Programming of Interactive Systems

Introduction & Definitions

Theophanis Tsandilas
fanis@iri.fr

Interactive systems

Course objectives

Discover what interactive systems are and how they are developed

Familiarize with concepts concerning their design
- Input devices, models of interaction, interaction styles, interaction techniques, user interface widgets

Learn how to program interaction

Brief intro to methods, research & innovation in Human-Computer Interaction
Content

Intro to HCI (brief history & importance)
User-interface programming (models & toolkits), UI widgets
Interaction modeling & design
Peripherals, input devices (mouse, touch, pen-based, gestures), interaction styles
Advanced interaction techniques & special UIs (e.g., sketching, multi-modal, mobile, Web)
Intro to 2D graphics (Java 2D)
Users (perception, cognition, motor performance)
Design and prototyping methods, user evaluation

Course information

Tutorials (lab): programming exercises
Java & some Javascript

Asistants: Cédric Fleury, Arnaud Prouzeau

2 programming assignments (34%)
+ 1 exam (66%)

Course web site: https://www.iri.fr/~fanis/teaching/ISI2014/

Email Contact: [IS] in the title

Definitions

User Interface (UI)

Part of an interactive system that:
- represents its internal state on output peripherals
- captures & manages input from input peripherals

All hardware and software that allows users to control, supervise and communicate with an interactive system

Interactive system = interface + functional layer
Human-Computer Interaction (HCI)

"Human Computer Interaction is a discipline concerned with the design, evaluation and implementation of interacting computing systems for human use and with the study of major phenomena surrounding them."

ACM SIGCHI

Definitions of HCI

Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them (ACM SIGCHI, 1996)

HCI is a study of how people design, implement and use interactive computer systems and how computers affect individuals, organizations and society (Myers, Hollan, Cruz, 1996)

HCI is the study of how people interact with computing technology (Olson and Olson 2003)

Designing interactive products to support the way people communicate and interact in their everyday and working lives (Sharp, Rogers and Preece 2007)

Multidisciplinary

Interaction design

Academic disciplines
- Psychology & Cognitive science
- Ergonomics
- Sociology
- Computer Science
- Engineering

Interdisciplinary fields
- Cognitive Ergonomics
- HCI
- Information Systems
- Computer-Supported Collaborative Work
- Film Industry

Design practices
- Graphic design
- Product design
- Industrial design
- Artist
- Design
- Film Industry
**Ergonomics vs. HCI**

**Ergonomics (human factors):**

Ergonomics (or human factors) is the scientific discipline concerned with the understanding of interactions among humans and other elements of a system, and the profession that applies theory, principles, data and methods to design in order to optimize human well-being, security, and overall system performance.

- Influences:
  - mechanical engineering and physics
  - psychology
  - physiology and kinesiology
  - combined with observations and studies

International Ergonomics Association

**Ergonomics**

Traditionally, its goal is to give **precise guidelines**

**Usability**

« The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use »

(ISO 9241)

A usable system is: easy to learn, easy to memorize, efficient, visually appealing and fast to recover from errors
Utility

Reach specific needs and support real tasks

Utility vs. Usability

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>low</td>
<td>high</td>
</tr>
</tbody>
</table>

Is D better than A? What do you think?

Importance of HCI

Examples of bad design

Car park ticket payment machine at Tullamarine airport, Melbourne. For a year, the machine required a uniformed attendant to help people!
Examples of bad design

Nokia N-Gage, Mobile and Handheld gaming device (2003)
Game console + mobile phone

Tabbed dialogues with multiple layers:
- clicking tabs reorganises position and row of tabs

Overuse/misuse of tabs, rows, icons and colors
Poor categorisation
Why do we find bad designs?

Engineers and computer scientists are not (by default) good interface designers: they (we) are expert computer users, and their (our) interest is the computer or the interface.

What interests users is what the interface and the computer helps them do.

We have to design **FOR** and **WITH** users.

---

**Importance of user-centered design**

**Development cost**
- Cost of user interfaces: ~50% of total cost

**Cost of maintenance**
- 20%: « bugs »
- 80%: unpredictable user needs

**Cost of problem corrections**
- $1 during the design stage
- $10 during the development
- $100 after the delivery
Why is hard to design UIs?

« It is easy to make things hard. It is hard to make things easy »

Seems easy, common sense, but seldom done right. But once done right, it seems obvious!

User interface design is a creative process
Designers have difficulty thinking like users
- don’t understand users’ domain
- can’t « unlearn »

(from Brad Myers’ slides)

Why is hard to design UIs? (2)

Software specifications are often wrong

“Only slightly more than 30% of the code developed in application software development ever gets used as intended by end-users. The reason for this statistic may be a result of developers not understanding what their users need.”


(from Brad Myers’ slides)

Why is hard to design UIs? (3)

Software becomes more and more complex
Word 1 (100 commands) vs. Word 2007 (>2000)

Theories & guidelines are not sufficient
- too general or too specific

UI design involves many tradeoffs
- standards
- graphic-design (artistic)
- performance issues
- social factors (e.g., cost, existing practices)
- multiple platforms (e.g., hardware, browsers)
- legal issues (can’t always copy other designs)

Why is hard to design UIs? (4)
**Why is hard to program UIs?**

They are reactive
- event-based programming, difficult to modularize

They require multi-processing
- deal with user events, aborts, window refreshing, multiple devices

Need for robustness
- no crashing, unexpected input, helpful error messages, aborts, undo, recovering mechanisms

Being responsive, real-time requirements
- Tracking input events, fast output
- Video, sound, multimedia

---

**Brief history of HCI**

---

**The history of interfaces**

**Phase 1** (Interface as hardware)
- 1950s
- Engineers / programmers
- Electrical engineering

Grudin (1990) The computer reaches out: The historical continuity of interface design
The history of interfaces

Phase 2 (Interface as software)
1960s-1970s
Programmers
Punched cards, batch processing
Users (indirect)
Computer Science

Phase 3 (Interfaces as terminals)
1970s-1990s
End users (time-sharing)
Human factors, cognitive psychology, graphic design
Time sharing creates the illusion of a personal machine
User can afford to think "at the terminal"
Focus on user behaviour and productivity
Computer mediated human-human interaction (CSCW)
Messages / Shared file systems

Phase 4 (Interface as dialogue)
1980s-
Personal computers
Many end-users
More cognitive psychology, graphic design

Phase 5 (Interface as work setting)
1990s-
Widespread use of networks
Groups of end users, communities
Social psychology, anthropology, organizational studies
The history of interfaces

Phase 6 (?)

2000s-
Mobile computing
Mobile users, ad-hoc communities
Pervasive / ubiquitous computing
Domestic computing
Social computing
Anthropology, arts and drama

Influences

Computer science
  Software engineering
  Technological advances

Human factors & psychology
  Computer programming and usage
  Work environments

Cognitive science
  Models, theories, frameworks

Software Engineering

Software crisis (NATO Software Engineering Conference, 1968)
  → Software engineering

Specifications
« Waterfall » and « Spiral » dev. models
Iterative development
Prototyping

Evolution of technologies

Speed increase
  Motivated more applications

Cost decrease
  Interfaces accessible to more people

New technologies
  New challenges & interaction needs
**Human Factors/Ergonomics**

Guides for improving interface design
Guides for evaluating interfaces

First psychological studies in HCI
- Programming psychology (Software psychology ‘60s)
- Behavior of programmers (Weinberg 1971)
- Comparison of batch processing and time-sharing (Sackman et al. 1968)
- Response time and productivity
- Individual differences among programmers (Sackman 1970)
- Design principles de (Hansen 1971)

**Cognitive sciences**

Study of perception, cognitive processes such as attention, memory, and learning

Provide guidance at early stages of the software development process

**History of HCI**

*Technological visions - Vannevar Bush*

MEMEX and Hypertext (1945)
- Vannevar Bush: “As We May Think”

**Game interfaces**

Spacewar!
- MIT - Steve “Slug” Russel et al. (1961-62)
- DEC PD1 “mini-computer”
History of HCI

Technological innovations - Sketchpad

Sketchpad – PhD thesis at MIT by Ivan Sutherland (1963)
1st graphical user interface
Pointing gestures (optical pen), drawing, zooming, copy-paste

Technological innovations – Hypertext, Xanadu

Coined the term hypertext (1965)
“non-sequential writing”
“Mr. Nelson pointed out that we often do not think in linear sequences but rather in “swirls” and in footnotes. He introduced the concept of the hypertext, which would be a more flexible, more generalized, non-linear presentation of material on a particular subject.

The educational possibilities in the use of the hypertext are vast. For example, it is possible that basic texts on a subject could be interindexed, so that the necessity and difficulty of tracing footnotes and rare sources would be eliminated. In this way the problems of information retrieval because of widespread writing today would be alleviated, making decisions in many fields easier.”

Leading to the sub-discipline of hypertext and hypermedia

Technological innovations - Douglas Engelbart

Augmentation not automation: “increasing the capability of a man to approach a complex problem situation, to gain comprehension to suit his particular needs, and to derive solutions to problems”

NLS (Online system) devices and concepts:
- The mouse pointing device for on-screen selection
- A one-hand chording device for keyboard entry
- Video-conferencing, document sharing
- On-line help systems
- The concept of consistency in user interfaces

The first mouse (1963)
Textual interfaces

(1969 - 1983)
command line, menus and input screens

First text editor
WordStar (MicroPro, 1979)

First spreadsheet: Visicalc
Dan Bricklin (1979)

Apple ][ (1977)

Xerox PARC (‘70)

PARC: Palo Alto Research Center created in 1970
- Three researchers/engineers have won the Turing Award

Object-oriented programming (Smalltalk)

Laser printer, Ethernet

WIMP: Windows, Icons, Menus & Pointers

Portable computers: Dynabook (1968)
- Designed but never built

Xerox Star (1981)

$16,500

Design influenced by software needs
(based on task analysis, scenarios,
600-700 hours of video)

Native function on a network

GUI based on office/desk metaphor

Use of icons and windows and the idea of WYSIWYG ("What You See Is What You Get »)

System focusing on documents
(users do not know the applications)

...but market failure

Xerox Star (1981)

$16,500

Too innovative, powerful, different
Target market missed (e.g. no spreadsheets)
Expensive ($16,500)
Closed architecture (impossible to develop applications outside Xerox)
Political reluctance to expand market beyond printers

...but it has greatly influenced future systems
Apple Macintosh (1984)

- Commercial success, more mature and a more open public
- Aggressive price ($2,500) accessible to larger public
- Menu bar, modal dialog boxes and visible applications inherited from Apple
- UI toolkit to help external developers
- Detailed style guides to help consistency between apps
- Three key applications: Finder, MacPaint, MacWrite

X Windows (1984)

- Athena project of MIT: connect 4000 UNIX machines, from different sponsors (DEC, IBM, Motorola, etc.)
- Client/server model:
  - division of what/how to facilitate portability
  - transparent use of network that permits remote displaying

MS Windows (1985)

- Moved to overlapping windows

Microsoft Windows 1

Microsoft Windows 2 (1987)
Desktop interface (1984 - )

More power and new uses (network), but little change in interaction: WIMP (Window, Icon, Menu & Pointing)

... this is not necessarily a bad thing!

HCI

Does not follow Moor’s law

<table>
<thead>
<tr>
<th></th>
<th>Original Macintosh</th>
<th>iMac 20”</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong> 68000 - 0.7 MIPS</td>
<td>x3000</td>
<td>CPU G5 - 2250 MIPS</td>
</tr>
<tr>
<td><strong>RAM</strong> 128kB</td>
<td>x2000</td>
<td>RAM 256MB</td>
</tr>
<tr>
<td><strong>Floppy</strong> 400kB</td>
<td>x200000</td>
<td>HD 80GB</td>
</tr>
<tr>
<td><strong>9” b&amp;w 512x342</strong></td>
<td>x2 / x10</td>
<td>20” colors, 1680x1050</td>
</tr>
<tr>
<td>keyboard, mouse</td>
<td>idem</td>
<td>keyboard, mouse</td>
</tr>
<tr>
<td>WIMP desktop</td>
<td>idem</td>
<td>WIMP desktop</td>
</tr>
</tbody>
</table>

...WILD in LRI

screen wall:
- 5.5m x 1.8m
- 20480 x 6400 = 131 million pixels
- 32 screens of 30”, in a 8x4 configuration
- driven by a cluster of 18 PC,
- linked by a high speed network

3D motion capture system that tracks users in real time
Multi-touch interactive tables

INRIA, LRI, Université Paris-Sud – The Wild project
But also: Disappearing devices

(Ni and Baudisch, 2009)

HCI and research

Most innovations come from research labs (academic or industrial)

It takes time to reach commercial products

Example 1: Touch displays

Touch displays
IBM (1971)

MultiTouch
U of Toronto (1982)

Gesture research
Bell Labs (1983)

Mainstream
Apple’s iPhone, iPad, iTouch
Microsoft Surface

Example 2: Wearable computing

Evolution of Steve Mann’s “wearable computer” invention

Steve Mann today...

Google Glasses
**virtual reality**

user immersion (sensor + motor)
input? usually body tracking or speech

e.g., a head mounted display and a cave

**DigitalDesk (Wellner, 1991-93)**

**Augmented paper**

physical « Toolglass » (Mackay, 2002)
Projection on paper with pocket-projectors (Song, 2010)

**Tangible interfaces**

Reactable: Tangible music interface

Dynamic Shape Display (MIT)

http://tangible.media.mit.edu/project/inform/
Foldable interfaces

Vertegaal et al., Queen’s University, Canada

Brain interfaces

HCI in popular culture


http://interaction.lille.inria.fr/~roussel/digital-library/media/2002-minority-report-clip.mov
http://vimeo.com/22292199