

http://commons.wikimedia.org/wiki/File:Joy_Oil_gas_station_blueprints.jpg

Unit 3

Conceptual Design

Interfaces Persona Computador

Depto. Sistemas Informáticos y Computación. UPV

VIRTUAL TEACHING

Purpose:

Provision of the higher education public service
(art. 1 LOU)

Responsible:

Universitat Politècnica de València.

**Rights of access, rectification, deletion,
portability, limitation or opposition to the
treatment in accordance with privacy
policies:**

<http://www.upv.es/contenidos/DPD/>

Intellectual property:

Exclusive use in the virtual classroom
environment.

Dissemination, distribution or disclosure of
recorded classes, particularly on social networks
or services dedicated to sharing notes, is
prohibited.

Violation of this prohibition may generate
disciplinary, administrative or civil liability.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Unit Goals

- ▶ Understand the difficulty of translating the requirements into a GUI
- ▶ Learn the importance of conceptual design
- ▶ Study a technique for describing the underlying structure of a UI
- ▶ Be able to generate a content diagram from a set of concrete use cases

Overview

- ▶ Introduction
- ▶ Conceptual Design: The content diagram
 1. Identify the primary task objects, attributes and actions
 2. Identify the different containers and the task objects that go into each one
 3. Link the containers to show the navigation flow
- ▶ References

Introduction

- ▶ This unit describes some techniques for bridging the gap between requirement analysis and the physical design of a UI

"The system should be able to create a monthly sales report"



Introduction

- ▶ Work reengineering:
 - Many projects consist of developing a new version of an existing system
 - Maybe users will have to start working differently
 - Sensitive handling: involve users in the development process
- ▶ Goals of the work reengineering:
 - Realize the power and efficiency of automation
 - Reengineer the work for more effectively support of business goals
 - Minimize retraining by taking advantage of user's current knowledge and take into account human cognitive constraints and capabilities when designing new tasks

Example: Digital Library

- ▶ The analysis phase has produced a number of items:
 - Task scenarios

Task scenario. Search and request resource

Julia, a lecturer in the department, is looking for a particular CD-ROM containing examples and exercises on Object Oriented Analysis and Design. She knows that Tom, another lecturer, mainly teaches Object Oriented Analysis and Design so she knocks on his door. Unfortunately he is not there, so she leaves a note on his door. Later he returns and searches for her, finding her in the coffee bar. He tells Julia that Geoff has the CD-ROM. Unfortunately Geoff is on leave, so Julia telephones him and he promises to post it to her.

Task scenario. View updates and request resource

Mark has recently returned from six months of study leave and wants to find out what books other members of the department have bought since he left. To do this he telephones everyone in the department and arranges an appointment. He has to do this because everyone is at the university at different times. He then meets everyone individually and checks through their bookcases, asking to borrow books that interest him. He only asks for one book at a time, as he is a slow reader!

Example: Digital Library

- ▶ The analysis phase has produced a number of items:
 - Use scenarios

Use scenario. Search and request resource

Julia is looking for a particular CD-ROM containing examples and exercises on Object Oriented Analysis and Design. She accesses the digital library from home and types in the key phrase 'Object Oriented Analysis'. The system retrieves one result. Geoff owns the appropriate CD-ROM. Julia then sends an e-mail to Geoff, asking to borrow the CD-ROM.

Use scenario. View updates and request resource

Mark has recently returned from study leave and wants to find out what are the latest additions to the digital library. He selects 'check updates', identifies the books he is interested in, and sends an e-mail to the owner of the one that interests him most.

Example: Digital Library

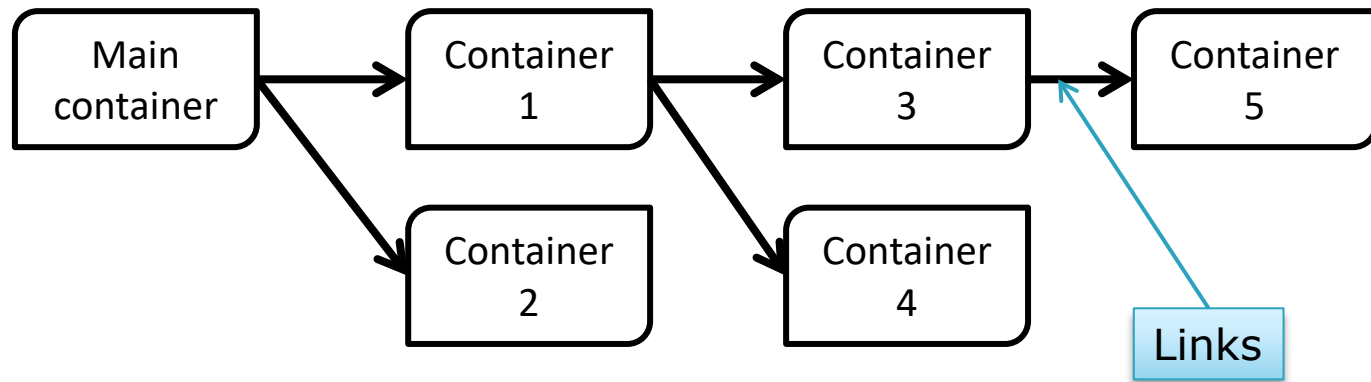
- ▶ The analysis phase has produced a number of items:
 - Concrete use cases

User action	System response
The academic enters one or more of the search parameters for the CD-ROM: title, year and platform	The system displays the search results
The academic selects a search result	The system displays the full details of the CD-ROM and the contact details for its owner who is a research student
The academic chooses the e-mail address	The system displays a message area
The academic writes and sends the e-mail request	The system confirms the sending of the request

User action	System response
The research student requests recent updates in the digital library	The system displays the availability of the latest books, CD-ROMs, videos and journals
The student selects this year's book by her favourite author: J. Nielsen	The system displays the full details of the book and contact details, including name, for the owner, who is an academic
The student chooses the e-mail address	The system displays a message area
The student writes and sends the e-mail request for the book	The system confirms the sending of the request

Conceptual Design

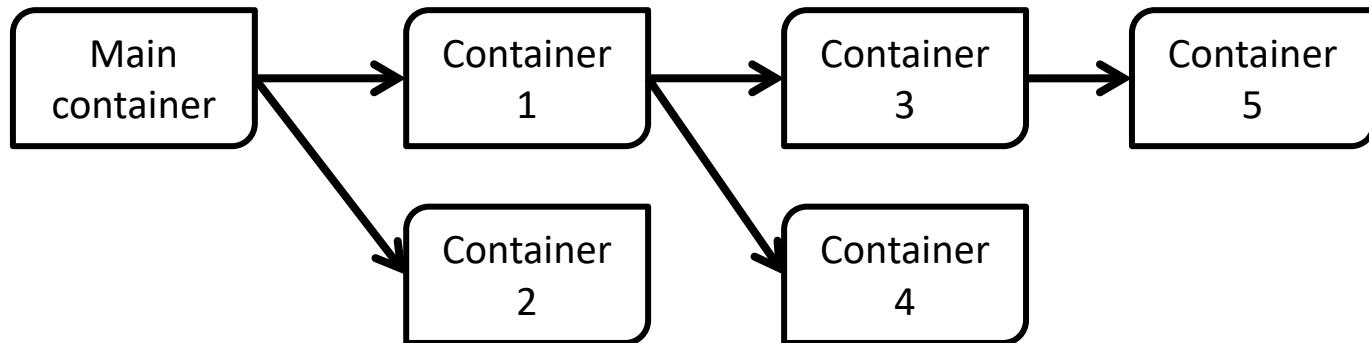
- ▶ Conceptual design is the process of establishing the underlying organization and structure of a UI
 - Decide which functionality each screen should support
- ▶ A content diagram is a low-fidelity prototype that represents the organization and structure of a UI from the designer's perspective



Conceptual Design

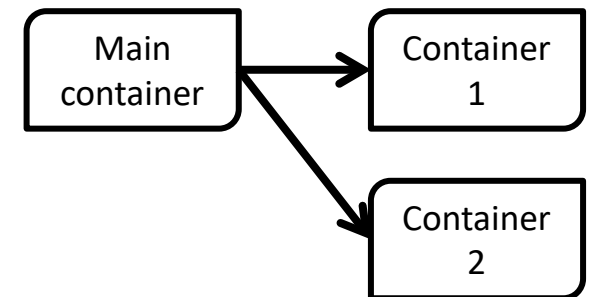
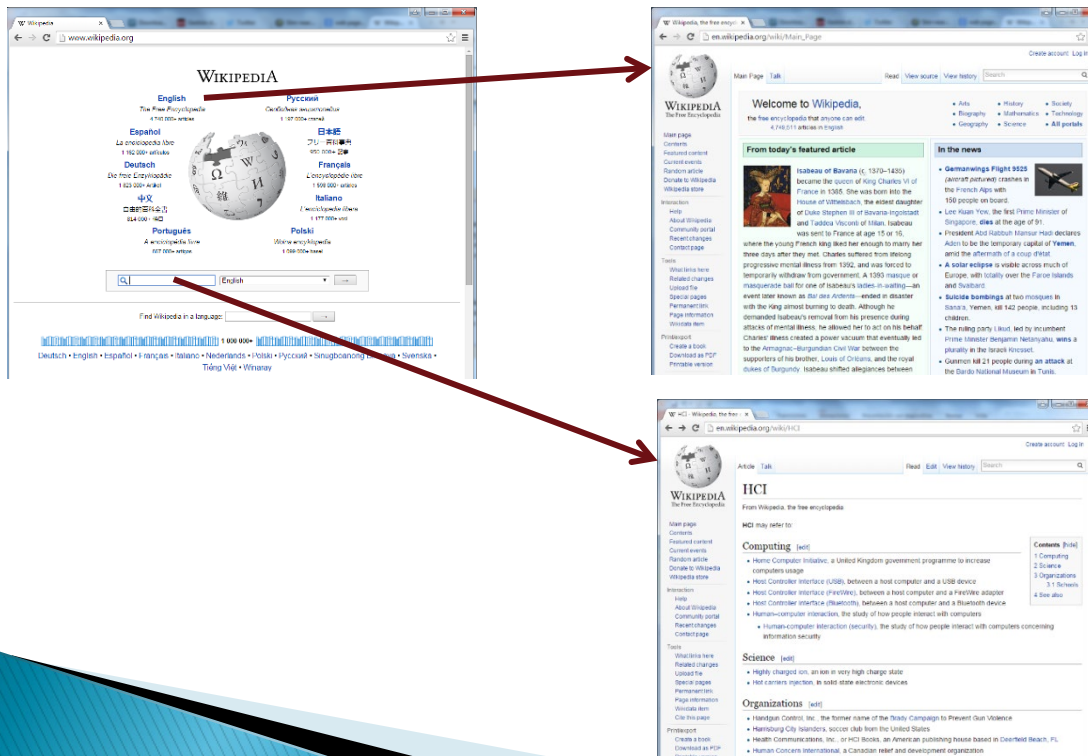
► Content diagram

- Containers: abstract representation of a part of the user's work and the supporting functions
 - E.g. enter the search criteria for a book
- Links: how the user will navigate between the functional areas within the UI



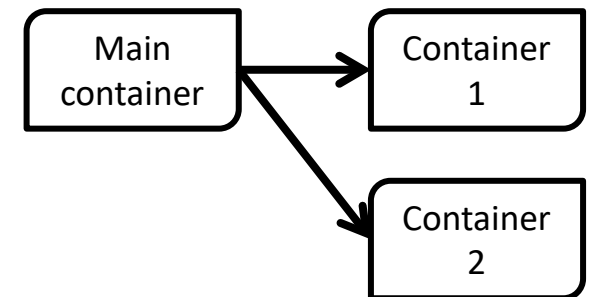
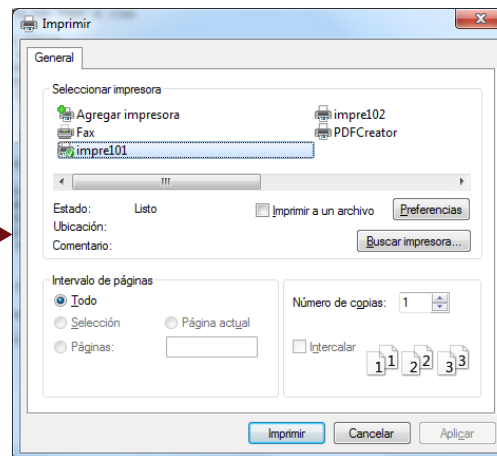
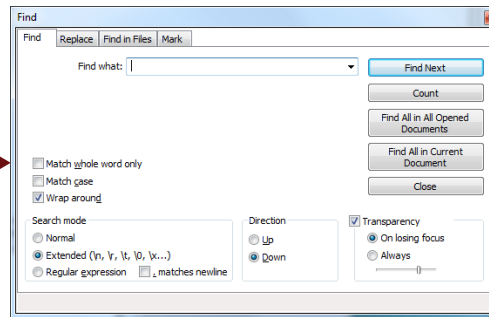
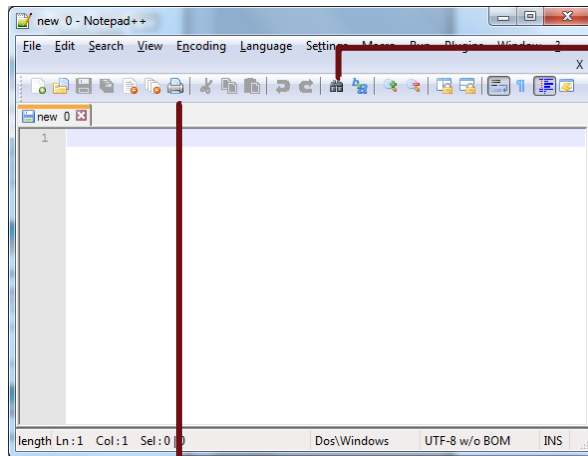
Conceptual Design

- ▶ Possible relations between the content diagram and the final UI
 - In the web



Conceptual Design

- ▶ Possible relations between the content diagram and the final UI
 - In a GUI



Conceptual Design

- ▶ Possible relations between the content diagram and the final UI
 - Voice-based interaction



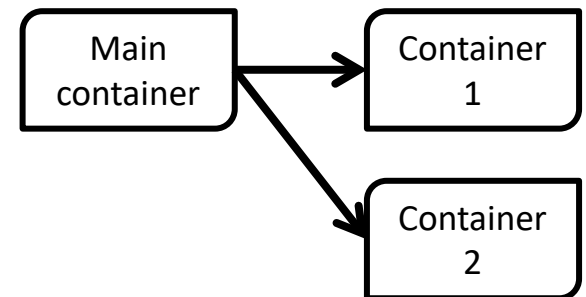
"Press 1 if you want to make an appointment or 2 if you want to cancel an appointment"



"Please, say the name of the doctor you want to make an appointment with"



"Please, type the telephone number you gave for your appointment"



Conceptual Design

- ▶ Possible relations between the content diagram and the final UI
 - When designing the contents and structure of the UI, the content diagram should serve as a guide, not as a strict recipe (for example, two containers may be combined in one screen)
- ▶ The content diagram is created from information obtained during the requirements gathering, and from the concrete use cases
 - It will probably be incomplete, but it is helpful for identifying the main functional areas and their relations

Conceptual Design: process

- ▶ Steps for creating a content diagram:
 1. Identify the primary task objects, attributes and actions
 2. Identify the different containers and the task objects that go into each one
 3. Link the containers to show the navigation flow
- ▶ This is a creative process, and should be improved iteratively
 - Use post-it notes and posters



Conceptual Design: process

Identify task objects, attributes and actions

- ▶ Step 1: Identify the primary task objects, attributes and actions
 - Used to decide what goes into each container, and the links needed between containers
 - Similar to object-oriented design:
 - Primary task objects: entities with data which the users interact to carry out their tasks
 - Attributes: the task objects' properties or links to other objects
 - Actions: functions the user can invoke on the task objects

Conceptual Design: process

Identify task objects, attributes and actions

- ▶ Primary task objects
 - High level objects; there are usually a few
 - Example: book, CD-ROM, video, journal, academic staff, research staff, research student
 - Where to look for task objects: requirements documentation and concrete use cases
 - Hint: find entities that can be searched through or modified in some way, or artifacts like forms, documents, papers, lists...

Conceptual Design: process

Identify task objects, attributes and actions

► Attributes

- A task object must have attributes, otherwise it is an attribute of another object
- Two kinds of attributes:
 - Properties: data that belongs exclusively to the object
 - Child objects: it is a task object in its own right, but has a relationship with the parent object
 - In a GUI, the child object will appear whenever the parent object is shown, usually inside it (either fully or as a reduced version)
- The task objects and its attributes will be translated onto a combination of user interface objects (list boxes, combo boxes, radio buttons, data fields, and so on.)

Conceptual Design: process

Identify task objects, attributes and actions

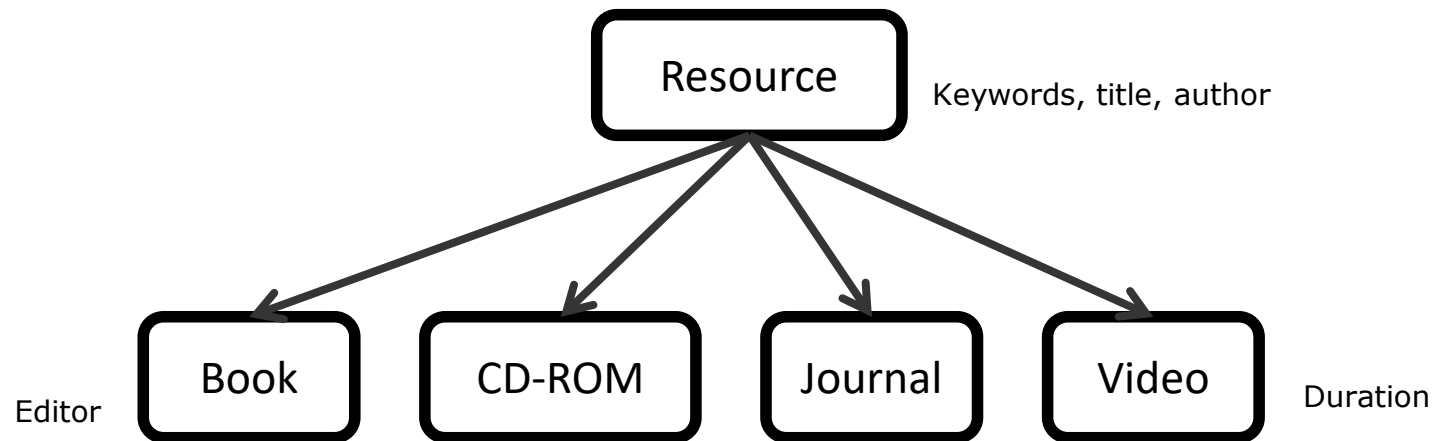
▶ Actions

- Users perform actions on the task objects (reserve a flight, create a new order, search a book...)
- Apart from the specific actions of the task objects, also consider the standard actions as view, create, delete, copy, save, edit, print...
- The actions will be translated onto menu items or items on the tool bar

Conceptual Design: process

Identify task objects, attributes and actions

- ▶ Primary task objects
 - Often task objects can be grouped in classes, that abstract the common attributes



Conceptual Design: process

Identify task objects, attributes and actions

- ▶ Using the concrete use cases for identifying task objects and their attributes
 - single-underline nouns that may correspond to task objects
 - double-underline the attributes of these task objects
- ▶ Verbs usually correspond to actions, but we won't mark them because the relationships are often less direct

Conceptual Design: process

Identify task objects, attributes and actions

► Example

User action	System response
The academic enters one or more of the search parameters for the CD-ROM: title, year and platform	The system displays the search results
The academic selects a search result	The system displays the full details of the CD-ROM and the contact details for its owner who is a research student
The academic chooses the e-mail address	The system displays a message area
The academic writes and sends the e-mail request	The system confirms the sending of the request

Conceptual Design: process

Identify task objects, attributes and actions

▶ Another example

User action	System response
The research student requests recent updates in the digital library	The system displays the availability of the latest books, CD-ROMs, videos and journals
The student selects this year's book by her favourite author: J. Nielsen	The system displays the full details of the book and contact details, including name, for the owner, who is an academic
The student chooses the e-mail address	The system displays a message area
The student writes and sends the e-mail request for the book	The system confirms the sending of the request

Conceptual Design: process

Identify task objects, attributes and actions

- ▶ After identifying the task objects and attributes, compile them in one table per task object

Task object	Attributes	Actions
CD-ROM	Keywords	Reserve
	Author	View
	Title	Add
	Year	Print
	Platform	Delete
	Owned by (academic, researcher or research student)	Save
		Edit

From the domain analysis

From the concrete use case

Specific

Standard

Conceptual Design: process

Identify task objects, attributes and actions

- ▶ After identifying the task objects and attributes, compile them in one table per task object

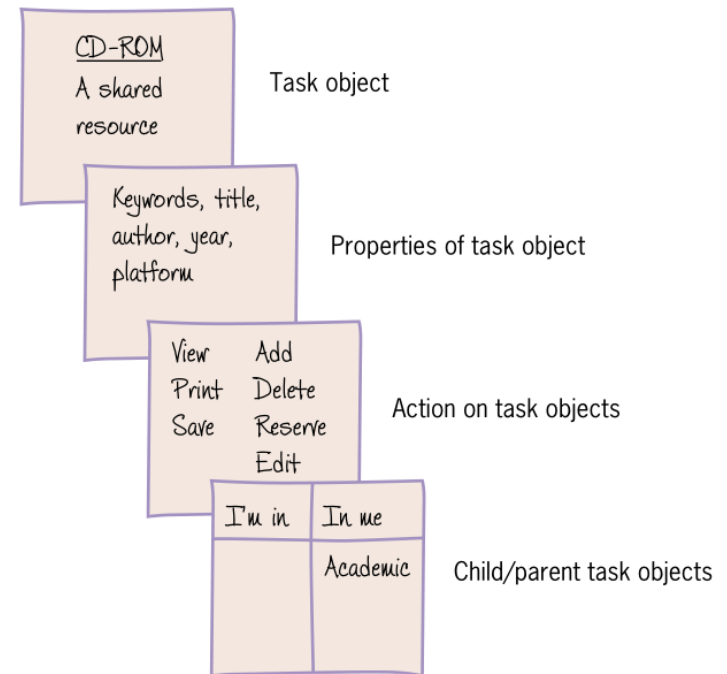
Task object	Attributes	Actions
Academic	Name Phone number Office number E-mail address	View Add Edit Print Save Delete

Conceptual Design: process

Identify task objects, attributes and actions

- ▶ Since it is hard to capture all the task objects, attributes and actions from the concrete use cases, we should use an iterative approach

- Find information in the concrete use cases, but also in the user's knowledge of the domain, or from your own domain's analysis
- Prototype your ideas and evaluate them with users



Conceptual Design: process

Identify the different containers and the task objects that go into each one

- ▶ Step 2: Identify the different containers and the task objects that go into each one
 - Each container helps the user to perform some work by collecting the required functions and task objects
 - They will become screens, windows, dialog boxes or message boxes
 - In the next step, we will connect containers with links

Conceptual Design: process

Identify the different containers and the task objects that go into each one

- ▶ Elements in a container:
 - Name: the name of the container
 - Purpose: a sentence with the purpose in supporting the user's task
 - Functions: ● invoked by the user, ■ invoked by the system
 - Links: The name of a container it is connected to: ▶ the new container replaces the current one, ▶▶ both containers work in parallel
 - Objects: the task objects whose attributes and actions are required
 - Constraints: any constraints for the container, such as speed, reliability, and availability

Conceptual Design: process

Identify the different containers and the task objects that go into each one

▶ Template for containers

Name

Purpose

Functions

- {performed by the user}
- {performed by the computer system}

Links

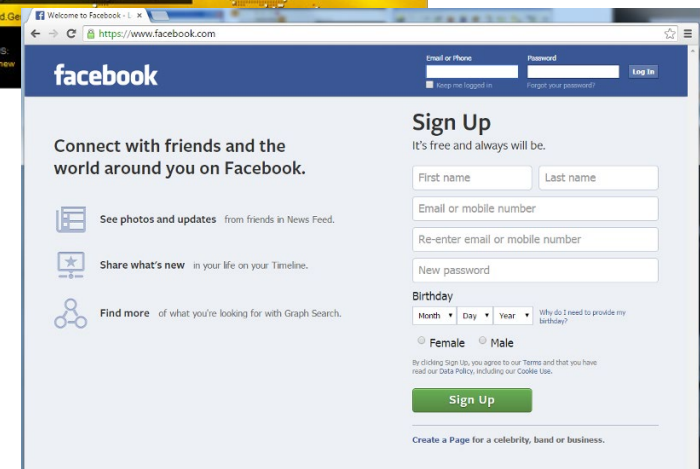
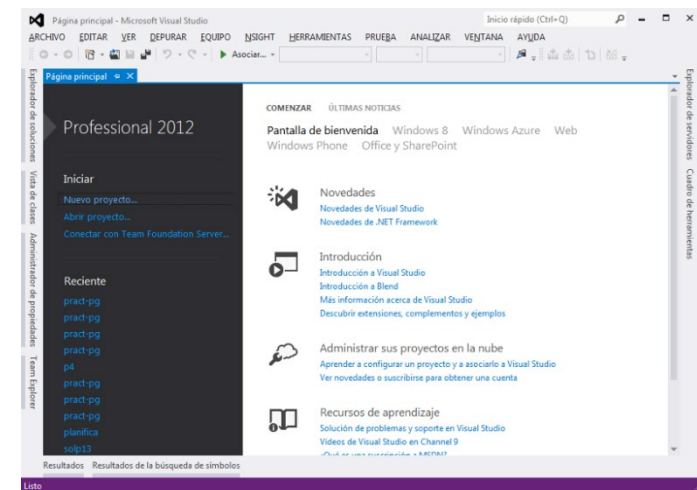
- ▶ {single link}
- ▶▶ {double link}

Objects

Constraints

Identify the different containers and the task objects that go into each one

- ▶ The main container represents the first screen the user encounters



Conceptual Design: process

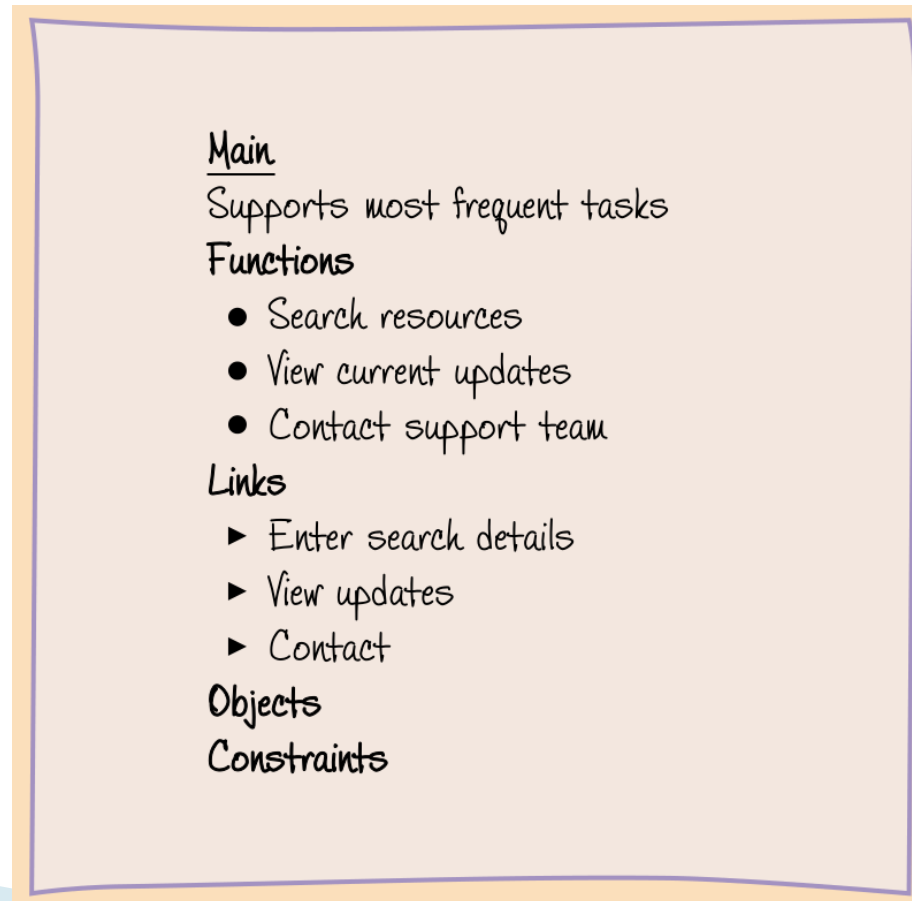
Identify the different containers and the task objects that go into each one

- ▶ The main container will have links to
 - Vital tasks: the user must perform these tasks quickly, even under stress
 - Frequent tasks: tasks the users spend the majority of their time performing. Should be fast to access
 - Navigational aids: helps the user to find what the application is capable of doing
- ▶ The main container does not perform any of those actions, it just provides the links to the container that will do

Conceptual Design: process

Identify the different containers and the task objects that go into each one

- ▶ Example of the main container of the library



Conceptual Design: process

Identify the different containers and the task objects that go into each one

▶ Other containers

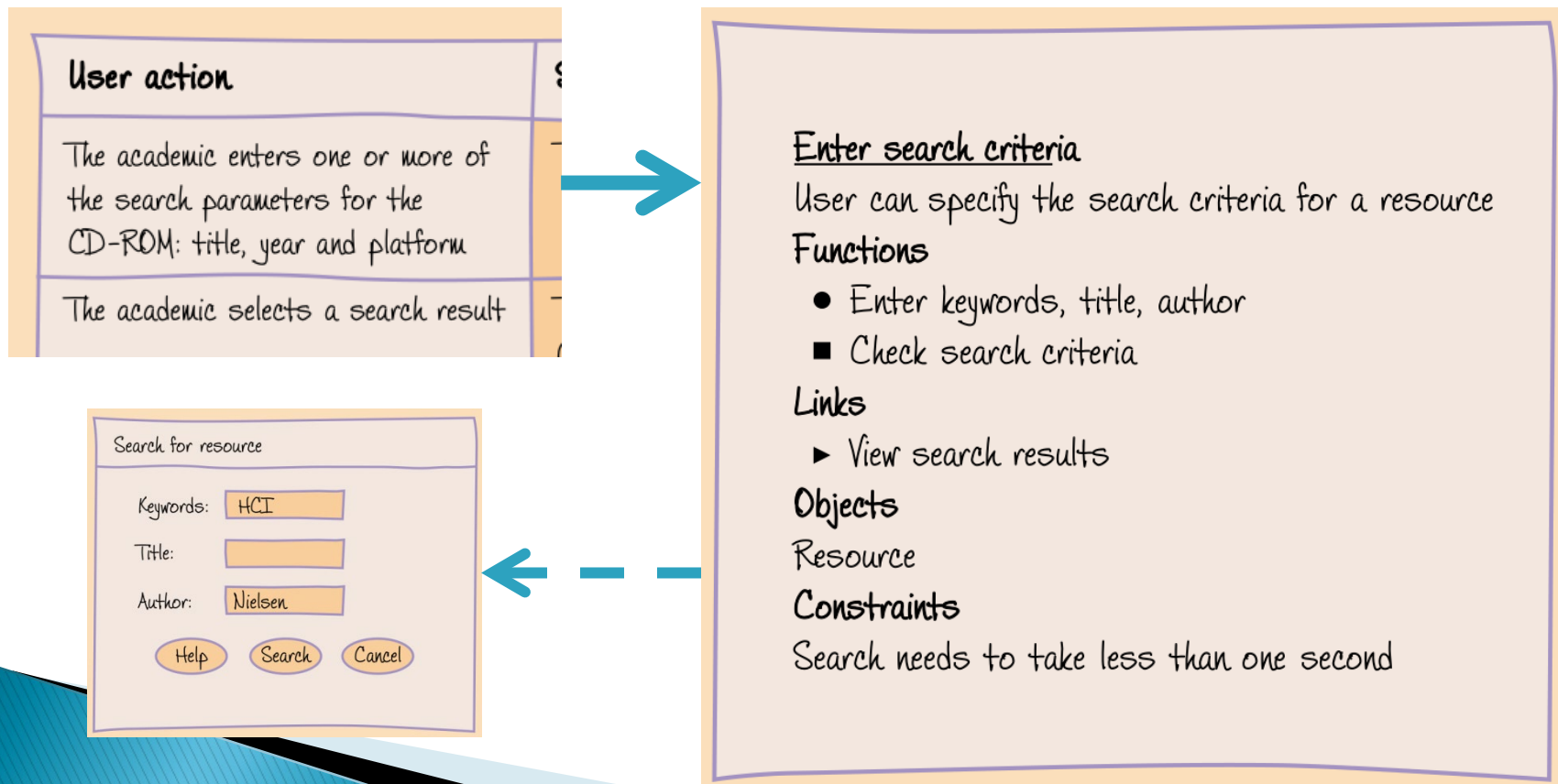
- They are derived from the concrete use cases (each use case shows the sequence of steps needed to accomplish a particular task)
 - That functionality can be divided between one or more containers

Conceptual Design: process

Identify the different containers and the task objects that go into each one

► Example:

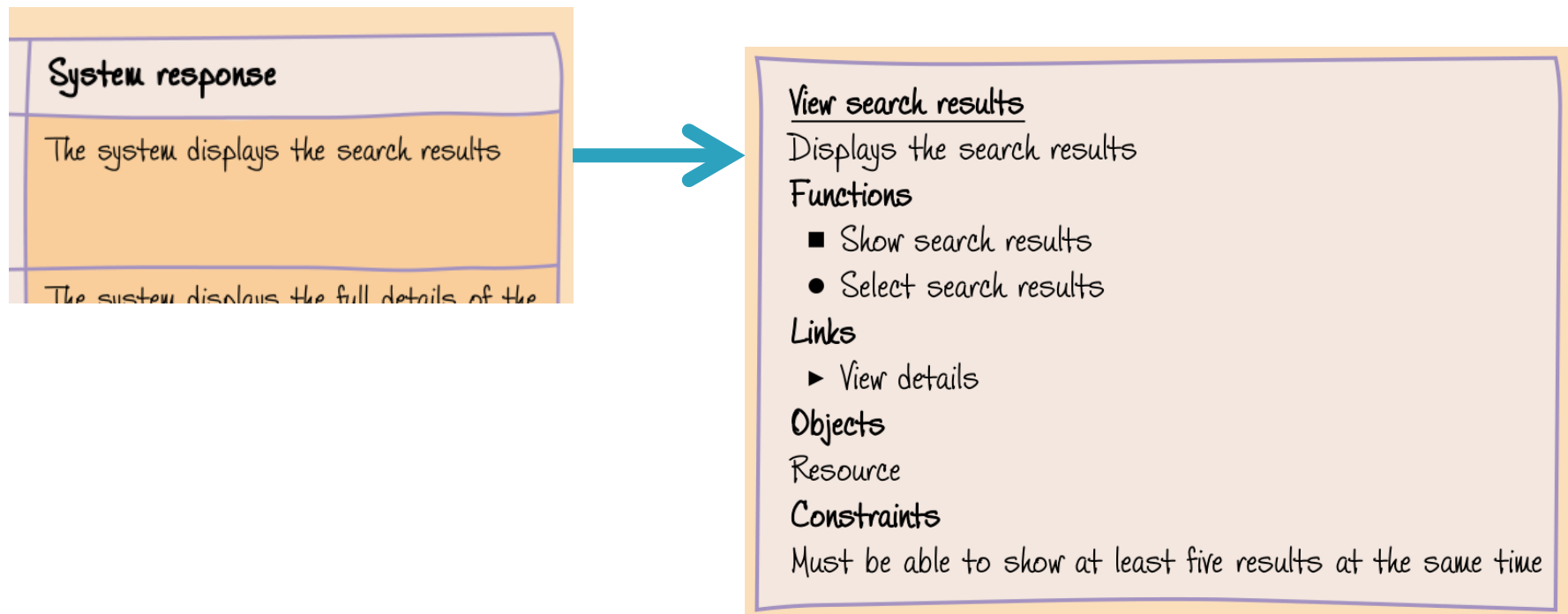
- If we use a single page for searching all resources:



Conceptual Design: process

Identify the different containers and the task objects that go into each one

▶ Another example



Conceptual Design: process

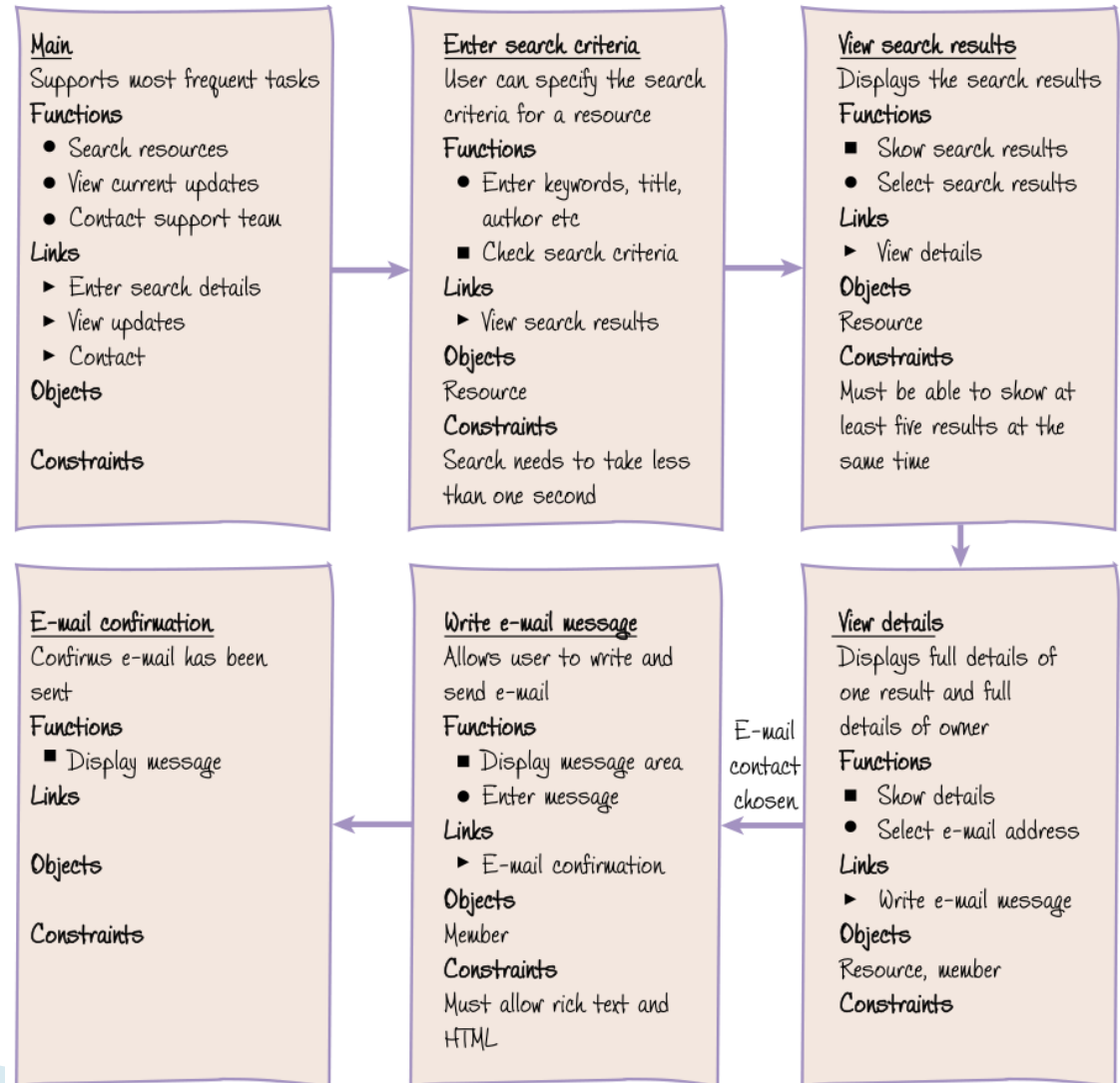
Link the containers to show the navigation flow

- ▶ Step 3: Link the containers to show the navigation flow
 - The links reflect the order of the actions in which the user perform some task
 - Usually identifying and linking the containers is done in parallel
 - Remember, the links can be:
 - > single-headed (the new container replaces the current)
 - >> double headed (both work in parallel)
 - The links can be labeled with conditions of interaction, that indicate a condition in which the flow can traverse the link
 - Links are represented in a GUI by a tool bar button, a link in a web page, etc.

Conceptual Design: process

Link the containers to show the navigation flow

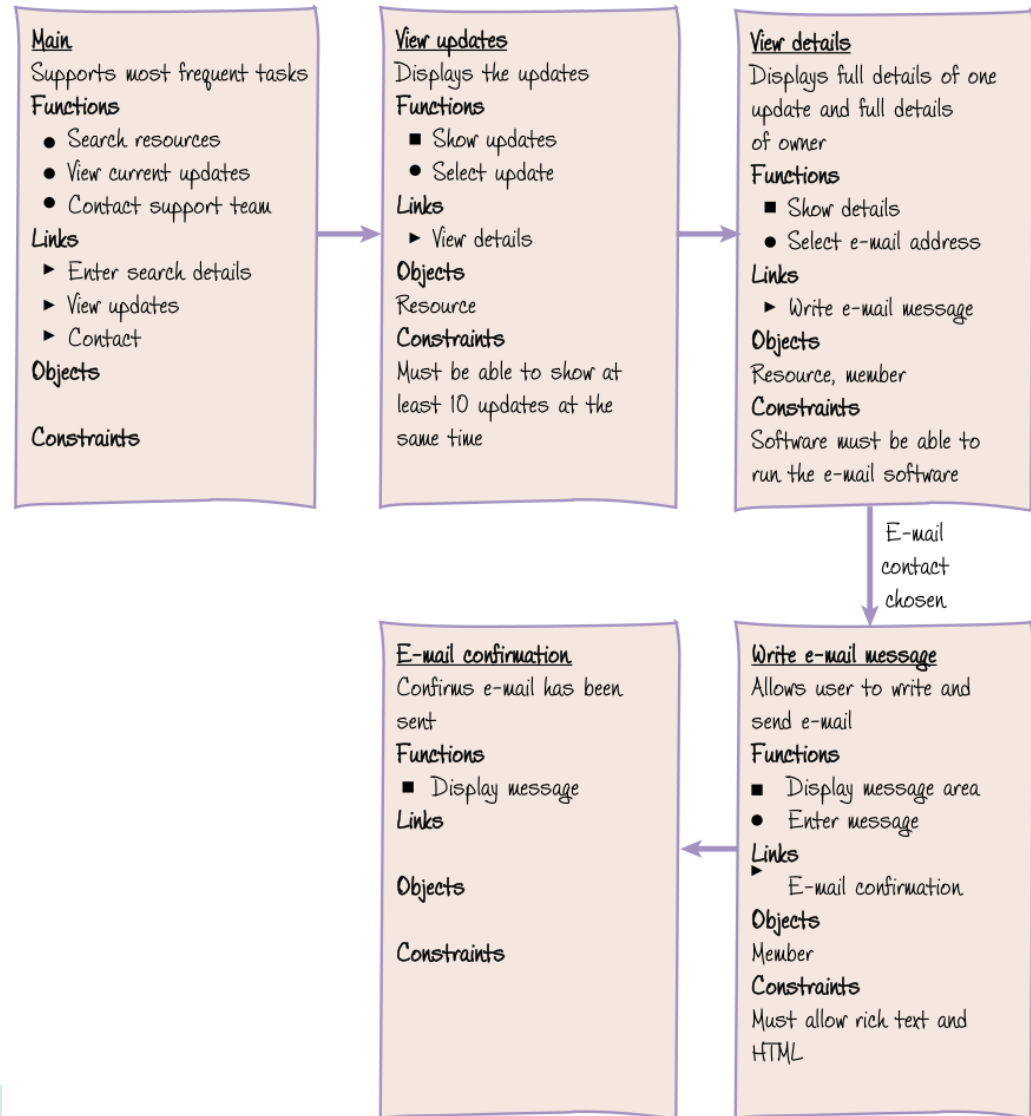
- ▶ Example:
section of the
content
diagram for
“Search and
request CD-
ROM”



Conceptual Design: process

Link the containers to show the navigation flow

- ▶ Example:
section of the
content
diagram for
“View updates
and request
book”



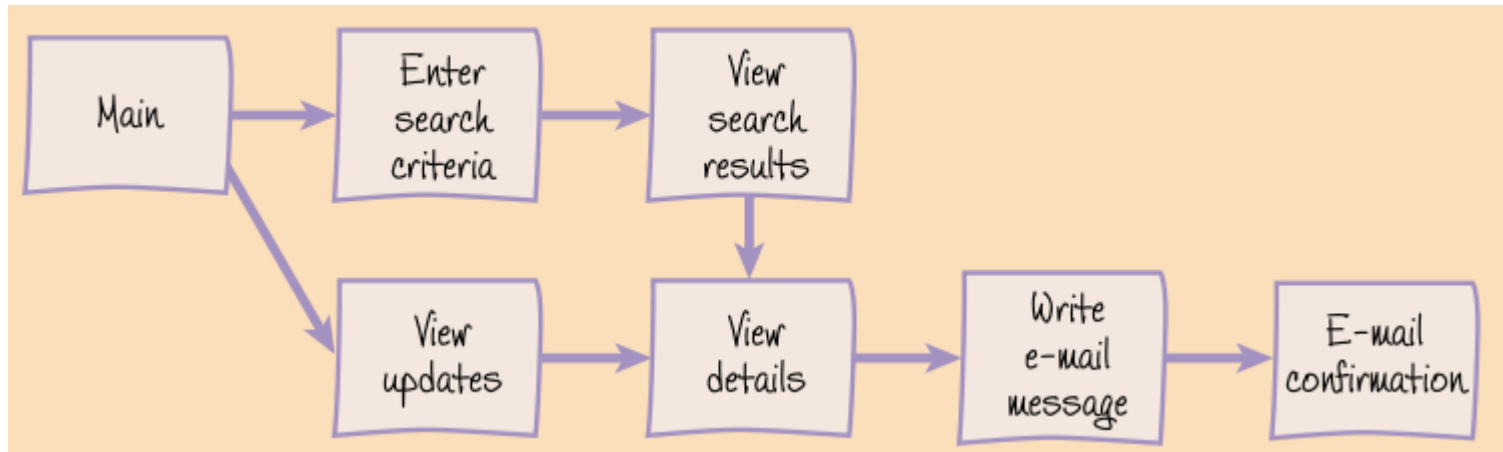
Conceptual Design: process

Link the containers to show the navigation flow

- ▶ Since creating a complete content diagram for a complex application is difficult, you should iterate over prototypes
- ▶ For evaluating the content diagram, you should step through some of the concrete use cases to ensure that the containers support the required functionality and the links allow the user to reach them

Conceptual Design: result

- ▶ Content diagram



- ▶ The translation of the containers to GUI elements is not one-to-one. A content diagram can be spread over several screens, or different containers can be combined in a single screen.
 - There are still a lot of decisions to be made.

References

- ▶ D. Stone, C. Jarrett, M. Woodroffe. User Interface Design and Evaluation. Morgan Kaufmann, 2005.
 - Chapter 3