IDI - Introduction to Usability and UI Design

Contents

1 Introduction

Through the last decades, there has been a constant increase of the presence of computers in our lives. Every day we find more and more situations in which we have to interact with the computer system: a mobile phone, a DVD recorder, and so on. All those systems have an interface to communicate with users. This interface is called human computer interface or sometimes simply user interface.

The objective of these course notes is not to provide the full contents of usability field or human computer interaction, but to introduce the students to the main aspects that should be taken into account when developing graphical user interfaces. In this course, we will talk about usability, we will visit some design tips, and provide some high-level rules for user interface design. Moreover, we will present a very important tool on usability evaluation, usability testing.

1.1 Human computer interaction

Human Computer Interaction is a very relevant issue when evaluating the quality of an application. An application has to fulfil its requirements, but it also has to provide an easy access to its features. When an application is difficult to use, it is perceived as a low-quality application. The user interface consists of all the tools and methods that are used to communicate between the user and the system. When designing the user interface, we have to consider the communication in both ways, from the user to the machine, and from the system to the user.

Human Computer Interaction (or HCI) is a field that deals with the study of how humans interact with machines. This interdisciplinary science began by combining the data gathering methods and intellectual framework of experimental psychology with a powerful and widely used tools developed from computer science. And then, contributions accrued from educational and industrial psychologists, instructional and graphic designers, technical writers, experts in human factors and ergonomics, information architects, and so on. Of course, there is no universal way of communicating with a system. It will depend on the application, hardware, etc. Some applications will require the use of buttons, such as in the case of a watch, while other applications will require the use of a complete keyboard, such as in the case of a computer.

Two important features of HCI must be taken into account:

- HCI is about understanding and critically evaluating the interactive technologies people use and experience
- HCI is about understanding contemporary human practices and aspirations

Usability is a field that deals with the study of conditions that make a certain program accessible, comprehensible, and easy to use. In many cases, designers only worry on the correctness of the code, and ease-of-use is completely left apart.

The process of interface design, and more concretely, the design of graphical interfaces must fulfil a number of conditions in order to make the applications easy to use and

accessible to users. In general, there are some requirements to those graphical interfaces must guarantee:

- Its appearance must be nice and comprehensible; they must provide the user with a feeling of controlling the application. Users must see how to achieve the desired goals and how to make their work.
- User interfaces must be transparent to the implementation of the application. User actions should be undoable and the work must be often saved automatically.
- Applications should do the maximum possible number of tasks with the minimum inputs from the user.

As we will see, it is not an easy task to guarantee those requirements.

The model Human Processor was an early cognitive engineering model intended to help developers apply principles from cognitive psychology. As we will see later, in this model (see Figure 1), there are many aspects that are commonly used in different human computer interaction fields in order to analyse how humans interact with computers. These aspects cover elements like our memory capacity, our visual perception, the speed at which we are able to interact, and so on and so forth.

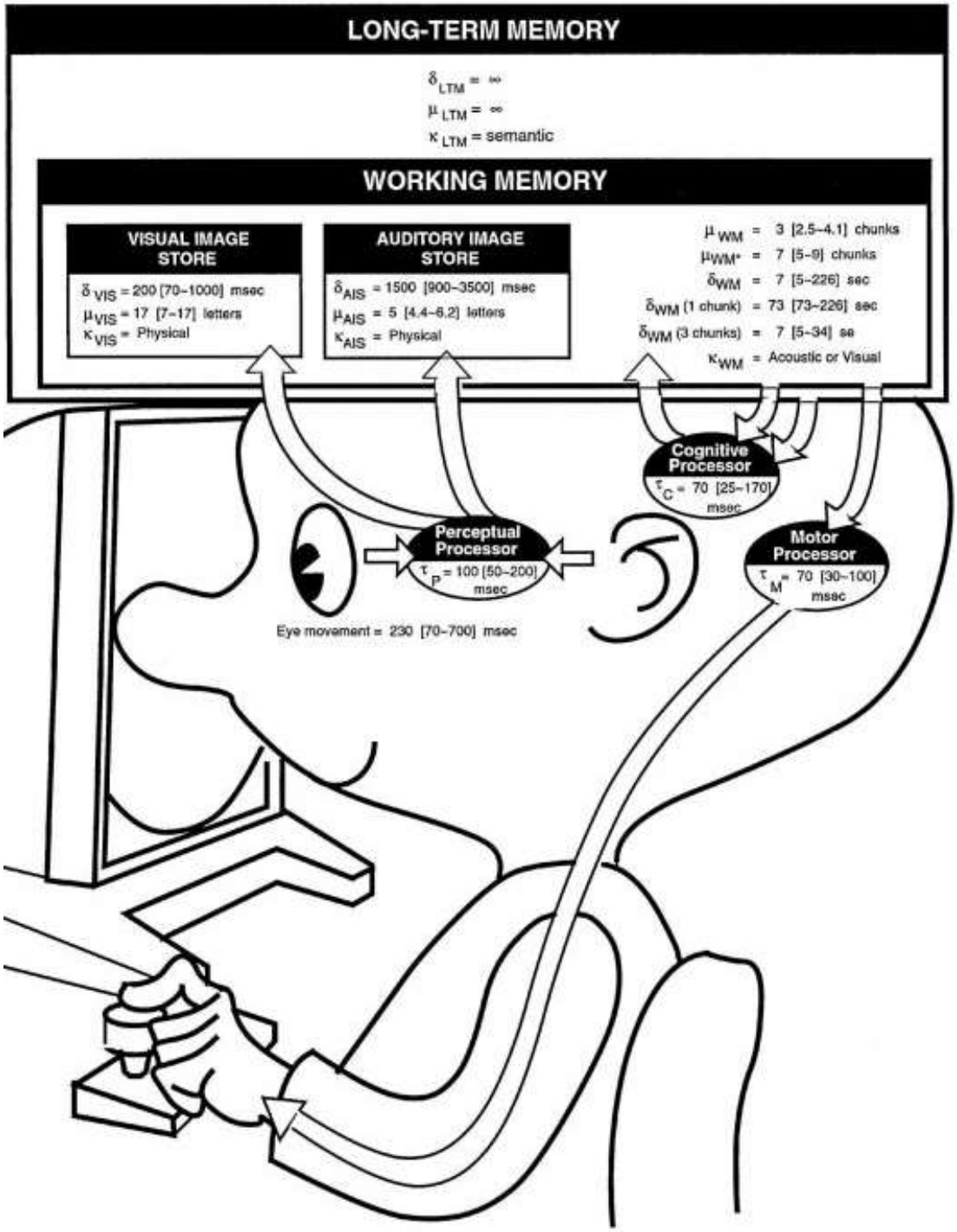


Figure 1: The Model Human Processor.

One of the original focus of HCI was usability.

The original definition of usability was stated as "easy to learn, easy to use". However, it then grew to cover many different aspects that we will see along the course. Its incorporation to software design is closely related to the software crisis on the 70s. Then, it was included in several areas, not solely restricted to computer science.

There are several taxonomies that try to separate and properly classify the different fields that relate to human computer interaction, from a computer science perspective.

However, the different classifications may be better understood on how we put right center of our interest. For instance, in Figure 2 we can see a classification of several areas that include human computer interaction (although some researchers would find that HCI covers some of them) from the perspective of how these fields collaborate to create the experience that is finally received by the user.

Figure 2: The different fields that compose the user experience.

A different infographics in Figure 3 depicts a relatively different picture. In this case, separating some fundamental aspects of the design: Visual Design and Interaction Design.

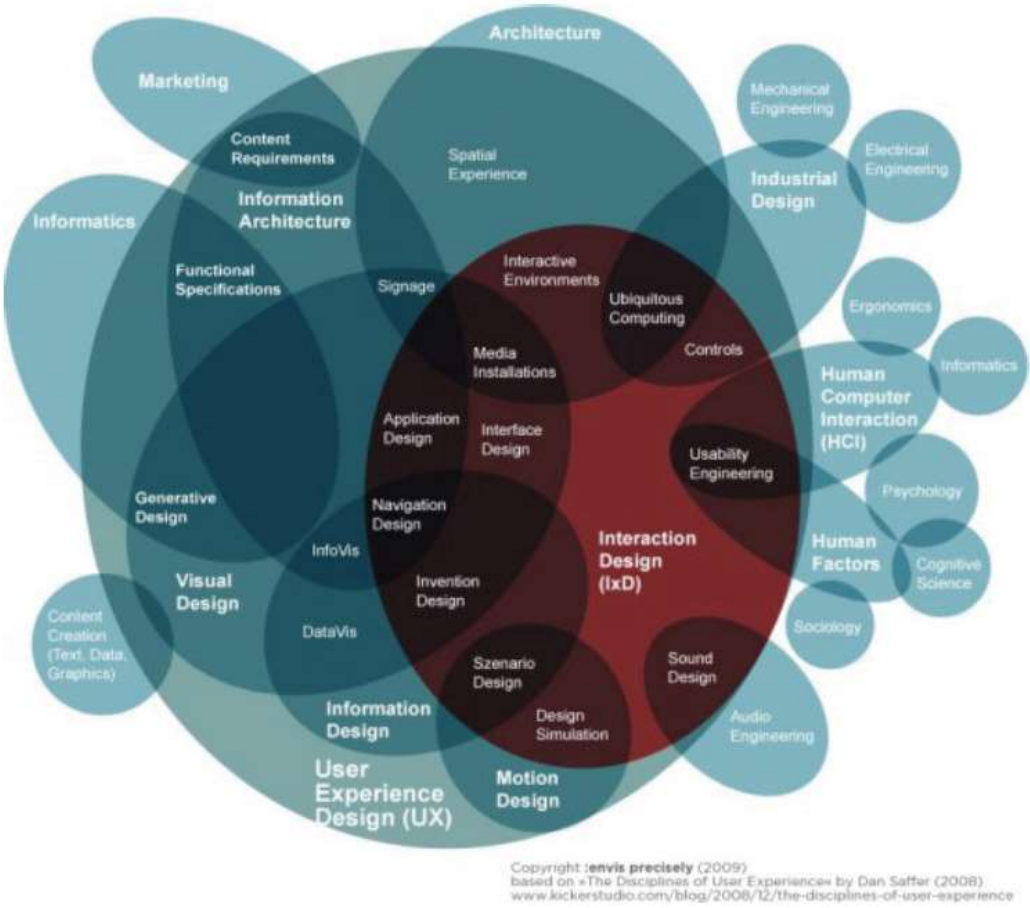


Figure 3: Interaction design, visual design and other fields that create the user experience.

For a better introduction, you should read the chapter 2 of the Interaction Design book (https://www.interaction-design.org/books/hci.html) until section 2.6, included.

1.2 User experience

We have mentioned the concept "User experience" before. Its definition is not easy, and it is the result of a set of elements in conjunction. Peter Morville describes it as the sum of several elements: utility, desirability, usability, value… in the infographic shown in Figure 4. We also saw before that there is a big bunch of elements that conform or influence the experience the user will receive from using a certain application or device. To put it simply, these three definitions would roughly explain what we can understand by user experience:

- "Experience or User Experience is not about technology, industrial design, or interfaces. It is about creating a meaningful experience through a device."
- …"the perception left in someone's mind following a series of interactions between people, devices, and events".
- What you **remember** and **feel** from/after the use of a device or application.

Figure 4: User experience as the result of the collaboration of several fields.

You should see the four videos appearing on the page that shows up when clicking into chapter 3 of the Interaction Design book linked above, although it is interesting to read all the chapter too.

1.3 Interaction Design

Finally, a closely related concept is Interaction Design. It can be defined as the field that deals on how we design the steps and processes the user will follow to interact with an application and device. The decision on which steps to make and how they are implemented is a key issue on how the user will perceive the ease (or lack of) of use of an application or device. For instance, in Figure 5 we can see a design decision made by LG to facilitate accessing the voice rockers in LG G2 smartphone. Since the size of smartwatches does increase more and more, while the hands of users do not, it is a common problem to properly grip the phone and, at the same time, be able to reach the volume buttons that are commonly placed at the other side. With this as the problem to solve in mind, the designers at LG decided to place the buttons in a position that is closer to the resting fingers' position while holding the device. This makes them more accessible. This, together with the fact that they have some shape, makes their use simple.

Figure 5: Volume buttons placed at the back face of a LG smartphone for easier access.

An interesting definition of Interaction Design is: "Interaction design is about shaping digital things for people's use".

For a better understanding, you should read chapter 1 from the Interaction design book commented above, up to section 1.3, included.

1.4 Usability

Usability is defined in ISO 9241 standard as **the ability in which a product may be used by specific users in order to carry out specific tasks effectively, efficiently, and with satisfaction in a specific use environment**. Three important concepts appear here:

- *Efficacy* is the ability of correctly and completely achieving a certain goal.
- *Efficiency* is the relation of used resources and the completeness and correctness of achieved goals.
- *Satisfaction* is the comfort and acceptation of a system by the users and other people that are affected by its use.

Note that in the previous definition, usability is always referred to a *concrete* user group and a *concrete* user application. Moreover, it is important to note that usability also implies a larger group than the ones directly affected by the product itself. For instance, if a hotel reservation application is prone to errors, those problems may affect to the people being attended by the professional that uses the application maybe by an error in the conditions of the hotel reservation. Buggy or slow interfaces may also increment the time people require to obtain a flight or a trip reservation.

1.5 User interface design

User interfaces are becoming more and more important in economy. They may make the difference between a universally appreciated operating system, such as Apple's iOS (in Figure 6 you can see a snapshot of one of its modern versions), and a more and more criticized one, such as RIM's Blackberry operating system (see Figure 6 – center), or the old Nokia Symbian. In the first case, iOS market share has been growing since iPhone was

released. BlackBerry's popularity, on the other hand, is slowly declining, despite its advantages over iOS on certain aspects such as privacy, data encryption, and so on.

The case of Apple's operating system for mobile devices is curious in several ways. For instance, the first version did not provide any new features compared to what was present in previous smartphones. For instance, there was no copy-and-paste utility (present in even quite old Windows Mobile). However, iOS was far more usable than any other previous operating systems for mobile devices, partly due to the use of capacitive screens that allowed the user to interact using fingers, instead of resistive ones that required stylus. This, together with the high popularity of Apple products, made the iPhone to become a game changer.

The history of Blackberry is the other way around, since the iPhone was launched, its market share has done nothing but decline, and the new operative system, BB10 (see Figure 6 – right), which can now face the OSs by Apple and Android was launched maybe to late to make it turn around. Many of its users moved to Apple and Android. The last years, most of the users leaving Blackberry in the professional market, were substituting them for Samsung's devices that sport Knox platform. Knox security solutions facilitate the management of Android phones by IT managers in companies. Previously to Knox, there were other suites, but none was so popular security. Knox has received the certification of different US departments, something that was only previously achieved by Blackberry.

Figure 6: The iOS 7.0 operative system by Apple vs Blackberry 7.0 (2011) and the new Blackberry OS (BB10, 2013).

From a user's perspective, user interfaces may change how people relate with computers. They may save time, they may improve the quality of our work, and they may improve our efficiency when carrying out certain tasks. On the other side, when the user interfaces are not properly designed, they may generate frustration, fear, and provoke mistakes.

1.5.1 Desktop Systems

As we will see later, there is no common *universal* user interface for *all* applications. Its suitability obviously changes according to a number of factors such as the task the user has to achieve, the size of the screen, the interaction mode (i. e. pointer-based, gesture-based…), the availability of a physical keyboard, and so on.

Typically, we will have plenty of room (see for instance Figure 7), so a large number of elements can be presented without cluttering the interface. Moreover, we expect from an OS interface that a lot of possibilities will be available, and the medium user will have a high number of hours of interface *exposure*, therefore, the degree of complexity of the interface can be relatively high. Another important issue with desktop-based UIs is the fact that a physical keyboard is (almost) always available. Thus, the whole screen can be devoted to presenting information.

Figure 7: Mac OSX Lion.

1.5.2 Mobile phones

Compared to those, the interfaces of mobile phones (shown in Figures 8) must take advantage of the small screen available. In some cases, such as iOS devices or Android-based, there is no physical keyboard. This has two main implications: First, we will have to devote some room in our screen to the keyboard. Second, raising the virtual keyboard should not be intrusive, or, at least, intrusion should be limited at most as possible. Sometimes this issue is not achieved at all. Mobile phones also have a feature not present in most devices: they can receive phone calls (or SMS). This may interrupt the *work* being done by the user. Again, these notifications should not be intrusive, and should not make the user to lose work he has been carrying out.

Figure 8: Android Lollipop (5.0, 2014) and Windows Phone (8.1, 2014)

1.5.3 Tablets

Tablets or pads are a new form factor. Although they have existed for several years in the form of tablet PCs with tactile screens, Apple's iPad and its highly optimized OS for mobile devices has been a game changer as it was for the phone industry (see Figure 9). Some people think tablets may be a replacement for cheap, light Netbooks. With the arrival of more tables with improved keyboards (e. g. Asus Transformer or Microsoft Surface), this seemingly less predictable change has started. However, the percentage of users that effectively replace a portable computer by a tablet is still low.

The user interfaces of tablets range from adapted mobile OSs, such as iOS and some versions of Android, to fully adapted interfaces, such as Android, or Windows RT (and nowadays even full windows port to Intel-based tablets such as Surface Pro), see for instance Figure 10.

A main feature with tablet-based operating systems is the presence (or absence) of real multitasking. If a tablet has to substitute a netbook or a laptop, the user will expect multiple windows and processes. This is something some tablet-based Operating Systems do not properly handle (such as iOS) at the time of this writing.

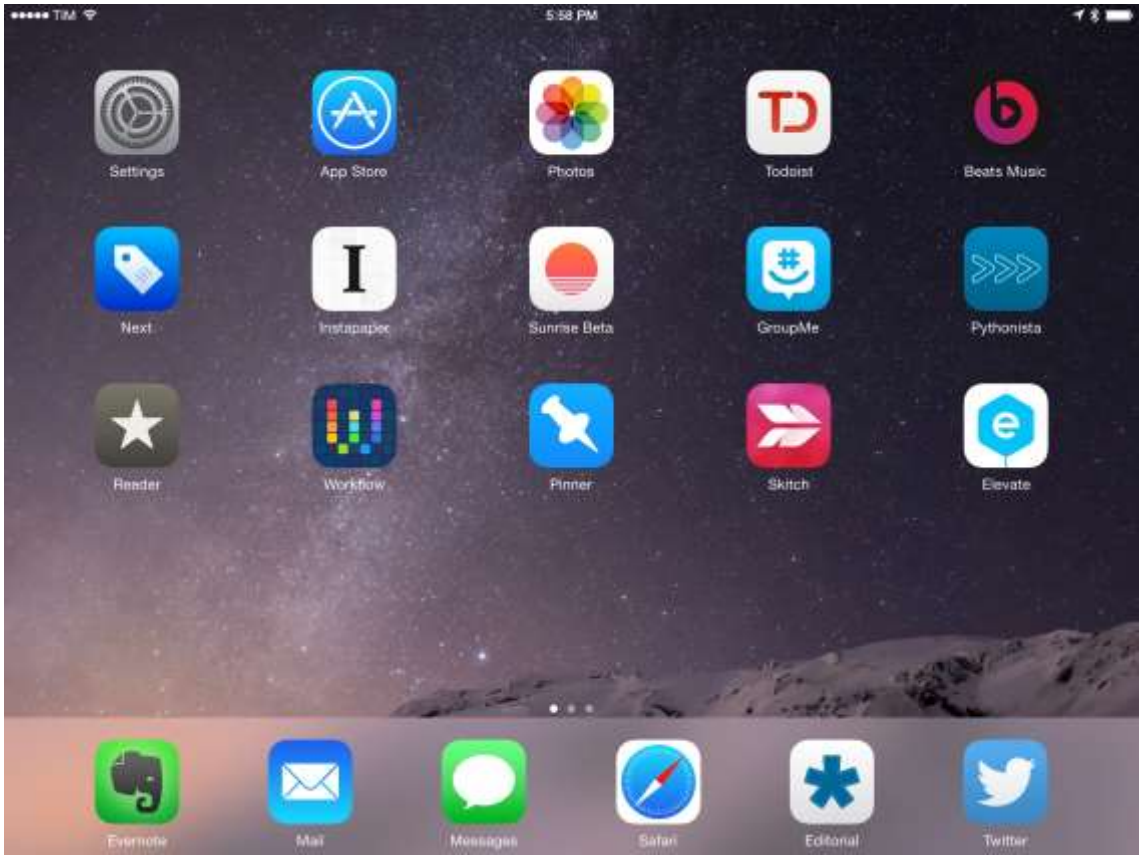


Figure 9: iOS 8 for iPad.

Figure 10: Windows RT 8.1.

2 User Interfaces

2.1 The evolution of human-computer interaction

We may classify person computer interaction in five stages:

- Years 50 – 60: Computers work using punched cards and in batch mode. There is not actual user-computer interaction in the sense we understand it now.
- Years 60 – 80: Through these two decades, computation was carried by shared time systems. The dominant interaction method was command line.
- Years 80 – middle 90s: Reign of the graphical interfaces. Although they were created at the beginning of seventies, its generalization came later. The main interaction metaphor is the desktop and the interaction was done through the use of a mouse pointer.
- Middle 90s – mid 2000s: New interfaces that use gestures and voice recognition. We could call these *advanced interfaces*, although the desktop metaphor relies the dominant in our interaction with computers.
- Mid 2000 – actuality: Tactile screens start is kingdom hand to hand with our almost ubiquitous use of mobile devices.

Although there are some interaction metaphors that would require a different set of rules, such as 3D user interfaces, most interfaces share desktop-based UI elements and its interaction does not differ much from the well-known Desktop + Mouse interaction (clickable elements appear both in our PCs and in our mobile phones). Therefore, we will mainly concentrate on elements that appear in applications in the form of menus, buttons, and textual elements.

2.2 A bit of history

One of the relevant moments in User Interface Design was the creation of the Apple Macintosh computer in 1984. It started the revolution in User Interface design, since it was the first commercial system that used the Desktop metaphor. There was actually a previous project in Xerox's Palo Alto Research Centre that already had defined the desktop metaphor, and actually, many people say it was the inspiration to the Macintosh system.

Figure 11: A snapshot of the famous 1984 Apple advertisement.

They prepared a very smart and intelligent presentation: They took advantage of the year to make an awarded advertisement that was first aired during the Super Bowl (one of the most prominent advertisement moments in a year), and presented the Mac as the disruption against to date dominant Microsoft system *à la* Big Brother. In Figure 11 we may see a snapshot of the advertisement.

Other achievements that helped this revolution were the improvements in printing systems, and the WYSIWYG (*What You See Is What You Get*) paradigm. This allowed the possibility of higher quality printing for a broad audience. The Macintosh introduced the mouse as a standard input system. Opening folders and documents was achieved using double click. And windows and menus could also be organized using the mouse.

Following the desktop metaphor, the rest of the objects of the interface are designed in coherence with it. One of the main objectives of Steve Jobs and Steve Wozniak was to make the computer available to normal people. Therefore, everything is though as to be in a familiar and friendly ambient. The system uses words such as *documents* instead of *files*. These objects can be thrown to *trash* to erase them, and even the dustbin remains full until it is emptied. The Macintosh was the first computer with a graphic interface that was a commercial success.

Joseph Licklider presented an interesting work, *Man-Computer Symbiosis* in 1960. It was the first proposal to improve the interaction with computers. In this work, he proposes a new interactive method to use computers. Two years later, he is hired by DARPA with the goal of finding new uses for computers. Up to that moment, they were only used for calculation problems, mainly in military uses. One of the potential military uses that were envisioned was the decision support in short times. Many projects were financed with the goal of improving interaction. One of the laboratories that were supported was MIT Lincoln Laboratory, where Ivan Sutherland presented his PhD thesis in 1963 entitled *Sketchpad: A Man-Machine Graphical Communication System*. Sketchpad was a system for

vector illustrations that was used with a pencil directly on to the screen. This system is the one that first proposes the *window* concept. It represented the beginning of Graphical User Interfaces. Curiously enough, Sutherland's thesis supervisor was Claude Shannon, father of information theory.

The same year, 1963, in Stanford's Augmentation Research Centre the mouse is invented. It is a result from a more general work by Doug Engelbart with the objective to improve human intellect.

A third important actor at this moment was Xerox, Inc. Xerox grow with the commercial success of copy machines since the beginning of sixties. In 1970 the company creates the Xerox Palo Alto Research Centre (PARC). Its main goal is exploring new opportunities to apply computers to office work. One of the multiple inventions by Xerox PARC was the laser printer. Actually, in 1981, they already had a computer with a graphical interface that used the notion of a desktop. They called this metaphor the *physical desktop metaphor*. Some rumours say that Macintosh was inspired on those projects in a visit made around 1979 by the Apple founders.

Finally, another achievement that paves the new way of graphical user interfaces and shows the potential of computers is the Aspen Movie Map. It was an application created by the MIT Architecture Machine Group. It allowed the user to virtually visit Aspen through the use of multiple images stored in a videodisc. The user was able to move in any direction. It was even possible to change the season of the year at any moment in the virtual visit session.

2.3 Graphical interface programming

Today there are a number of tools for the development of graphical interfaces. They come in different flavours: from tools that are platform independent to window system-dependent tools. We may for instance work using Visual C++ and Microsoft Foundation Classes (MFC) in windows, or we can use GTK+ or Qt if we want to be platform-independent.

2.3.1 Designing and programming the User Interface

The use of graphical interfaces means an important change in software architecture. Traditional programs have a linear structure, that is, users execute one instruction after another in a sequential manner. On the other hand, when using a graphical interface in our applications, the interaction is produced through the use of asynchronous events that are generated during the user interaction. Such kinds of programs are usually called *event-oriented*. Window resizing, mouse clicking, or key typing are usual actions that generate events. The tools for graphics interface design must provide APIs for the management of interaction-related events.

As a consequence, when designing an application with a graphical user interface, the process involves the following steps:

- First, a preliminary design of the user interface: The elements of the interface are chosen and arranged, and the main look-and-feel of the interface is determined.
- Secondly, the interactions with the elements of the interface are implemented.

For general applications, those two steps are looped many times: First, an initial design is created, and we add the main application widgets. Then, the connections between the elements are incorporated. From this moment, we increase the features and thus, the interface is modified and so is the implementation of the new features. From time to time, this requires the addition of a new view or dialog.

2.3.2 Programming mobile devices. Native vs web apps

When implementing a mobile application we have typically to make several choices. The first one is the kind of application to develop: web app or native.

Going native usually will lead to a more efficient app and will have some other advantages, but it has its shortcomings too, for instance when dealing with updates. Web applications usually have a lower level access to the device's sensors, but are easier to deploy.

The main reasons to use web apps instead of native apps are:

- Develop once & deploy everywhere: We may find capable browsers in many systems, and therefore it may become quite easy to develop compatible or quasi-compatible code that can be used across platforms.
- Easy updating: Since the app is loaded every time the browser connects to the page, the only code that really needs to be updated is the server part. As a consequence, no device-by-device updating is necessary and the different versions may have higher maintainability. Besides, the updates are immediate to all the users, no complex delayed rolling out is required.
- Most of the techniques that are used to develop this kind of applications, although extensive, they are well-known (PHP, Java…). Thus, specialized profiles are not required.

Despite that, the generalization of techniques and tools come with some shortcomings that, depending on the features required for the application, can be more important or not:

- Limited User Interfaces: There is a more limited number of elements (e. g. widgets) that can be used for the UI design.
- Communication protocols are limited, and often less secure.
- Limited access to local resources: Mobile devices have a huge bunch of sensors (GPS, compass…) or other resources (camera, local storage…) that are more difficult if not impossible to access from a web app. If those are crucial for your app, you might consider turning to native development.
- Limited or difficult I/O: Commonly, several technologies may be combined to solve this, although not in a simple way.
- Mainly designed for large displays with mouse: Although this is less and less the case, we may still find elements that are not as usable as in desktop devices.

The alternative to web apps is the development of a fully native application. The main advantages are:

- Richer UI: Since the proprietary SDKs provide access to the different elements, we may commonly be able to create richer UIs.

- Many controls: The OSs developers offer a large set of controls that make the development of UIs flexible and with a lot of possibilities. Moreover, such controls are rendered in the most efficient way because they are part of the UI toolkit library which may be hardware accelerated when possible.

- Safe and fast access to local resources: The many resources and sensors present in mobile devices can be accessed quite simply. Moreover, the application development environment ensures the access is secure, and may be managed (at least from the point of view of the permission granting) by the final users. This includes both sensors such as GPS or compass, to local resources such as the camera, files, and the use of communication protocols of any kind.

- Slower variety in languages and tools: Each OSs provides its own SDK which is commonly a set of tools to program in a single language. Therefore, inside the same OS, the variety of languages is smaller.

- Designed for small screens and touch controls: The widgets and the rest of elements of the OS libraries are designed for and optimized for its use with mobile, which includes the response to the different touch events to the management of other aspects such as memory consumption and so on.

The use of native apps has some disadvantages too:

- No universal access: Each OS has a different app format and development environment.

- Difficult to manage updates: Commonly, the developer updates the product up to one of the distribution systems (iTunes, Google Play…) and then each user must manage the updates per device.

- Less general than desktop programming: Developing for mobile requires a set of techniques and restrictions. Therefore, developers need a specialized training. On the other hand, there is a lot of new material on the web that may facilitate this process.

Application installation is quite easy. For Android devices, it can be done through a web page, e-mail, USB connection, Dropbox, and so on. For Apple devices, it is not so simple but some problems let you install in individual iPhones or iPads. Finally, the deployment is achieved through the publication of the applications in the different distribution platforms: Google Play, iTunes, Amazon App Store...

2.3.3 Programming mobile devices. Tools

There are some requirements for programming native apps that go beyond the language or libraries. For instance, Android development requires a Java installation, which is commonly available throughout a large set of OSs (Linux, Windows, Apple...) but the development for iOS (Apple products) requires the use of a Mac computer since the SDK is only available in this platform.

Features of Native Tools for Mobile:

- Provide the most efficient, compliant code
  - Higher level of control over GUI
  - Steeper learning curve
- Some tools are widely available and programming languages common:
  - Android SDK: eclipse + plugins is available across platforms (Linux, Windows, Mac).
    - Programming is in Java
- Some are less common:
  - iOS SDK: XCode + iOS development toolkit, only available for Mac computers
    - Programming is in ObjectiveC, also a not quite popular language
  - Windows Phone: Windows Phone SDK + XAML, only available in Windows
    - Programming is in C#, also not so popular

Cross-platform programming for Mobile. Different possibilities:

- PhoneGap (Cordova): Web-based development (HTML5). The result is a web app.
  - The result is a web app for Amazon Fire OS, Android, Blackberry, Firefox OS, iOS, Ubuntu, Windows Phone, Windows 8, or Tizen.
  - Development can be carried out in Mac, Windows or Linux
  - Limited UI, as it is a web app
- Xamarin: Development in C#
  - IDE for Mac and Windows
  - No abstraction of the GUI (different development for each platform)
  - As of Nov. 2014 is free for students
- Appcelerator Titanium: Development in JavaScript
  - Abstraction of GUI: facilitates one development, multiple deployment

Other tools for Mobile development exist, such as AppInventor, which tries to bring the ease of programming of Scratch to mobile. Its philosophy is block-based programming and started as a Google project that was taken over by MIT.

One of the good things of today's mobile programming is the large set of tools available from the SDK developers, which makes the development process not as long as was several years ago. This happens not only for mobile, but also for PC. The core question is that mobile development comes with its own odds and the presence of so many learning materials and libraries makes the process softer.

2 Usability requirements

2.1 Introduction

These course notes will not focus on a concrete application type. Therefore, we will study general principles and guidelines for user interface design.

Effective interfaces generate positive feelings. The users must feel they control the system, that the information is clear, and that they know what to expect after each of their actions. They must provide the tools to make the users 'goals to be accomplished without pain. The U.S. Military Standard for Human Engineering Design Criteria stated back in 1999 these purposes:

- Achieve required performance by operator, control, and maintenance personnel.
- Minimize skill and personnel requirements and training time.
- Achieve required reliability of personnel-equipment/software combinations.
- Foster design standardization within and among systems.

Setting those goals is a difficult task. Moreover, they may vary across cultures. User interfaces must adapt both to the application and to users. In order to achieve those goals, we must plan carefully, be very sensitive to the users needs, analyse thoroughly the requirements, and perform accurate testing.

2.2 Requirements analysis

Requirements analysis is one of the central problems in user interface design. It may be divided into four different subparts:

- Task and subtask identification
- Reliability ensuring
- Standardization and portability
- Schedule fulfilling

Understanding the tasks and subtasks that must be carried out by the application may be a complex issue. Exceptional tasks, such as the ones for emergency conditions, are often difficult to identify.

Once the tasks have been identified and analysed, we have to ensure proper reliability, that is, actions should perform as expected, data display must be coherent with internal data (i.e. files or databases) contents, and updates must be applied correctly.

Striving for consistency produces great benefits by error avoidance and softer learning curves. Moreover, the accomplishment of standards and working towards a portable and consistent product may facilitate updates or software version changes, and reduce the number of annoying or dangerous errors.

Finally, like any other projects, interface design should stick to deadlines in order to avoid delaying other parts of the main project.

2.3 Usability profiles

When talking about usability of a user interface, we do not refer to a common design as valid for all the users. It is not the same to develop a user interface for the control panel in the cockpit of an airplane, or a set of buttons and/or screens for the control of a music player. The inherent complexity of the first task will imply a complex interface, while the second task may have a lower number of features and thus, less buttons, displays, and other controls.

A common mistake is to label a tool, for instance a smartphone as *usable* because it may be used by a large number of people. There may be a group of users, maybe advanced ones, which find the interface is too dull or simple, and does not allow performing certain tasks manually.

User interfaces may be classified upon the service they are focused to provide. We may distinguish five different uses or systems:

1. Life-critical systems
2. Industrial and commercial
3. Home and entertainment
4. Exploratory applications
5. Collaborative systems

2.3.1 Life-critical systems

Life-critical systems include those for controlling air traffic, power plants, military operations, and so on. It is common that managing those systems implies a high degree of complexity. For instance, the control panel of a Boing 787 Dreamliner, shown in Figure 12 contains a large number of buttons and displays, since a lot of parameters are expected to change and to be monitored during the flight.

Figure 12: Boing 787 Dreamliner cockpit control panel.

Complex problems may have complex solutions. As a consequence, in life-critical systems, complex interfaces may be unavoidable. Users require lengthy training periods to obtain rapid, error-free responses under highly stress situations. On the other hand, such systems are expensive, and, more importantly, high costs are expected. As a consequence, the expense is affordable. Usually, the users are well-motivated professionals and thus, subjective satisfaction is less important than in other systems.

In Figure 13 we can see a snapshot of the control room of Fukushima 1 nuclear power reactor. Like in the case of the airplane, a high number of parameters must be monitored and controlled the whole time, and proper warning displays must be designed.

Figure 13: Fukushima nuclear plant control room. Image by Kawamoto Takuo.

2.3.2 Industrial and commercial

Industrial and commercial software include the ones used in banking, insurance, and airline or hotel reservations management. Typically, small budget for user training is available. Thus, systems should be easy to learn. Retention is obtained by frequent use. Speed may be one of the central features of these applications because a high volume of

transactions will be carried out. Key accelerators may be an important feature. Commonly, forms will have to be filled, so the designer has to take care on readability, the order of control change upon "Tab" key clicks, and so on. Interfaces should avoid operator fatigue or stress. In some cases, the designers should focus on internationalization, since many companies may have offices around the world.

In Figure 14 we can see a form filling application of an insurance program. Note the high number of fields to visit. Guaranteeing proper navigation through controls, i. e. in the reading direction may ease its use and improve speed.

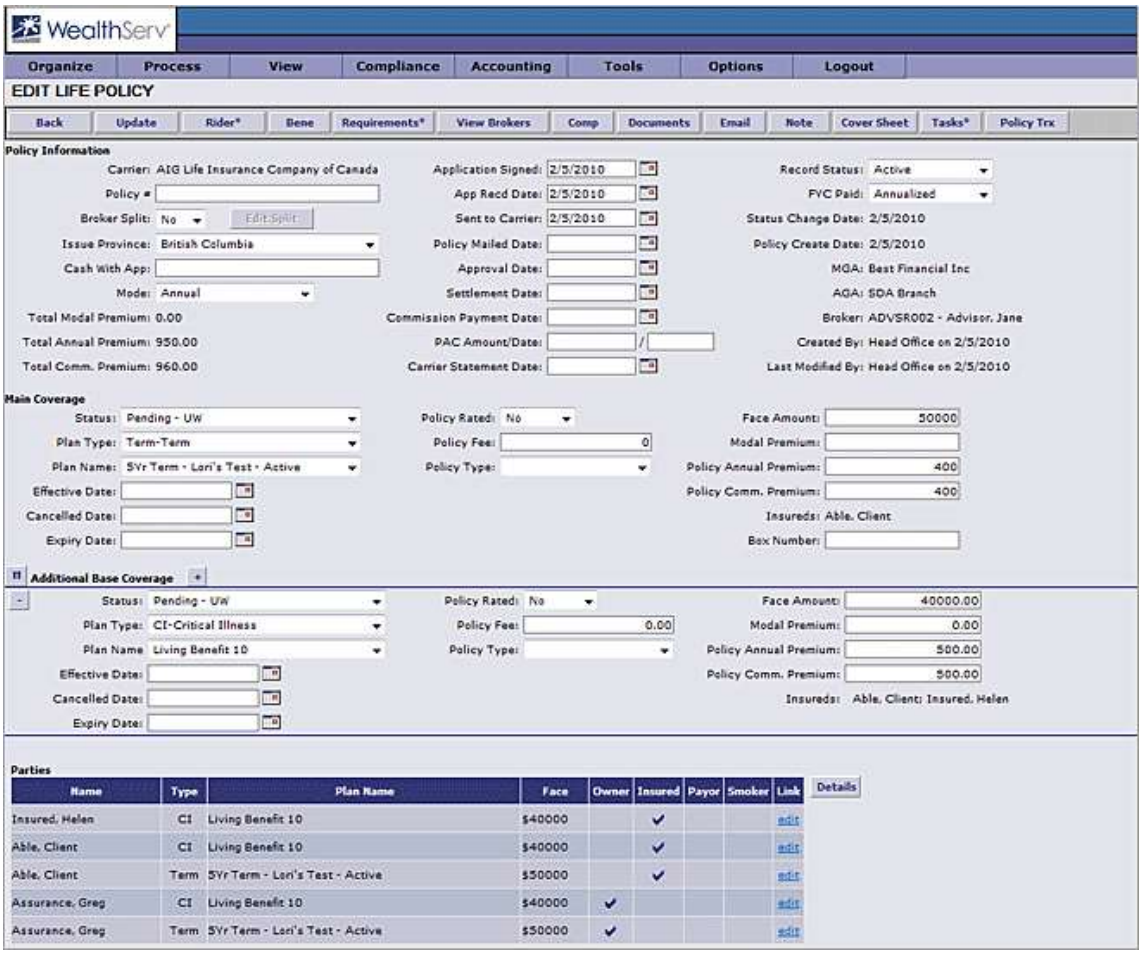


Figure 14: WealthServ insurance software screen shot.

2.3.3 Home and entertainment

Compared to the previous ones, these applications are usually addressed to a broader audience. Some texts include office software into the same group than home and entertainment, although it could be also included in the previous group because the type of user may be different.

When we are at home, we usually want no complications. User interfaces should be easy to learn, because the universe of possible users varies a lot: from 8 or 10 year-old children, to elder people. Dealing with a DVD player, a music player or a TV must be an easy task. In some cases, menus are not properly labelled or their commands are not described properly. We may see an example in Figure 15. Two of the menu options: 'Program install'

and 'Configuration' are a bit confusing. In many TV menu interfaces, the 'Configuration' option would lead us to adding new channels. Moreover, having a 'Configuration' option inside a 'Setup' menu does also fail to provide information on the options that will be possible if we select the option.

Home and entertainment interfaces should provide high rates of subjective satisfaction because their use is frequent, and there is a high level of competition in the market. It is one of the cases where all sorts of users (novice to power users) should be supported. Examples of those interfaces that scale on the experience of the users are search engines: Usually, they have a simple version where we write our search query in a simple text box. For advanced users, they provide an interface where more complex queries (such as allowing limiting domains, providing a date of publication, and so on) are possible.

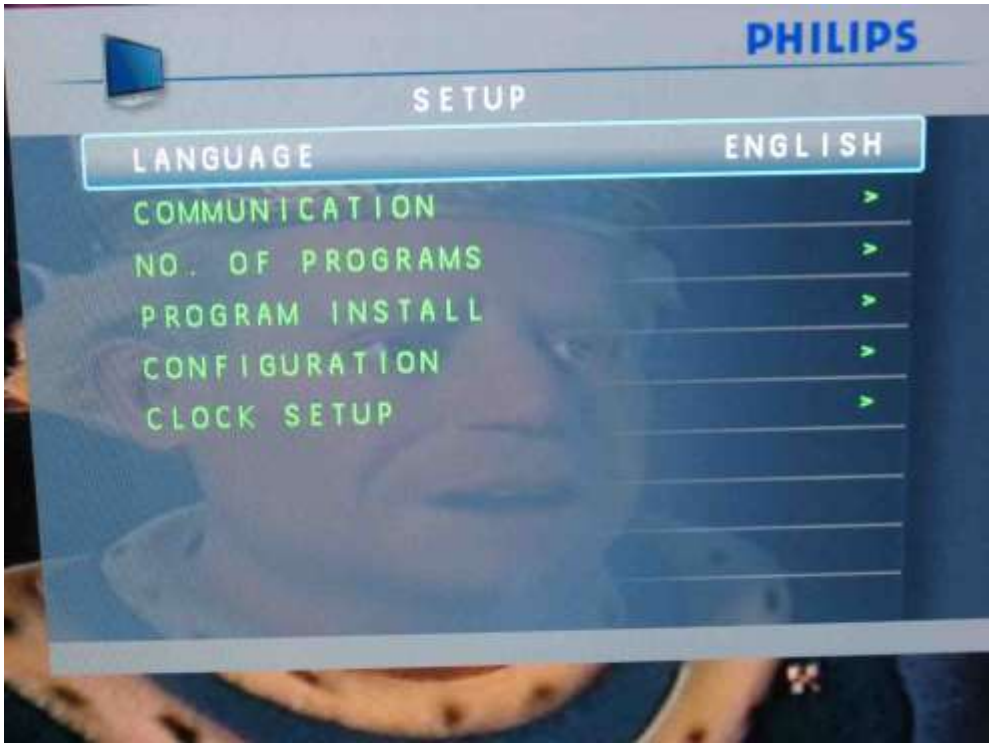


Figure 15: A snapshot of a Philips TV menu.

2.3.4 Creative applications

Those obligations are used for creating new content. These include office packages, music composition systems, or CAD applications (see Figure 16). In these systems, the user may vary from occasional very experienced. Their motivation may be high, but they may not understand the underlying computer concepts. As a consequence, it may be very difficult to design an interface for such applications.

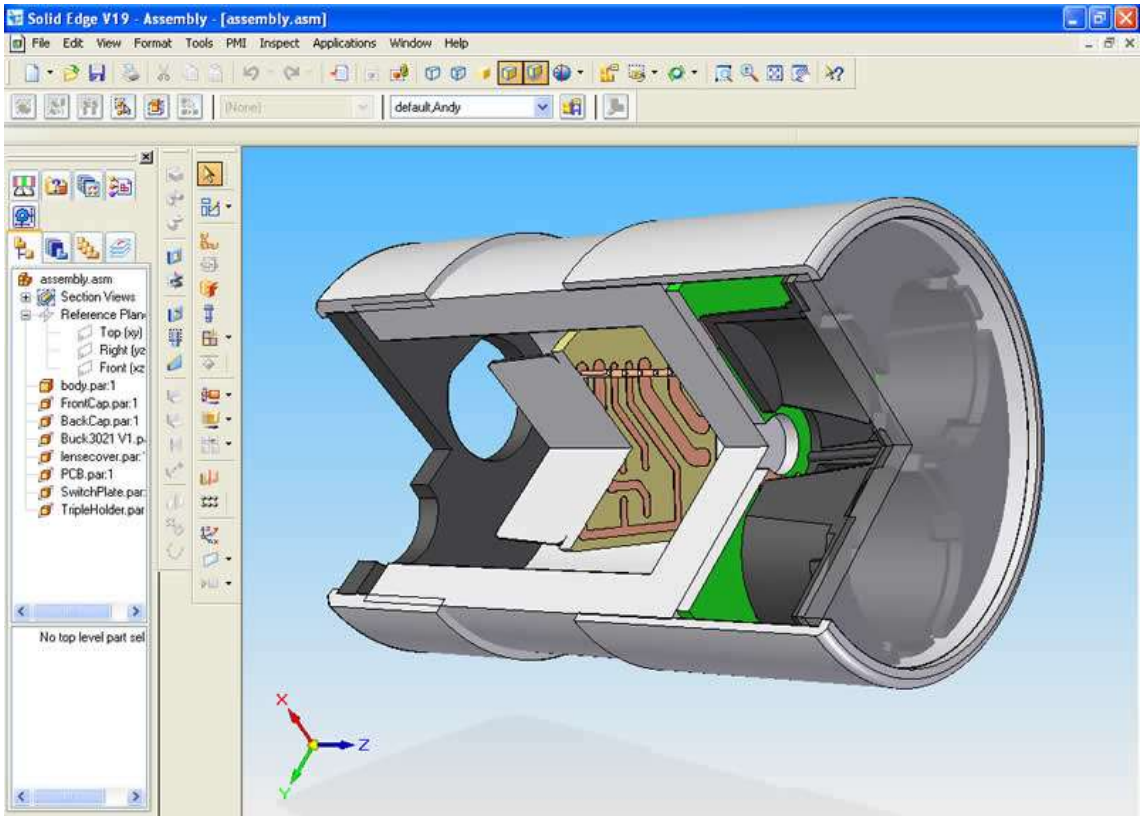


Figure 16: Solid Edge CAD system interface.

Collaborative interfaces allow two or more people to work together, even if they are separated by a space of time. These applications may require text, voice, or video in our important challenge because the way of presenting information to several users may be very difficult due to several limitations: Internet bandwidth, size of the data, or display technology. Some of his applications include design systems, such as for cars or planes, 3D data inspection systems such as for surgery planning, and so on. An example can be seen in Figure 17.

2.3.5 Collaborative systems

Figure 17: Cambiera: A collaborative system for exploring document collections.

2.3.6 Sociotechnical systems

There is a growing interest in creating systems that involve many people for a long time period. Systems for electronic voting, health support, or crime reporting, are more and more necessary in our society. Often, those systems are created by governments, and have to deal with privacy, responsibility, and the work must be performed with security. The users will have different levels of expertise in may require accessible feedback from their actions. For instance, a receipt on a vote emitted in an election could have avoided the scandal of the butterfly ballot in the 2000 United States Presidential elections. In Figure 18 we can see such ballot, belonging to the Palm Beach County. As noted in it, while the second hole belongs to the Reform party, the way the ballot is designed made a lot of electors to punch this second hole while their option was to vote Democrats. American election system allows the winner to depend only on one state, and this can be decided by a bunch of votes.

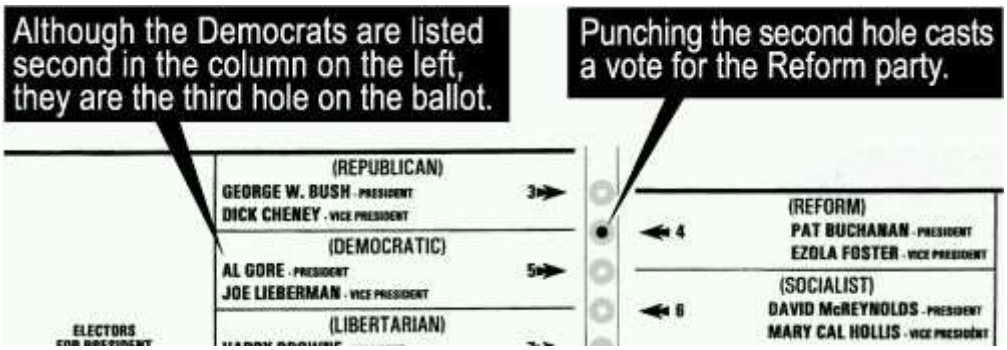


Figure 18: The famous Palm Beach County ballot.

In this district, the result was 1700 votes in favour of the Republicans over the Democrats. However, there are a series of circumstances that make the result, at least, suspicious:

First, it is a classically Democrat state. Second, Pat Buchanan obtained 3407 votes, a higher number than in other, more Republican districts. Third, 19000 votes were declared null, because the users had voted twice. Moreover, posterior findings revealed that many people found the ballot not very comprehensible. Add to this that the voters had to emit other decisions in five minutes, which made them 12 seconds for decision, and you have the probable explanation for the shocking election result.

2.4 Universal usability

There is a great amount of diversity between humans. People have different abilities, backgrounds, motivations, personalities, cultures, and work styles. These are all challenges to user interface designers. Understanding the physical, intellectual, and personality differences may be fundamental if we are focusing a broad market.

Although sometimes an application cannot adapt for every kind of user, sometimes adaptation to a certain group may benefit others. See for instance the examples of sidewalks. If we want to adapt them for wheelchairs, adding curb cuts, other groups such as parents with baby strollers, travellers with wheeled luggage, and so on, may also benefit.

In general when focusing universal usability, we should take care of the following aspects:

- Variations in physical abilities and workplaces.
- Diverse cognitive and perceptual abilities.
- Personality differences.
- Cultural and international diversity.
- Users with disabilities.
- Adult users.
- Children.
- Accommodating to software and hardware diversity.

These adaptations may come in different ways, and, as said, may benefit other groups. For instance, there is an approximately 4% of males with some kind of colour blindness (while less than 1% of women). If we adapt the colours in our user interface in a way that allows the users with colour blindness to see all the elements, this will not disturb other users.

In many cases, the adaptation to different user groups may not imply a high variation in cost or implementation, but early decisions such as including the possibility of language translation, selection of the appropriate colour palette, and so on.

3. Actual problems

Not all the possibilities in user interface development involve future products. There is a lot of problems today caused, at least not solved, by user interface design. Some of the problems are:

Organizational fragility: The availability of services throughout the Internet also make those services prone to malicious attacks, such as the ones suffered by the Sony PlayStation Network on second quarter of 2011. A high exposure to Internet requires important resources devoted to security.

Anxiety or rage: The use of computers produces anxiety on a high amount of people. Several anxieties have been identified such as computer shock, web worry, network neurosis, or computer rage. Some users feel they are going to break the machine, or they are going to lose the control over it. Sometimes this problem may become more important. In March, 2003, George Doughty, a 48-year-old from Lafayette, Colorado, shot four times his laptop computer. He was working in an office in the bar he owned went to announce to patrons that he was going to shoot his laptop computer. He warned his customers to cover their ears and then shot the computer and hung the computer on his bar wall in the style of a hunting trophy. When asked by the policemen, he said that he knew it was not correct, but, at that moment, 'It seemed appropriate'. You can check this story at (http://news.bbc.co.uk/2/hi/americas/2826587.stm).

Alienation: As people spend more time using computers, they may become less connected to other people. Nowadays, youngsters spend a lot of time chatting through Facebook or other applications. This sometimes turns people into more introverted.

Invasion of privacy: Since most of the people is spending more and more time using Internet connected devices, the amount of information stored by the companies that make the software we're using, is increasing continuously. This poses some problems when the company is making use of our private data, especially when information on the private data treatment is not clear. Facebook requires a special remarks here, since the treatment of our data has always been obscure. This has raised concerns on various governments. Confidentiality must not be compromised.

Complexity of public services: Many of our interactions with public institutions, such as tax paying, can nowadays be carried out using the Internet. However, complex interfaces, changing bureaucracy, and difficult processes make it extremely difficult for individuals to make informed choices. Sticking to basic principles of design may be a safer way to face those applications. Unfortunately, those principles are often ignored.

4. Things to improve in the future

However, it is important to remark that there are a bunch of opportunities for the future, since there are some applications that will probably be significant business in the future. Experts have identified the following distinct fields that would require appropriate interfaces:

- **Terror prevention and response:** The report after the terrorist attacks in New York and Washington determined that one of the fundamental reasons why the attacks were not properly prevented was the lack of adequate tools for the analysis of information. This may be due to the lack of proper UIs. Actually, the problem has been coined elsewhere as the *Too Much Information* problem (or TMI). Of course this is not the only factor, but its identification has fostered *Visual Analytics* field. Effective data analysis is tightly coupled with adequate interfaces.
- **Development:** Countries with poor or scarce resources may improve the efficiency of services with the use of communication technologies. Education and sanitation are two areas that may benefit from the use of technologies. These

systems may be centralized on more evolved cities or towns, and provide services as medical consultation, distance education, and so on.

- **Medical Informatics:** Remote surgery, remote diagnosis, and many other applications require high performance computers, and may benefit from modern technologies and adequate user interfaces. Modern projects by Intel and others that focus on the incorporation of sensors to house elements such as carpets in order to store and eventually evaluate person's data (even presence-non presence) open the possibilities of cheap, semi-automatic patient monitoring or diagnosis.
- **Electronic commerce:** Commercial transactions have been continuously increasing for the last years. New markets such as app stores or group shopping have added to the more *traditional* such as trips or books. Like with other services, not only the characteristics of the offer will determine the success of the company, but also the easiness on which this offer can reach to a broader audience.
- **Government services:** Citizens want to use more and more their computers to perform tasks such as tax paying, business licenses, and so on. Adequate applications, coupled with adequate citizen identification allow the governments to offer 24-hour service for several needs.
- **Creativity support tools:** Adventurous artists and researchers have always been early adopters of new technologies. Creation in art and science may benefit from advanced interfaces that allow the users to explore new compositions in the case of music, for example.

Annex. A note on Facebook's Privacy settings

Most of you have a Facebook account, and maybe some of you have been surprised by somebody else knowing some information from you that you did not expect.

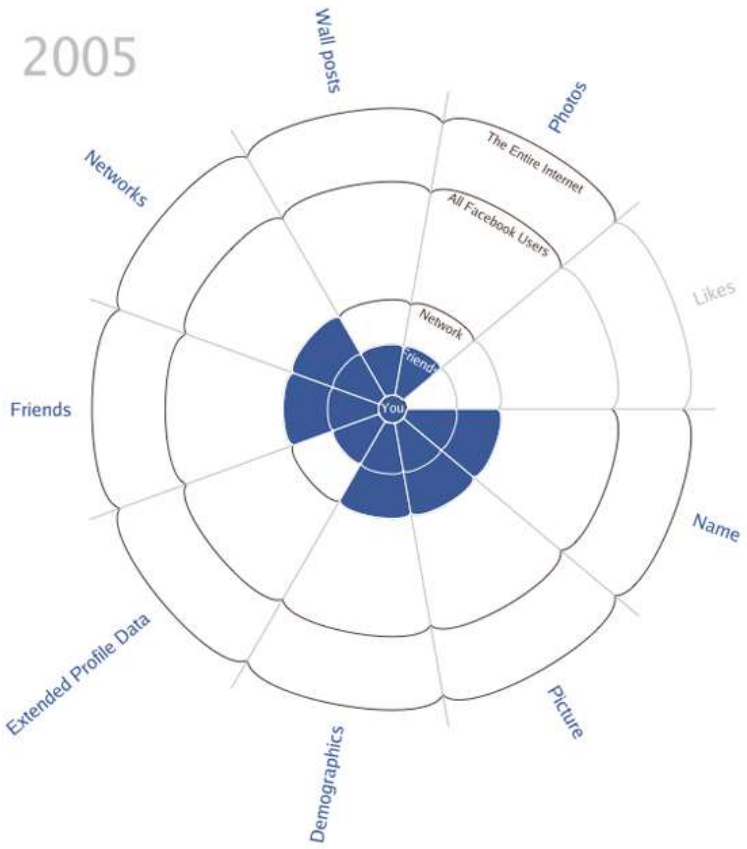
Facebook has a history on privacy disruption that started early, close to its creation. From time to time, Facebook changes its privacy settings and makes that the default information published is beyond our knowledge or control. This has happened many times since the creation of the social network.

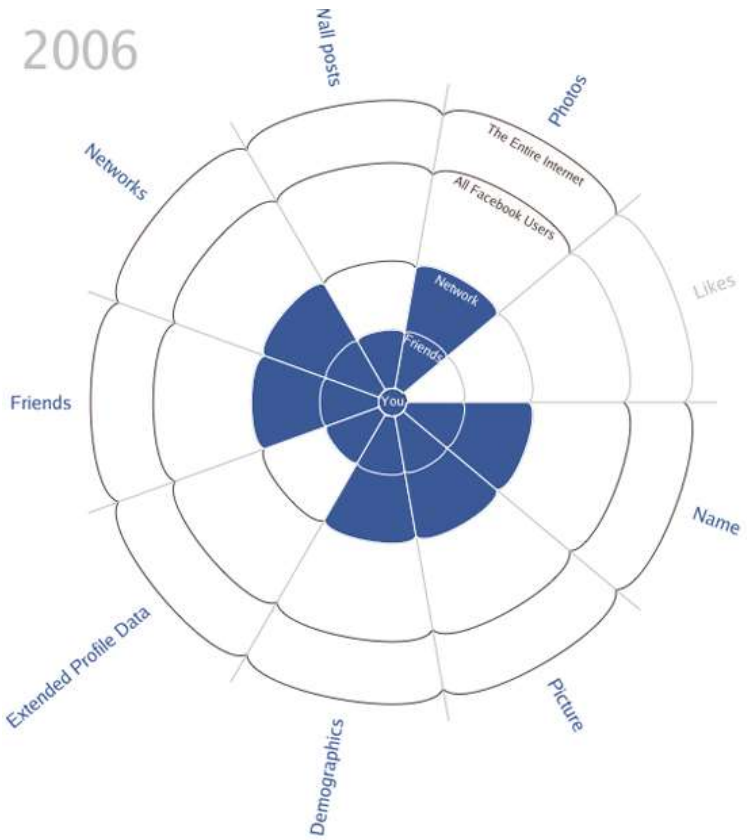
Several authors and institutions have devoted some time and effort to make those privacy invasions public and to fight against them. One of the most recent Facebook features that have raised some criticism is Places because of the possibility of someone else to tag you in a place without your authorization.

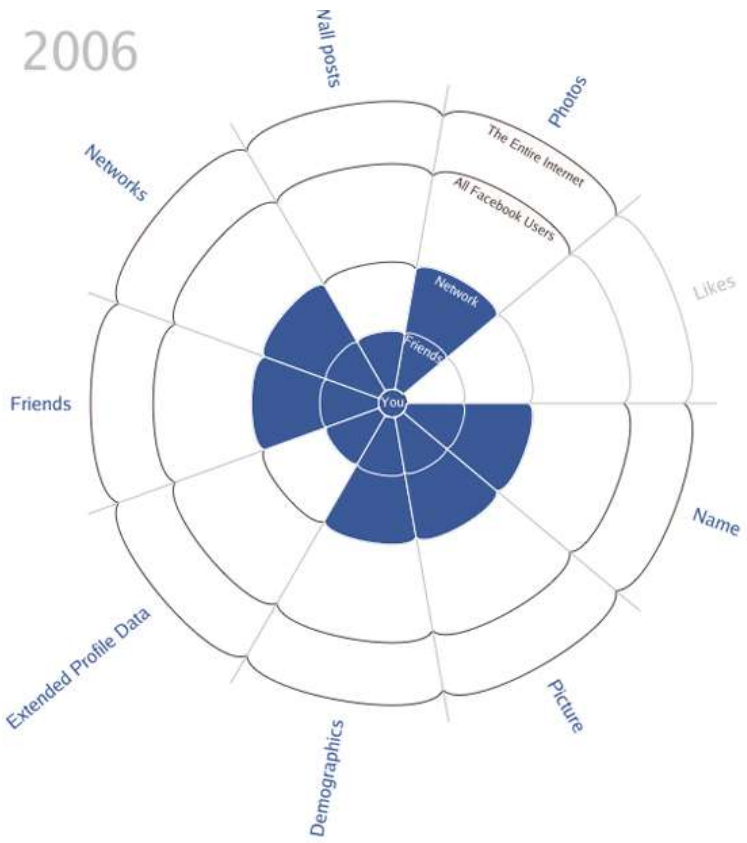
Here you have a couple of documents that were published in the web that may help you to understand the different Facebook's attacks to privacy along its short history.

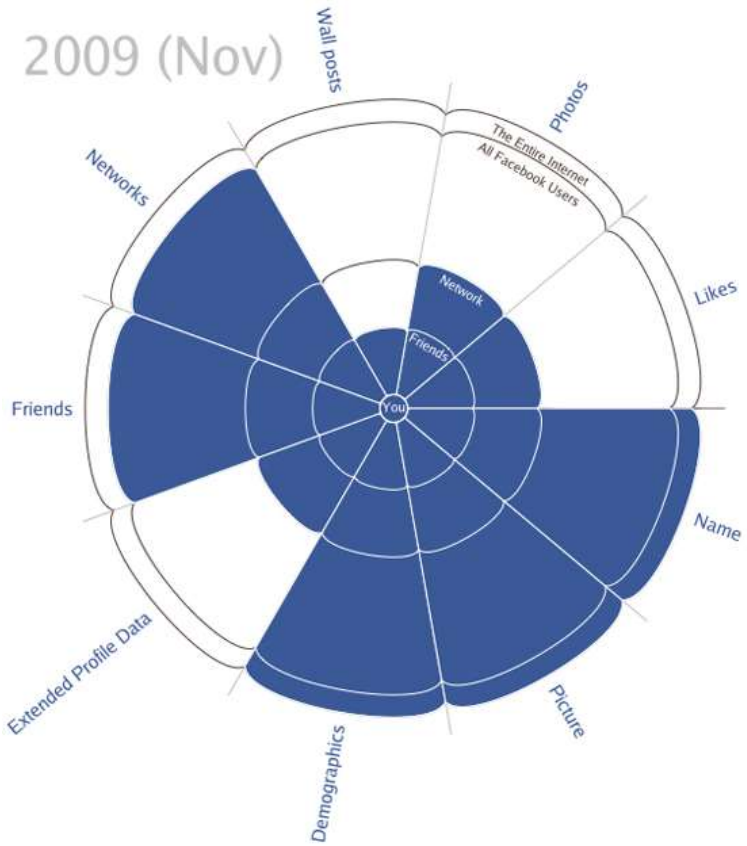
Many companies are using Facebook's profiles to filter the potential candidates of a hiring. Actually, in some places, such as in Germany, this practice has been forbidden. However, you never know if they are going to use it anyway. So, take care on what you publish on Facebook…

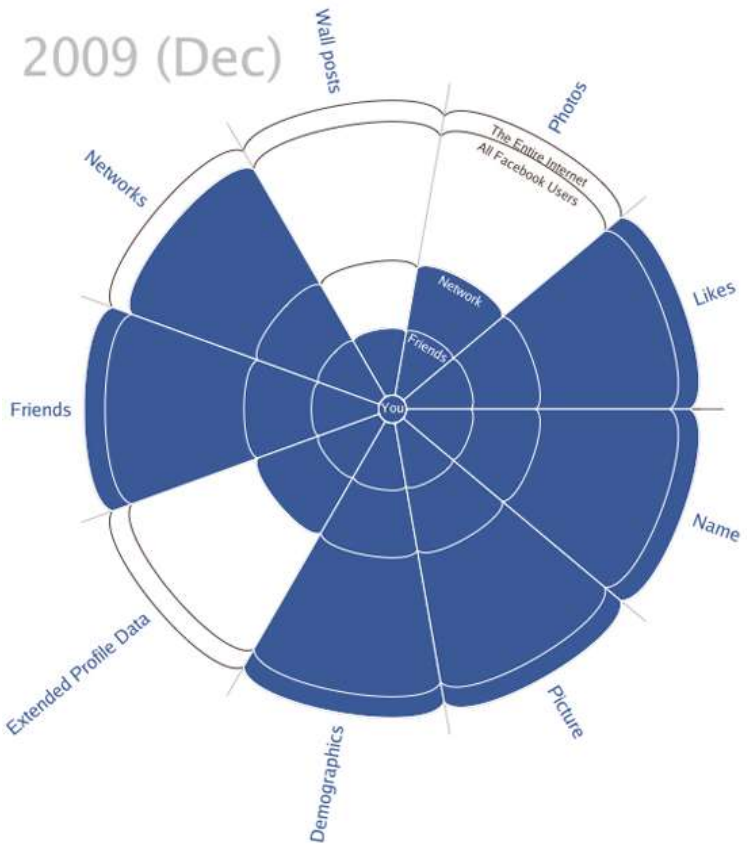
This first infographic is due to Matt McKeon, and has been remixed by allfacebook.com.

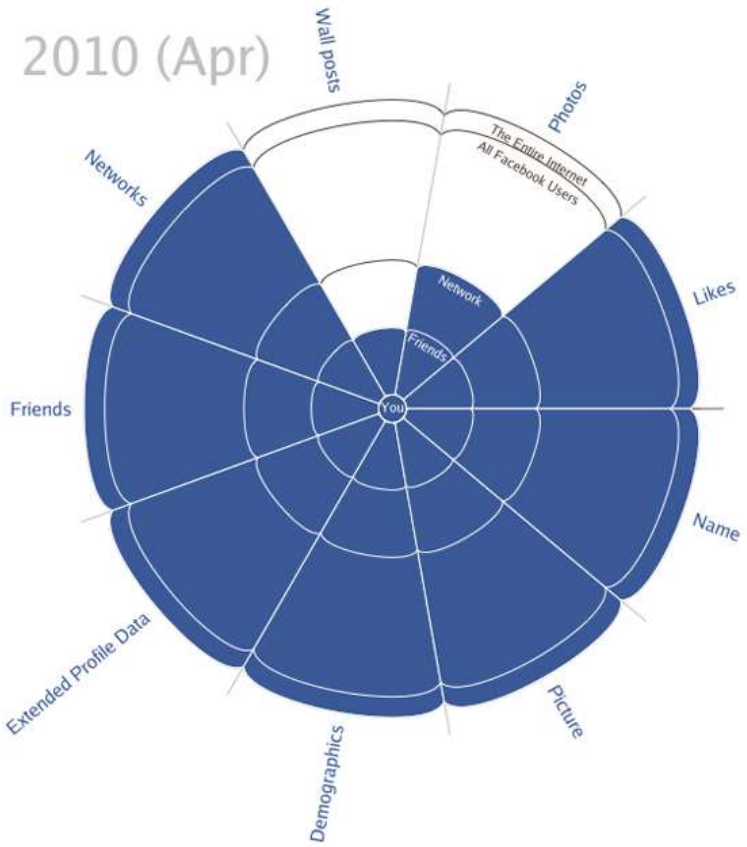








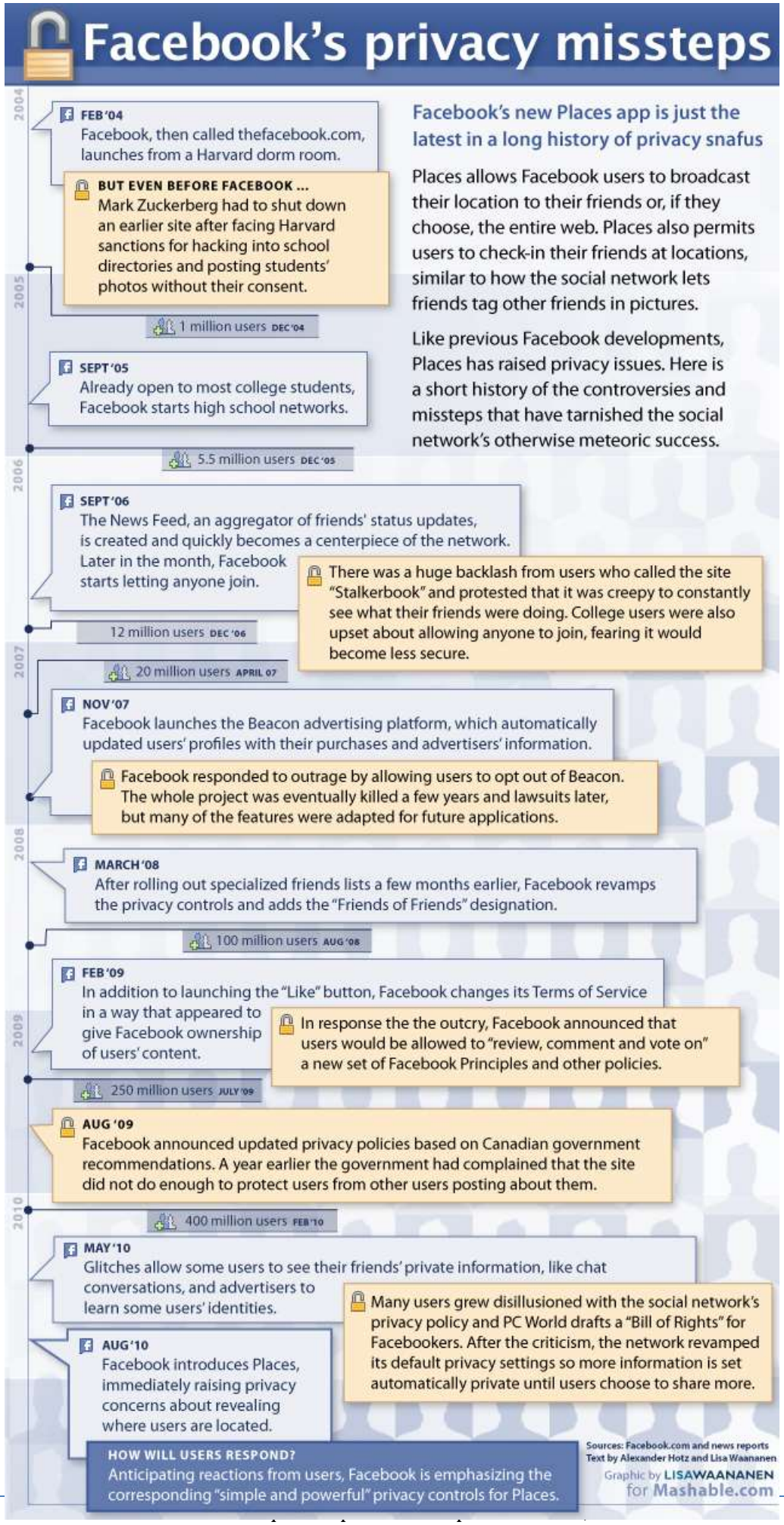

2009 (Dec)
Wall posts
Photos
The Entire Internet
All Facebook Users
Likes
Name
Picture
Demographics
Extended Profile Data
Friends
Networks
Network
Friends
You


2010 (Apr)
Wall posts
Photos
The Entire Internet
All Facebook Users
Likes
Name
Picture
Demographics
Extended Profile Data
Friends
Networks
Network
Friends
You

You may see from these pictures that, if you are not very careful, any privacy policy change does by default reveal more and more information from your profile and from your posts.

Unfortunately, it is difficult to see if this will see an end, since Facebook is still a private company whose main stockholders are private venture capital funds. There have been numerous rumours about Facebook going into a public offer, but nothing has confirmed yet. For the moment, as a private company, Facebook does not need to publish financial results. But when it goes public, stockholders will expect income from Facebook, and the obvious way is advertisement. Advertisers, as well as other companies, such as the ones that study demographics for politicians, are VERY interested in your profile information. They want to know what you like and what you dislike, and act accordingly. With a user base of more than 700 million users, Facebook is a sweet candy bar for many companies.

Lisa Waananen created the next one for Mashable. It shows how the breaches of privacy have been tightly linked to Facebook's short history:


Facebook's privacy missteps
2004
FEB '04
Facebook, then called thefacebook.com, launches from a Harvard dorm room.
BUT EVEN BEFORE FACEBOOK ...
Mark Zuckerberg had to shut down an earlier site after facing Harvard sanctions for hacking into school directories and posting students' photos without their consent.
2005
1 million users DEC '04
SEPT '05
Already open to most college students, Facebook starts high school networks.
5.5 million users DEC '05
2006
SEPT '06
The News Feed, an aggregator of friends' status updates, is created and quickly becomes a centerpiece of the network. Later in the month, Facebook starts letting anyone join.
There was a huge backlash from users who called the site "Stalkerbook" and protested that it was creepy to constantly see what their friends were doing. College users were also upset about allowing anyone to join, fearing it would become less secure.
12 million users DEC '06
2007
20 million users APRIL 07
NOV '07
Facebook launches the Beacon advertising platform, which automatically updated users' profiles with their purchases and advertisers' information.
Facebook responded to outrage by allowing users to opt out of Beacon. The whole project was eventually killed a few years and lawsuits later, but many of the features were adapted for future applications.
2008
MARCH '08
After rolling out specialized friends lists a few months earlier, Facebook revamps the privacy controls and adds the "Friends of Friends" designation.
100 million users AUG '08
FEB '09
In addition to launching the "Like" button, Facebook changes its Terms of Service in a way that appeared to give Facebook ownership of users' content.
In response the the outcry, Facebook announced that users would be allowed to "review, comment and vote on" a new set of Facebook Principles and other policies.
2009
250 million users JULY '09
AUG '09
Facebook announced updated privacy policies based on Canadian government recommendations. A year earlier the government had complained that the site did not do enough to protect users from other users posting about them.
2010
400 million users FEB '10
MAY '10
Glitches allow some users to see their friends' private information, like chat conversations, and advertisers to learn some users' identities.
Many users grew disillusioned with the social network's privacy policy and PC World drafts a "Bill of Rights" for Facebookers. After the criticism, the network revamped its default privacy settings so more information is set automatically private until users choose to share more.
AUG '10
Facebook introduces Places, immediately raising privacy concerns about revealing where users are located.
HOW WILL USERS RESPOND?
Anticipating reactions from users, Facebook is emphasizing the corresponding "simple and powerful" privacy controls for Places.
Facebook's new Places app is just the latest in a long history of privacy snafus
Places allows Facebook users to broadcast their location to their friends or, if they choose, the entire web. Places also permits users to check-in their friends at locations, similar to how the social network lets friends tag other friends in pictures.
Like previous Facebook developments, Places has raised privacy issues. Here is a short history of the controversies and missteps that have tarnished the social network's otherwise meteoric success.
Sources: Facebook.com and news reports
Text by Alexander Hotz and Lisa Waananen
Graphic by LISAWAANANEN for Mashable.com

References

Many useful articles can be found in the following pages.

- http://usability.gov
- http://www.smashingmagazine.com
- http://www.nngroup.com/articles/
- http://www.interaction-design.org/encyclopedia/
- http://www.usabilitycounts.com