Rogowitz (2008), as in their case the participants’ task was purely one of layout.

Informal demographic analysis suggests a tendency for male participants with some Computer Science experience to particularly favour a grid layout.

5.2 Limitations and Future work

While this experiment has successfully addressed some of the limitations of the work of van Ham and Rogowitz (2008) (as described in section 2 above) it has itself been subject to some methodological constraints. We only have 17 participants and 34 graphs, which makes any extensive demographic analysis and generalisation of results difficult; the strategy employed by most of our participants was to interleave the creation and layout processes, making it difficult to analyse the process of layout separately, and both our graphs were small. Future work would, of course, entail the drawing of larger graphs (possibly using a digital whiteboard or similar technology), and would attempt to clarify to the participants the usefulness of manipulating the nodes in the graph after they have been created.

No computational analysis has been performed on this data; such analysis could, for example, determine (as in van Ham and Rogowitz (2008)) the edge-length distribution, the orientation and the extent of clustering as derived from inter-node differences.

6 Conclusion

The van Ham and Rogowitz (2008) experiment considered how users would manipulate graph drawings so as to improve their layout. The research reported here extends this work significantly with the use of sketch-based graph drawing software, by removing any initial layout bias, and by including consideration of key participant demographics. While some results (e.g. the importance of minimising crosses) clearly follow several previous findings, we have also established the importance of a grid-based layout, and shown that the layout of a graph drawing should not simply be judged by the product, but should also be considered in the light of the process that created it.

7 Acknowledgements

We are grateful to the experimental participants, and to Hong Yul Yang, who implemented SketchNode. Ethical clearance for this study was granted by the University of Auckland, 2008.

8 References


Appendix

Graph drawings 3A and 3B are similar, as are 15B and 16B. However, the process of drawing in each case is different, as shown by the snapshots shown below.
Identifying Cultural Design Requirements for an Australian Indigenous Website

Reece George, Keith Nesbitt, Patricia Gillard, Michael Donovan
University of Newcastle
University Drive, Callaghan 2308, NSW, Australia
{reece.george, keith.nesbitt, patricia.gillard, michael.donovan}@newcastle.edu.au

Abstract
This paper examines literature concerning the design of culture into websites and in particular indigenous websites. The intention is to identify design requirements as the first phase of building an indigenous website for the Wollotuka Institute located in the Awabakal nation and situated at the University of Newcastle, NSW, Australia. The aim of the project is to produce a website that best reflects the identity, needs and culture of the local Wollotuka community. Wollotuka supports a broad range of indigenous programs incorporating administrative, academic and research activities and provides support and development services for indigenous staff and students. Unfortunately many existing frameworks and indeed Western conceptions about what knowledge is and how it should be captured, organised and presented do not necessarily meet the culture of the intended indigenous users. The work described in this paper focuses on the many issues of concern when attempting to design a website that is specific for a local cultural community such as Wollotuka. We review previous work in cultural design and discuss some generic issues related to the representation and capture of indigenous knowledge. This review provides us with some general considerations and more specific guidelines for culture-specific design. Of particular interest is that narrative and object are conceptualised as a duality in knowledge representations found in Australian Indigenous culture. We also examine issues of design process and use our findings to support the choice of a user-centric design method, where we localise the design, through an iterative, prototyping process.

Keywords: culture specific, web design, indigenous website, focus group.

1 Introduction
The Wollotuka Institute is situated at the University of Newcastle, NSW, Australia and located in the Awabakal nation. Wollotuka was formed to consolidate all the Indigenous activities of the University into one operational and strategic body in order to serve the University's strategic priority and commitment to Indigenous Collaboration. The four basic functions of the Institute incorporate academic, research, Indigenous student support and the employment and development of Indigenous staff. The staff are comprised of Aboriginal and Torres Strait Islanders.

Wollotuka is housed in a specially designed learning space, the Birabahn building, which was developed to incorporate aspects of Indigenous practices and culture and to present staff, students and community with a warm familiar environment. The design of the Birabahn building incorporates the motifs of the Eaglehawk, which is a primary totem of the Awabakal nation. In a similar way we set out to study the design and development of a web site that incorporated the knowledge motifs or culture of the local community. For, while Wollotuka already provides an existing web site (Wollotuka 2009), it has been designed to fit a more traditional western, corporate framework as used by the University.

Designing culture into a website may seem at first to be a relatively simple task, but it soon raises a number of difficult questions, such as “What is culture?” and “How do you capture culture in a web page?” More general questions are raised about culture and usability and the localisation of user interfaces. Furthermore, some fundamental questions of aesthetics and schemas for knowledge representation also arise. For example, “How should we represent knowledge in an indigenous versus western way?”

With these questions in mind we reviewed existing literature in the area to provide an overall guide to our design work. As is often characteristic of design we identified many guidelines and issues to assist in our work although no prescriptive approaches were identified. To address this problem we also examined processes we could adopt for the project and decided to use an approach founded on user-centric design and iterative prototyping with regular feedback from our target community. However, before we consider the question of “How to capture indigenous culture in a web page?” we return to the fundamental question of “What is culture?”.

2 What is Culture?
“Culture” is a complex concept and can mean different things in different contexts and to different people. Indeed Kroeber and Kluckhohn provide 200 different definitions for the concept of culture (1952) and there is a large body of historical work in the area of Cultural
Theory (Cashmore & Rojeck 1999). Culture has also been studied in other diverse fields including, philosophy, economics, sociology, literary criticism, linguistics, psychology and human computer interaction. In a general sense, we understand “culture” to mean the way of life for an entire society or community, the sum total of all the ways of life including arts, beliefs and institutions of a population that are passed down from generation to generation.

It is difficult to analytically describe aspects of individual cultures although the cultural theorist Geert Hofstede did develop a four (later five) dimensional cultural model for classifying different national cultures (Hofstede 2005). Hofstede’s model is typically used in business studies but has also been applied in the domain of human computer interaction. This model is of interest to us because there have been several studies looking at the relationship between his cultural model and the design of user interfaces and websites (Callahan 2005; Marcus & Gould 2000; Robbins & Stylianou 2002; Yuan et al. 2005; Singh 2005).

Hofstede describes culture as the “collective programming of the mind which distinguishes the members of one category of people from another.” (Hofstede 2005, p.51). Culture is described as a collective phenomenon because it is shared with the people who live in the same social context where it is learned. He stresses that culture is learned and not inherited and distinguishes culture from human nature which he considers a universal aspect of mental programming and something that is inherited. He further distinguishes culture from an individual’s personality that he describes as partly learned and partly inherited. A number of different layers of culture are also described, from a national level, to a regional, ethnic, religious or linguistic level and then a gender level. Further levels may be related to the person’s generation, their social class and the work culture of the organisation they work for. These different levels all relate to the different types of social groups that a person may interact with. In Hofstede’s terms cultural differences are seen as manifesting themselves in several different ways, such as the symbols, heroes, rituals and values of the group. In this sense, symbols are words, objects, and gestures that carry a particular meaning that can only be recognised by that culture. Rituals are group activities that serve no purpose and yet are considered socially essential activities within the group. Hofstede describes the core of culture as being composed of values, which are broad preferences or feelings for a particular state of things. For example, values would determine what is evil or good, what is ugly or beautiful and what is normal or abnormal.

As we have already noted there has been some work relating Hofstede’s model to the design of user interfaces and so this is where we begin our examination of culture and design. The four, original dimensions of Hofstede’s model are: power distribution; individual versus collective relationships; masculine versus feminine characteristics and the tendency to avoid uncertainty (Hofstede 2005). Hofstede’s model was based on a survey completed by staff working for subsidiaries of IBM across 50 different countries between 1967 and 1973. This model was later extended to include a fifth dimension (Hofstede 2005) that distinguished between short-term and long-term orientation. Some examples of national cultures and estimated values for an indigenous Australian group are listed in Table 2.

The power distance index is related to the extent that power is distributed in the culture’s society. Higher values indicate that power is exercised centrally from above, while lower values indicate a more even spread of power through all levels of society. This has been suggested to impact on website design in terms of the level of structure, use of national and authority symbols, the emphasis of leaders versus general community in the web page and the level of security (Marcus & Gould 2000).

The individualism measure relates to the way larger, strong cohesive social groups function as opposed to smaller individual and tight family groupings. We might typically associate Asian cultures with a lower value of individualism compared to western cultures such as America and Australia. This has been suggested to influence how different images are used on a website, individuals versus groups as well as rhetorical styles and the use of young versus older and new versus traditional knowledge (Marcus & Gould 2000).

The masculinity index is intended to estimate the way roles are distributed between genders in the culture. While female values were found not to vary greatly between cultures, male attitudes were found to vary greatly between cultures. They could be very similar to female attitudes where roles were often shared to the other extreme where they were maximally different in terms of assertiveness, modesty and competitiveness. It is suggested that interfaces for higher masculinity cultures should focus on providing efficient results for a limited number of tasks along with an exploratory style of navigation and the use of competition and games. In contrast more feminine cultures would blur gender roles and support mutual cooperation (Marcus & Gould 2000).

The uncertainty avoidance index measures the tolerance for ambiguity and uncertainty and indicates the acceptance or not of less structured or surprising situations. Cultures with higher uncertainty avoidance measures are suggested to prefer simple limited choice interfaces. Navigation schemes should focus on preventing user becoming lost, and redundant cues can be used for reduce ambiguity and user errors. By contrast cultures with a low score on this index may enjoy more uncontrolled navigation in complex web sites focusing on more content and a maximum use of information cues (Marcus & Gould 2000).

The long-term orientation measure was added to the Hofstede model after some criticism that it failed to capture some key differences between some cultures. Values such as thrift and perseverance are associated with a high long-term orientation while respect for tradition and meeting social obligations are relates with lower values. Long-term cultures suggest a preference for content in websites that provide practical value, while
users will have greater patience in achieving their goals. Short-term cultures will seek immediate results from the interface and rely on rules as a source of information and credibility rather than truth being interpreted based on relationships (Marcus & Gould 2000).

Hofstede’s model was used in a study of university websites in the Netherlands and Austria. The study found some correlations between feminine values and the masculinity index in the low masculinity country of Netherlands compared to the high masculinity culture in Austria (Dormann & Chisalita 2003). When Indian and American university websites were compared differences in the design were measured in the three dimensions of; uncertainty avoidance, individualism and long-term orientation (Rajkumar 2003 cited in Callahan 2005). A study by Callahan (2005) compared the way web sites were structured as well as the types of graphical elements that were used in university web pages from eight different countries. This study found a general correlation with Hofstede’s cultural dimensions.

If we compare Hofstede’s cultural dimensions for Australia as a whole and the indigenous Australian group surveyed in Cape York, Queensland (Simonsen 1999; Hofstede 2005) we note that the indigenous group has a higher centralised power measure, a similar individualism ranking, a lower measure for the masculinity index and a much higher risk avoidance along with longer term orientation. If we accept the outcomes of Marcus and Gould (2000) this would suggest we design web pages for this indigenous group with some emphasis on leaders, with a simple interface with practical content and good navigation cues. Of course we emphasise that these measures are for a regional group in Queensland and that cultural values for our indigenous target group may well be different. Further work to measure these values is ongoing.

We should note that Hofstede’s cultural model is not without criticisms, and some of these include the use of an initial sample made up of employees from a single company and then how well these relate to the national culture as a whole (Sondergaard 1994). There is also the question of whether the survey is even an appropriate method to measure culture and whether the data, much of which is from around 1970 is still relevant to current cultures (Sondergaard 1994). Callahan provides a good review of these issues and the debate in general surrounding Hofstede’s model (Callahan 2005). Despite the criticisms, we believe, like many others, that Hofstede’s model provides a valuable framework for considering some issues that we need to address in our design. It is certainly not prescriptive and at best can lead to general guidelines to assist in the design process. On the positive side it does also provide a well-known, and structured framework for studying culture.

### 3 Culturability

We have considered Hofstede’s model of culture and it’s relationship to issues of web design. In this section we discuss more general work related to developing interfaces that are targeted for a specific culture. A summary of these issues is provided in Table 2.

The term “culturability” is used to describe the merging of culture and usability (Barber & Badre 1998). A number of culturability issues have certainly been identified for designers to consider. These include language, social context, time, currency, units of measure, cultural values, body positions, symbols and aesthetics (Fernandez 2000). A study by Ewa Callahan (2005) found that designers use different iconography and also notes the different standards for representing dates, time, and numbers” (257). Similar results from a further study provide a checklist of items to consider, including: “images, symbols, colours, flow, and functionality.” (Vatrapu 2002, p.17).

A general issue in cultural sensitive interface design is “understanding representation.” (Bourges-Waldeg & Scrivener 1996) These authors argue that factors such as religion, government, language, art, marriage and a sense of humour are universal, it is simply the way they are represented that is specific to each culture. A study that supports this view looks at how different cultural users responded to various visual stimuli in websites (Tsai et al. 2008). The conclusion was that information uncertainty, as characterised by the use of visual elements, does indeed have a direct impact on the user’s emotional

### Table 1: Hofstede dimensions* for a selection of countries (Hofstede 2005)

<table>
<thead>
<tr>
<th>Cultural Group</th>
<th>PDI</th>
<th>IDV</th>
<th>DVI</th>
<th>IDV</th>
<th>MAS</th>
<th>UAI</th>
<th>RAI</th>
<th>LTO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australia (western)</td>
<td>36</td>
<td>90</td>
<td>61</td>
<td>51</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Australia (indigenous)</td>
<td>80</td>
<td>89</td>
<td>22</td>
<td>128</td>
<td>-10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Austria</td>
<td>11</td>
<td>55</td>
<td>79</td>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>China</td>
<td>80</td>
<td>20</td>
<td>66</td>
<td>30</td>
<td>118</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Finland</td>
<td>33</td>
<td>63</td>
<td>26</td>
<td>59</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>India</td>
<td>77</td>
<td>48</td>
<td>56</td>
<td>40</td>
<td>61</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Japan</td>
<td>54</td>
<td>46</td>
<td>95</td>
<td>92</td>
<td>80</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Netherlands</td>
<td>38</td>
<td>80</td>
<td>14</td>
<td>53</td>
<td>44</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>South Korea</td>
<td>60</td>
<td>18</td>
<td>39</td>
<td>85</td>
<td>75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>United Kingdom</td>
<td>35</td>
<td>89</td>
<td>66</td>
<td>35</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>United States</td>
<td>40</td>
<td>91</td>
<td>62</td>
<td>46</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Note that this is characterised by a higher value on a scale generally measured by a standard survey instrument to be between 0-100. However lower and higher values are sometimes used when these dimensions are estimated.
response and this was directly related to the user’s culture.

A important relationship has also been identified “between language, cultural context and usability” (Hillier 2003). On the general issue of usability Faiola and Matei contend that different cultures employ different usage strategies (2004). They explore “cultural cognition theory,” to look at how web design is impacted by the designers’ cognitive style. In an on-line experiment they used websites created by both American and Chinese designers and found that users performed information-seeking tasks faster when they used websites created by designers from their own cultures. Using a well-known cognitive model that distinguishes between holistic and analytical thought (Nisbett et al. 2001) a further cross-cultural study of Chinese, American and Korean users found that the user’s cognitive style was reflected in their perception of the usability of the web site (Dong & Lee 2008).

A recent study even questions if the very notion of usability is constant across cultures (Frandsen-Thorlacius 2009). This work used a sample of 412 users from China and Denmark, and found a basic difference in how the users understood and prioritised different aspects of usability. For example, Chinese users were more concerned with visual appearance, satisfaction, and fun than the Danish users. The Danish users nominated effectiveness, efficiency, and lack of frustration as more important factors in the interface. It has also been pointed out that cultural difficulties in understanding a web site have a natural flow on effect on the learnability of the system (Bourges-Waldegg & Scrivener 1996).

As Nielsen points out: “an interface which is used in another country than the one [for which] it was designed, is a new interface. One cannot trust the original usability work on the user interface to necessarily have produced a design which will be equally usable around the world.” (Nielsen 1990, p.39). Other studies also conclude that usability evaluations need to carefully consider the cultural characteristics of the target group as they may, or may not, impact on the outcomes depending on the group being studied (Herman 1996; Dray 1996).

Much of the research in HCI regarding culture and usability has surrounded the internationalisation-localisation process (Bourges-Waldegg & Scrivener 1996). Internationalisation “seeks to eliminate culture” (Young 2008b) by eliminating cultural symbols, religious references, and so on, while localization caters to the needs of the local target group (2008b p.2) and is intended to incorporate local content and functionality (Shannon 2000) as well as local, context and culture (McLoughlin & Oliver 1999).

Localization of a web site therefore intends to make the site completely appropriate for the user and goes well beyond mere translation (Tixier 2005). When localizing a website interface, possible options include the change of language, time zones, currency, local colour sensitivities, product or service names, gender roles, and geographic examples (Cyr & Trevor-Smith 2004). A user looks for “perfectly clear and understandable information, but he does not want to be culturally offended by language, images, colors, and so on” (Sandrini 2005). It is therefore the designer’s role to ensure that web sites for international audiences do not contain “symbols that have different meanings in various cultures, variations in language dialect and idiom, starkly different cultural reactions to the same colour, and a host of similar nuances and subtleties that can deeply offend and/or confuse a potential customer or trading partner” (Dysart 2001). For example “in some Asian sites the icon representing home is a pair of shoes, instead of a little house” (Fernandez 2000, p.20). Icons and symbols may also have culturally different meanings. In studies by Brugger (1990), only 13 percent of Japanese recognized a first-aid symbol based on the Red Cross, and most did not associate the symbol letter ‘I’ as referring to information services.

<table>
<thead>
<tr>
<th>Design Issue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use of non-local images (scenes, faces, architecture, and customs) can affect learnability (Barber &amp; Badre 1998)</td>
<td>New technical words in other countries have to be created by adapting English words or creating new ones based on native concepts (Callahan 2005)</td>
</tr>
<tr>
<td>Design to fit the local writing style. e.g. languages such as Arabic are written right-to-left (Amar &amp; Porterani 1996)</td>
<td>Translation of the menus, boxes, and icon text can also be problematic because the length of words varies between languages. (Dray, 1996)</td>
</tr>
<tr>
<td>Icons based on metaphors such as the mailbox, trashcans may be interpreted differently (Duncker 2002) (Shen et al. 2006)</td>
<td>Care should also be given to the presentation of pictures. Some cultures are very sensitive to how human features are represented (Russo &amp; Boor 1993)</td>
</tr>
<tr>
<td>Icons considered international are not necessarily understood globally (Brugger 1990)</td>
<td>Cultures vary in how they present numbers, time, and dates (Callahan 2005; del Gardo 1990)</td>
</tr>
<tr>
<td>Specific orientations and page placement vary by culture (Barber &amp; Badre 1998)</td>
<td>Use of colour in web design can impact on the user’s expectations about navigation, content, and links, as well as overall satisfaction (Barber &amp; Badre 1998)</td>
</tr>
<tr>
<td>The way holistically versus analytically minded people scan a web page is different. Ordering and arrangement of information needs to be considered (Dong &amp; Lee 2008)</td>
<td>Using icons versus text for navigation can affect error rates and task completion times depending on culture (Choong &amp; Salvendy 1998)</td>
</tr>
</tbody>
</table>

Table 2: Issues of Culturability and Localisation
4 Representing Indigenous Knowledge

The work described previously has identified a number of factors, some general, some more specific that need to be considered in the design of culturally specific websites. These issues range from cognitive style through to the many manifestations of culture such as symbols and rituals and even the choice of colours, fonts and formatting. Now we investigate work that is more specific in the area of indigenous knowledge representation and discuss previous works that concern the design of indigenous websites. A summary of these issues is provided in Table 3.

The idea of a motif or schema for representing knowledge concepts has been well studied in the domains of psychology, education and cognitive science. The term, ‘schema’ was originally introduced by Bartlett in 1932 to describe a set of abstract mental structures used to represent an individual’s understanding of the world (Bartlett 1932). For example, in order to recognize a landscape as a landscape, we must first cognitively create a schema of a landscape as an object that is composed of trees, hills, rocks, ground, sky, and wildlife in a particular order, and then recall the schema when we see it. Schema help people organise their current understanding of the world and help them adapt future knowledge into a framework they already understand. Schemata are not limited to physical objects and include knowledge about different types of people, social roles, sequences of events and processing rules (Nishida et al. 1998). Different schemata may need to be constructed for each individual culture.

Any representation of Aboriginal knowledge needs to address the characteristics, the epistemology and ontology of their culture practices. (Pumpa et al. 2006). This includes performances of traditional practices such as dance and song. This is also found in other indigenous cultures such as the Maori, people of New Zealand whose oral culture transfers knowledge by stories, songs, and art, rather than by written texts (Duncker 2002). Aboriginal modes of communication are “extensions of the oral and face-to-face nature of that society” (Buchtmann 1999). Aboriginal stories and songs are “the prerogative of senior men and women (elders) and the rules governing transmission are highly regulated.” (Buchtmann 1999).

It is also clear that the landscape plays an important role in Australian indigenous stories (Turner 2006). In a study on Aboriginal knowledge systems, Malcolm Pumpa and Theodor G Wyeld (2006) noted: “In Aboriginal knowledge, the landscape is both the visualization of knowledge and the narratives of knowledge. Therefore, any attempt to present the complexity of Aboriginal knowledge and practices in a digital environment needs to provide a visualization of the landscape that is capable of embedding a wide range of knowledge objects.”

Indigenous knowledge can also be thought of as reifying culture and identity and existing within “kinship, language and humour.” (Pumpa & Wyeld 2006). This contrast with “Western traditions, which emphasize the differences between what exists and how we represent it in a variety of symbolic systems”. Aboriginal knowledge traditions “emphasize the unity of symbol and object—of what exist and how we represent it (Pumpa & Wyeld 2006). Indeed in “Aboriginal knowledge traditions, language, ceremony, singing, dancing and other representational forms can influence events and cause real world events to happen. Objects and phenomena can be ‘sung’ or ‘talked’ into and out of existence. These processes of the amalgamation of representation and reality have been going on since the Dreamtime (in Australian Aboriginal terms, the time of creation of things) and continues to this day.” (Pumpa & Wyeld 2006).

One significant project in the area of Australian indigenous web design began in 2003 and is called Digital Songlines (DigitalSonglines 2009). This is a detailed project, which examines how Aboriginal knowledge practices have been, until recently, represented in digital media (Pumpa and Wyeld, 2006). The study looks at how to best represent traditional indigenous knowledge using a 3D game engine (Pumpa et al. 2006, 2007). The researchers contend that any digital representation of Aboriginal knowledge must “resolve the conflict between database and narrative views of knowledge”, asserting that they need to allow for the “environmentally contextualised narratological nature of Aboriginal knowledge traditions”. They contend that the use of a game engine enables a landscape metaphor for hosting Australian Aboriginal knowledge practices based on performance narratives and that formats such as; audio, video and graphics which can be accommodated in this way are is better suited for maintaining oral traditions of knowledge exchange (Pumpa et al. 2006)

Typically in a Western database to visualise something, it is first “objectified” and then organized into related categories. However, Aboriginal knowledge objects are non-representational in nature and are embedded in a networked narrative of story, song, dance, art and ceremony. The narrative embeds any knowledge ‘object’ in a matrix of relationships that is both temporal and spatial. Aboriginal knowledge therefore resists being represented in a conventional rational database schema. The Digital Songline project attempts to solves this problem by immersing participants in a narrative landscape, which has elements of sentence or responsiveness. This is achieved by embedding data objects the data object in “an interconnected network of multi-layered pathways or Songlines” (Pumpa et al. 2006) The authors contend that this catches the Indigenous tradition effectively as the end result “is a vehicle for the unfolding of real-time narratives involving Elders and the ancestral spirits of the landscape.” (Pumpa & Wyeld 2006).

The Digital Songlines project also examines the role of the user in any valid representation of Aboriginal knowledge. They believe the user is to be “involved in an extended collaborative, performative narrative which pursues a purposeful journey through a sentient (responsive) landscape, exploring and reaffirming relationships with significant people and the land.” (Pumpa & Wyeld 2006). This implies an essential
requirement is for the user to ‘perform knowledge’, that is to actively participate in knowledge construction, rather than merely accessing and manipulating what is provided” (Pumpa & Wyeld 2006).

Most importantly, contemporary Aboriginal knowledge also finds its way into the project. They cite an example: “we have developed some animated sequences from an Aboriginal dreamtime story that is included as part of the world in a transparent manner.” They also stress the involvement of the Indigenous people in the development of the programme. For instance they refer to “the regular consultation with Indigenous artists and representatives from the country” in an attempt to ensure that they are represented correctly. An important observation from the project is that “Aboriginal people in many Australian communities have a real desire to preserve and pass on knowledge practices in spite of the decline in their use of traditional language, loss of ritual and passing away of elders.” This discussion of indigenous involvement in the design process informs our own intended design approach which will occur in close collaboration with our intended community of users.

<table>
<thead>
<tr>
<th>Design Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aboriginal students prefer simple, “straight to the point” and easy to read English (Gibb 2006)</td>
</tr>
<tr>
<td>Familiar images of concrete things that are understood and loved are the key communication device and are a text in themselves. Use of local pictures and images of people are essential (Williams 2002)</td>
</tr>
<tr>
<td>Navigation by images is preferred over navigation linked to words (Williams 2002)</td>
</tr>
<tr>
<td>There may be use of icons that provide an alternative form of language (Munn 1973)</td>
</tr>
<tr>
<td>There is a preference toward real-time communication (Clemens 2002)</td>
</tr>
<tr>
<td>Respected teachers or elders are typically used to impart knowledge (Trudgen 1983)</td>
</tr>
<tr>
<td>Signing and dancing are often used to teach in the traditional Aboriginal teaching situations (Fischer 1995)</td>
</tr>
<tr>
<td>The geographical land is the foundation of Indigenous thinking. They have a strong respect for the land, as well as their culture and language (Auld 2007)</td>
</tr>
<tr>
<td>Aboriginal community, family life and children always come before individual pursuits (Gibb 2006)</td>
</tr>
<tr>
<td>A site needs to plan for change and provide the ability for redesign as the needs of the target groups change (Clemens 2002)</td>
</tr>
<tr>
<td>Provide multimedia rich environments rather than text based and incorporate a range of audio and visual techniques to encourage usage. (Fischer 1995, Buchtmann 1999)</td>
</tr>
<tr>
<td>Use of our stories, our songs and our images to pass on the message (Remedio 1996)</td>
</tr>
</tbody>
</table>

Table 3: Design Issues of Indigenous Knowledge

5 The Design Process

Our project focuses on a number of key levels of culture. While we are interested in general issues of representing national indigenous culture in a website, we must also address the issues of both regional and organisational culture in the website. While we have used literature to identify some helpful guidelines for our project there are also many specific, local issues that we expect to impact the success of the project. Considering localization as “a process through which a site is not only translated but also adapted to the cultural, technical, and administrative specificities” (Tixier 2005) we appreciate not only the importance of knowledge representation but also the key role to be played in process when we design our website.

Our intention is to employ a user-centric design process carried out in close collaboration with the Wollotuka community. This will involve an initial focus group followed by prototyping and individual interviews with members of the community. Indeed such a user-centric approach that includes focus groups and interviews, as well as prototypes to gather feedback using structured interviews has been recommended for identifying the “meanings” of a representation within cross-cultural “contexts” (Bourges-Waldegg & Scrivener 1996).

User-centred systems design has been highlighted by many researchers and although user involvement might appear to be “complicated and time-consuming” there are many benefits in involving users during all stages of the design process (Goransson 2003; Goransson et al. 2003). The importance of involving the user community in the planning and design stages of any program for indigenous Australians has been borne out by further studies (Johnston 2001; Dyson 2003; Turner 2006; Fernandez 2000). Furthermore a positive preference for real-time communication, face-to-face contacts and one-to-one or small group meetings has been identified (Clemens 2002).

Studies involving cultural sensitive design also support this approach. For localization requires cultural-unique specifications. However it also impacts on the design process because what may be an acceptable process in one culture may not be an acceptable in another (Young 2008b, p.5). For example, the first step in the design process is to collect data about the intended target audience. In this process “Individual user characteristics and variability of the tasks are the two factors with the largest impact on usability, so they have to be studied carefully” (Nielsen 1993). Therefore each target audience for which the interface is intended needs to take part in this stage of design process (Callahan 2005).

Designing and building the web site in close collaboration with the indigenous community also allows for unforeseen problems that arise to be quickly identified and addressed. We do intend to use the identified issues and guidelines discussed in this paper in a formative way but these guidelines to not offer a prescriptive solution and therefore the final design will rely on user participation and feedback to evaluate our design decisions.
6 Conclusion

We have discussed the notion of culture and how it impacts on knowledge representation. A review of previous work has identified a number of issues involved in the design of an indigenous web site. These include general issues dealing with cultural usability and localisation of web sites. More specific issues related to the intended indigenous audience at Wollotuka have also been identified. We intend to use these as guidelines to inform our design work in a formative way. However, due to the difficult nature of designing for specific cultures we also intend to adopt a user-centric approach to the design. This will involve close collaboration with the community, involving focus groups to identify initial requirements and iterative prototypes and interviews at regular stages to refine the design.

7 References

Buchtmann, L 1999, Digital Songlines The Use of Modern Communication Technology by an Aboriginal Community in Remote Australia University of Canberra Canberra.
Fernandez, NC 2000, 'Web Site Localisation and Internationalisation: a Case Study', City University.
Fischer, RA 1995, Protohistoric roots of the network self: On wired aborigines and the emancipation from alphabetic imperialism, Zurich.


Nielsen, J 1993, Usability Engineering, Morgan Kaufmann, San Diego, CA.


Remedio, J 1996, 'Chairperson of the National Indigenous Media Association (quote referenced from Buchtmann. No further information available)'


Trudgen, RI 1983, Aboriginal traditional economic system in Central and East Arnhemland, Northern Regional Council of Congress, Darwin.


Author Index

Abell, Walt, 37
Baker, Rosemary, 87
Calder, Paul, iii
Chang, Samuel Hsiao-Heng, 57
Churcher, Clare, 37, 47
Close, Ben, 77
Cockburn, Andy, 3
Cunningham, Andrew, 77
Daradkeh, Mohammad, 47
Delwadia, Vipul, 7
Donovan, Michael, 97
Doshi, Meghavi, 57
George, Reece, 97
Gillard, Patricia, 97
Hutterer, Peter, 77
Laycock, Laura, 57
Lutteroth, Christof, iii
Marshall, Stuart, 7

McCallum, John, 37
McKinnon, Alan, 37, 47
Nesbitt, Keith, 97
Pitcher, Christopher, 87
Plimmer, Beryl, 17, 57, 87
Puglisi, Simon J., 27
Purchase, Helen, 87
Rutherford, Paul, 37
Scholer, Falk, 27
Seneviratne, Nilanthi, 57
Thom, James A., 27
Thomas, Bruce, 77
Turpin, Andrew, 27
Welch, Ian, 7
Wu, Mingfang, 27
Wuensche, Burkhard, 67
Yang, Rong, 67
Recent Volumes in the CRPIT Series

ISSN 1445-1336

Listed below are some of the latest volumes published in the ACS Series Conferences in Research and Practice in Information Technology. The full text of most papers (in either PDF or Postscript format) is available at the series website http://crpit.com.

Volume 94 - Artificial Intelligence and Data Mining 2007
Edited by Kok-Leong Ong, Deakin University, Australia, Younan Li, University of Texas at Dallas, USA and Junbin Gao, Charles Sturt University, Australia. December, 2007. 978-1-920682-85-1
Contains the proceedings of the 2nd International Workshop on Integrating AI and Data Mining (AIDM 2007), Gold Coast, Australia. December 2007.

Volume 95 - Advances in Ontologies 2007
Edited by Thomas Meyer, Moraq Institute, South Africa and Acharya Nayak, Macquarie University, Australia. December, 2007. 978-1-920682-66-4
Contains the proceedings of the 3rd Australasian Ontology Workshop (AGOW 2007), Gold Coast, Queensland, Australia.

Volume 96 - Safety Critical Systems and Software 2007
Edited by Tony Cant, Defence Science and Technology Organisation, Australia. December, 2007. 978-1-920682-64-7

Volume 97 - Data Mining and Analytics 2008

Volume 98 - Koll Calling 2007
Contains the proceedings of the 7th Baltic Sea Conference on Computing Education Research.

Volume 99 - Australian Video
Edited by Haig Tan Shen and Michael Freter. October, 2008. 978-1-920682-70-5
Contains the proceedings of the 1st Australian Video Conference.

Volume 100 - Advances in Ontologies
Edited by Thomas Meyer, Moraq Institute, South Africa and Mehmet Orgun, Macquarie University, Australia. September, 2008. 978-1-920682-71-2
Contains the proceedings of the Knowledge Representation Ontology Workshop (KROW 2008), Sydney, September 2008.

Volume 101 - Computer Science 2009
Contains the proceedings of the Thirty-Second Australasian Computer Science Conference (ACSC2009), Wellington, New Zealand, January 2009.

Volume 102 - Database Technologies 2009
Edited by Xiaomin Lin, University of New South Wales and Athian Bougassaya, CSIRO. January, 2009. 978-1-920682-73-6
Contains the proceedings of the Twentieth Australasian Database Conference (ADC2009), Wellington, New Zealand, January 2009.

Volume 103 - User Interfaces 2009
Contains the proceedings of the Tenth Australasian User Interface Conference (AUIC2009), Wellington, New Zealand, January 2009.

Volume 104 - Theory of Computing 2009
Edited by Prathu Manjun, University of Ballarat and Rod Downey, Victoria University of Wellington. January, 2009. 978-1-920682-75-0

Volume 105 - Computing Education 2009
Contains the proceedings of the Eleventh Australasian Computing Education Conference (ACE2009), Wellington, New Zealand, January 2009.

Volume 106 - Conceptual Modelling 2009
Contains the proceedings of the Fifth Asia-Pacific Conference on Conceptual Modelling (APCCM2008), Wollongong, NSW, Australia, January 2008.

Volume 107 - Health Data and Knowledge Management 2009
Contains the proceedings of the Third Australasian Workshop on Health Data and Knowledge Management (HDRM 2009), Wellington, New Zealand, January 2009.

Volume 108 - Information Security 2009
Contains the proceedings of the Australasian Information Security Conference (AISC 2009), Wellington, New Zealand, January 2009.

Volume 109 - Grid Computing and e-Research 2009
Edited by Paul Roe and Wayne Kelly, QUT. January, 2009. 978-1-920682-80-4

Edited by Tony Cant, Defence Science and Technology Organisation, Australia. December, 2008. 978-1-920682-81-1
Contains the proceedings of the 13th Australian Conference on Safety Critical Systems and Software, Canberra Australia.

Volume 111 - Data Mining and Analytics 2009
Edited by Paul J. Kennedy, University of Technology, Sydney, Kok-Leong Ong, Deakin University and Peter Christen, The Australian National University. November, 2009. 978-1-920682-82-8
Contains the proceedings of the 6th Australasian Data Mining Conference (AusDM 2009), Melbourne Australia.