Design for Interaction

Planning and Design

Contents

- identifying needs and requirements
- task description and analysis
- conceptual and practical design

Needs Analysis

- for any development task, we must understand what it is we are trying to achieve
- the first stage in this process is specifying the needs and requirements associated with the project
  - overall project goals
  - range of functionality we wish to cover

Why Bother?

- is this stage really necessary?
- YES!
- the main causes of failed development projects are:
  - poor specification of requirements
  - lack of understanding of project goals
- without a clear plan based on an analytical assessment, any project is liable to fail

Requirements

- can be divided into two types
- functional requirements:
  - what a system should be able to do
  - example: image editing software might be required to handle a specific range of formats
- non-functional requirements:
  - what constraints the system will work under
  - example: the image editing package might be required to run on a handheld device

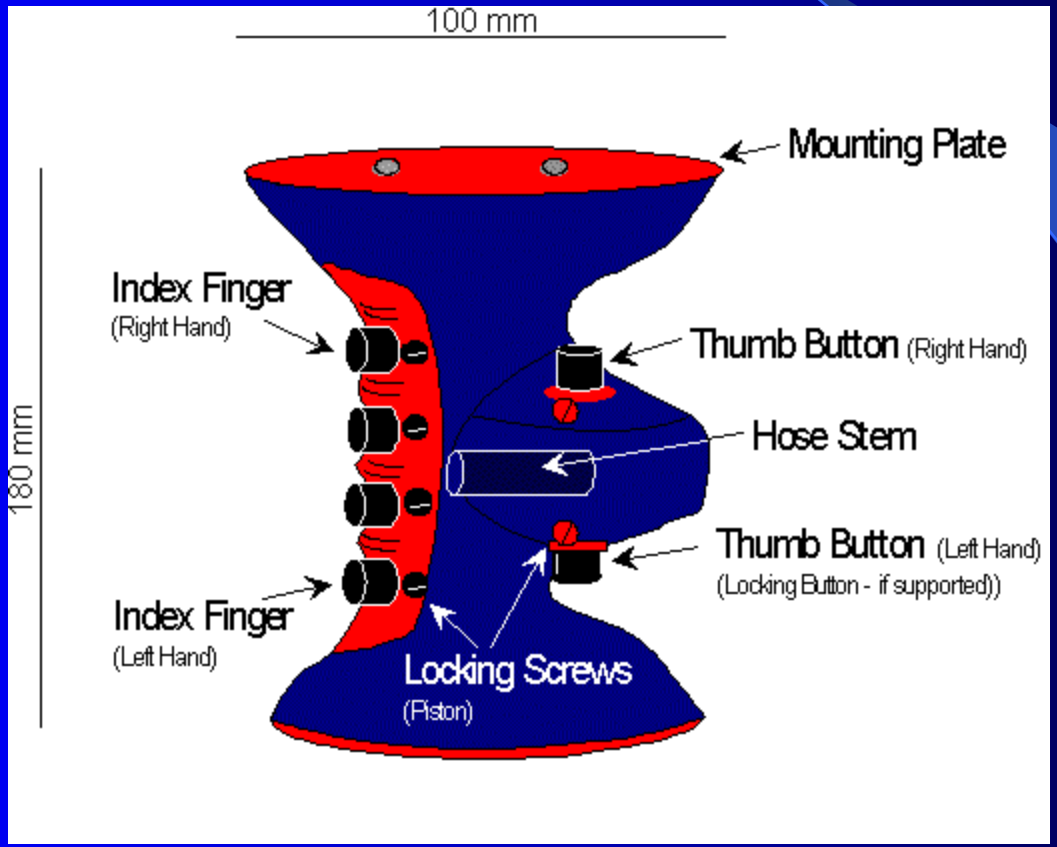
Non-functional Requirements

- various categories:
  - look-and-feel requirements
  - usability requirements
  - performance requirements
  - operational requirements
  - security requirements
  - legal requirements
  - environment / portability requirements

Example: Underwater PC

- needed for undersea divers carrying out tasks such as maintaining oil platform supports or sea surface mapping
- functional requirements: PC functionality
- non-functional requirements:
  - waterproofing
  - input device?

Underwater Input Device (KordGrip)



Functional Requirements

- important distinction between genuinely novel products and those which replicate existing functions in a new context
  - previous slide showed an example of the latter case
- for new products, the specification process must begin by looking at the tasks it will be used for
  - task description and analysis

Task Description

- mechanism for determining what roles and requirements a potential product will need to fulfil
- examples:
  - setting up a website to support an existing business
  - creating an interface for a media player device
- in each case, we need to allow certain functionality under specific constraints

Scenario-Based Description

- one useful technique is to use scenarios
  - scenario: “informal narrative description”
- consider the various narratives through which the system might be of use
- for the website example, these may include:
  - an existing customer / client accessing the site
  - a new customer / client accessing the site
  - the business owner editing the site

Using Scenarios

- the use of scenarios has the advantage of being intuitive and easy to relate to
- offers a reflection of people's own experiences
- may be useful for identifying user experience goals
- however, they can also be limiting and ignore wider issues of implementation

User Cases

- an alternative approach to task description is to identify user cases
- these are more specific, and focus on the interactions between a user and the system
- example cases for the media player:
  - user wants to search for a specific item
  - user wants to play tracks at random
  - user wants to create a new playlist

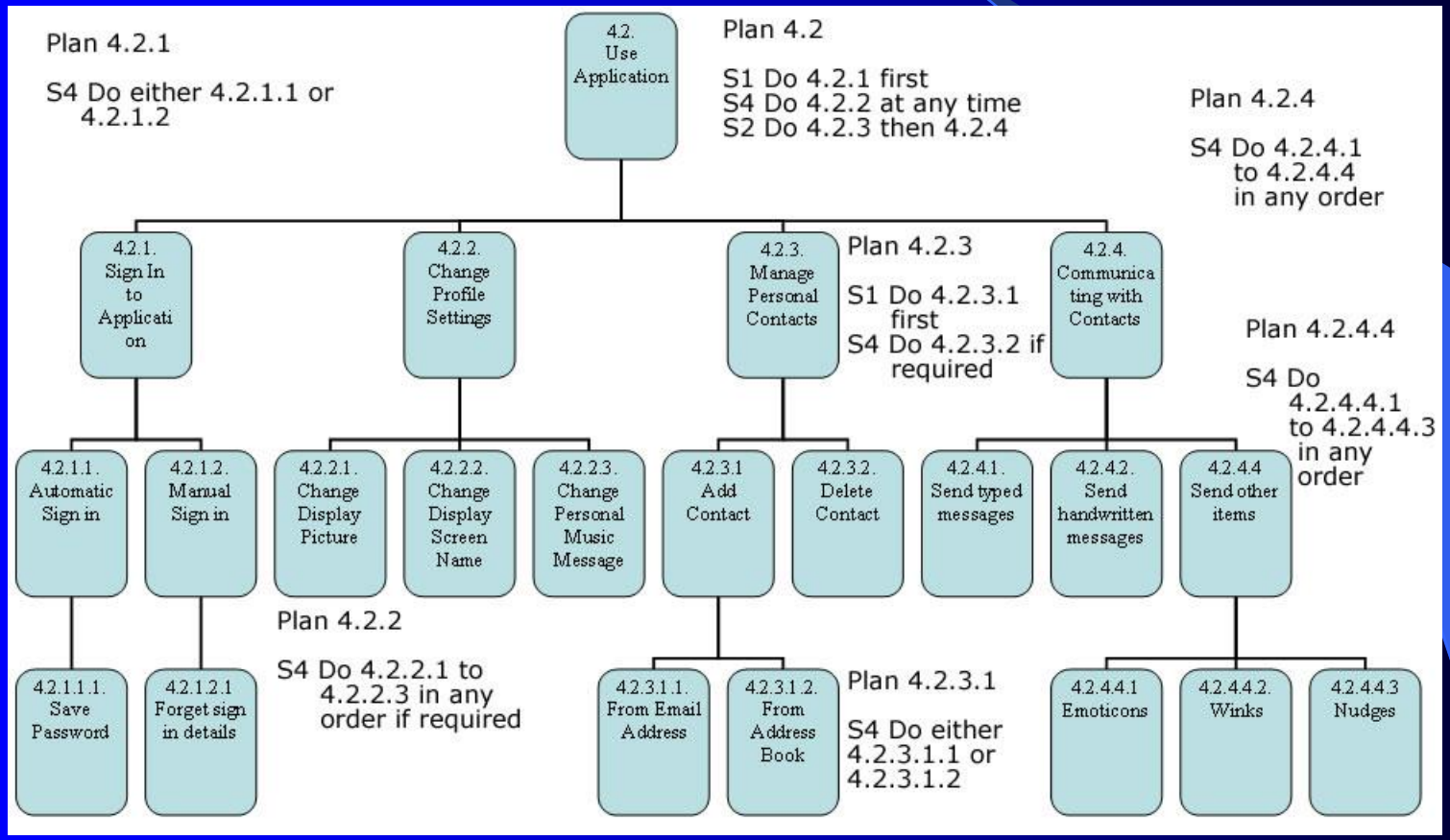
Using User Cases

- user cases have the benefit of allowing specific interaction goals to be identified and analysed
- description can focus more on the fine details of interactivity
- one problem with this approach is that it can enforce assumptions about how a system will work
  - eg. assuming we will have a playlist facility

Task Analysis

- our set of descriptions of interaction tasks can be refined using task analysis
- essentially, this breaks down tasks into their basic elements
  - reflects practice in software design
- a common method is hierarchical task analysis (HTA)

Example: HTA for Using MSN Messenger



Conceptual and Practical Design

- the design process can be divided into two distinct areas
- conceptual design considers what form the product will take
  - what role(s) will our product fulfil?
  - how (in a basic sense) will we interact with it?
- practical design looks at the details of how this will be carried out
  - sometimes referred to as physical design

Conceptual Design

- this is the part of the development process where imagination and creativity are most important
- at the same time, designers must bear in mind usability and the project requirements
- issues for conceptual design for interaction:
  - which interaction modes will be used
  - do we invoke a particular interface metaphor
  - can we identify an interface paradigm

Metaphors and Paradigms

- often users find metaphors helpful in understanding an application's role
  - good when the metaphor relates directly to a well-understood real-world system
  - for example, e-mail uses metaphors from real-world mail systems (eg. mailbox)
- a paradigm is a governing principle determining the workings of a system

Interaction Paradigm Examples

- classic example of an interaction paradigm is the Windows-style (WIMP) interface
  - replacement for command-line interaction
- similarly, web browsers radically altered the ways in which people could access and interact with online information
- touch-screen displays and motion-sensitive game controllers are other examples

Practical Design Issues

- it is in the practical design process that we must apply the rules discussed in previous lectures
  - usability aims
  - supporting cognitive processing
  - affective factors
- we also need to consider the constraints we identify as non-functional requirements

Practical Design Efficiency

- it is generally useful to make our practical designs as efficient as possible
- reuse of design elements and components is an example of this:
  - saves development time
  - helps developers maintain consistency, which in turn is good for users
- efficiency must be considered from the very beginnings of a project