

CS 494



Object-Oriented Analysis & Design

Requirements and Use Cases

© 2001 T. Horton

2/6/03 C-1

BTW... Specification Documents

- Steven McConnell (IEEE *Software*, Oct. 2000) says any of the following are called "requirements document":
 - Half-page summary of software *product vision*
 - Two-page key *features list*
 - 50-page list of details about end-user requirements (he calls this a *function-requirements document*)
 - 250-page exhaustive list of details about screens and GUI, input and input conditions, all system states and state changes, all persistent data, etc.
- This 4th item is what we usually mean by a Software Requirements Specification (SRS) document

2/6/03 C-4

Requirements

- Defining what the system should do
 - What the clients needs (as opposed to wants)
 - Not how the solution should be designed or implemented
- We recognize three iterative activities :
 - Elicitation: capturing information from sources
 - Documentation: "putting it on paper"
 - Validation: confirming it meets users' needs
- *Analysis* (or definition) versus *Specification*
 - Customer-oriented requirements
 - Developer-oriented requirements

2/6/03 C-2

Example: Automated Library System

- "Vision" Statement
 - You have been contracted to develop a computer system for a university library. The library currently uses a 1960s program, written in an obsolete language, for some simple bookkeeping tasks, and a card index, for user browsing. You are asked to build an interactive system which handles both of these aspects online.

2/6/03 C-5

Gathering Requirements

- Many sources of requirements
 - Interviews (structured vs. non-structured)
 - Stakeholder documents
 - Questionnaires
 - Existing or similar systems
 - Standards in that domain
 - Rapid prototypes
- Often information is documented in text
- From these build OO requirements models

2/6/03 C-3

Example: More Details on Needs

- Books and Journals
 - Library contains books and journals
 - Books may have several copies
 - Some books are short-term loans, all others 3 weeks
 - Only staff can borrow journals
 - Members can borrow up to 6 items at a time
 - Staff can borrow up to 12 items
 - New items arrive, old items are disposed of
 - Current year's journals are bound at year-end

2/6/03 C-6

Example: More Details... (cont'd)

- **Borrowing**
 - System keeps track of when items are borrowed and returned
 - Produce reminders when an item is over-due
 - (Future) Extend the loan of an item (if not reserved)
- **Browsing**
 - Users can search for a book by topic, by author, etc.
 - Users can check if item is available
 - Can reserve a book
 - Anyone can browse the library

2/6/03 C-7

Use Cases

- Each use case has a name
 - e.g. Borrow Copy of Book
- A family (or set, or class) of scenarios
 - A sequence of interactions
 - A set of different but related scenarios
- Documenting Use Cases
 - A UML Diagram showing all of them
 - Actors are stick-figures; use cases are ovals
 - For each use case define using English
 - A clear textual description
 - A set of scenarios in outline form

2/6/03 C-10

User-centered Requirements

- From such textual descriptions, could we build a system? How likely would it be to meet users' needs?
- Goal: Center system definition and development on user needs.
 - Identify various users of the system
 - Define what tasks they undertake with the system, and task outcomes
 - Document this in a way that can be used for requirements specification and later in the lifecycle

2/6/03 C-8

Example: Actors and Use Cases

- **Actors**
 - BookBorrower
 - JournalBorrower
 - Browser (person who browses, not SW)
 - Librarian
- **Use Cases**
 - Borrow copy of a book
 - Reserve a book
 - Return copy of book
 - Borrow journal
 - Browse
 - Update Catalog

2/6/03 C-11

Use Case Modeling

- Use Case:
 - “A sequence of actions a system performs to yield an observable result of value to a particular actor.”
 - Stevens/Pooley: A task which an actor needs to perform with the help of the system
- Actor:
 - Someone or something outside the system that interacts with the system
 - A user of the system in a particular role
- **Important:** We want an “external view” of the system

2/6/03 C-9

What Form Does a Use Case Take?

- We can *describe* Use Cases in a variety of ways
- First, text paragraphs
- Describes the Actors who participate with the system
- Describes the sequence of events

2/6/03 C-12

Forms of Such Descriptions

- Informal "Scenarios"
an informal narrative story, simple, 'natural', personal, not generalisable
- Use cases
—assume interaction with a system
—assume detailed understanding of the interaction
- Essential use cases
—abstract away from the details
—does not have the same assumptions as use cases

2/6/03 C-13

Example Template for Use Cases

- **Use case number or id:**
- **Use case title:**
- **Actors:**
- **Text description (a few sentences)**
- **Preconditions (if applicable):**
- **Flow of Events:**
- **Basic path:**
 1. *First step*
 2. *Second step*
 3. *etc*
- **Alternative Paths:**
 - *Name and short description (in words) of first alternative path/scenario.*
 - *Name and short description (in words) of 2nd alternative path/scenario.*
 - *etc.*
- **Postconditions (if applicable)**
- **Special conditions (if applicable).**

2/6/03 C-16

Example Text Description

- **Borrow copy of a book:**

A Bookborrower presents a copy of a book. The system checks that the s/he is a library member, and that s/he has not checked out too many books. If both checks succeed, then the system records that the member now as this copy of the book. Otherwise it refuses the loan.

2/6/03 C-14

Scenario for shared calendar

"The user types in all the names of the meeting participants together with some constraints such as the length of the meeting, roughly when the meeting needs to take place, and possibly where it needs to take place. The system then checks against the individuals' calendars and the central departmental calendar and presents the user with a series of dates on which everyone is free all at the same time. Then the meeting could be confirmed and written into people's calendars. Some people, though, will want to be asked before the calendar entry is made. Perhaps the system could email them automatically and ask that it be confirmed before it is written in."

2/6/03 C-17

What Else Is In a Use Case Description?

- **Pre- and Post-conditions**
 - Values of variables, system conditions, other use cases etc.
- **Normal vs. alternative behavior**
 - Can be shown in the text description (somehow)
 - Exceptions vs. acceptable alternatives

2/6/03 C-15

Use case for shared calendar

1. The user chooses the option to arrange a meeting.
2. The system prompts user for the names of attendees.
3. The user types in a list of names.
4. The system checks that the list is valid.
5. The system prompts the user for meeting constraints.
6. The user types in meeting constraints.
7. The system searches the calendars for a date that satisfies the constraints.
8. The system displays a list of potential dates.
9. The user chooses one of the dates.
10. The system writes the meeting into the calendar.
11. The system emails all the meeting participants informing them of their appointment

2/6/03 C-18

Alternative paths for shared calendar

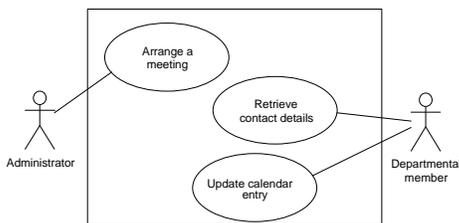
- Step 5. If the list of people is invalid, then:
- 5.1 The system displays an error message.
 - 5.2 The system returns to step 2.
- Step 8. If no potential dates are found, then:
- 8.1 The system displays a suitable message.
 - 8.2 The system returns to step 5.

2/6/03 C-19

The following slides not covered in class (yet)

2/6/03 C-22

Example use case diagram for shared calendar



2/6/03 C-20

Relationships between Use Cases

- UML supports two relationships between two use cases
 - <<includes>> and <<extends>>
 - Note: before UML 1.3 <<includes>> was <<uses>>

2/6/03 C-23

Essential use-case for *arrangeMeeting*

Use-case Name: *arrangeMeeting*

USER INTENTION	SYSTEM RESPONSIBILITY
1. arrange a meeting	2. request meeting attendees & constraints
3. identify meeting attendees & constraints	4. search calendars for suitable dates
	5. suggest potential dates
6. choose preferred date	7. book meeting

2/6/03 C-21

<<includes>> in Use Cases

- Meaning:
 - The source use case always includes the actions specified in the target use case

2/6/03 C-24

<<extends>> in Use Cases

- **Meaning:**
 - The target use case may include the behavior of the source use case

2/6/03 C-25

FYI... Extending UML

- Possible to extend the “vocabulary” of UML
- Creates a new kind of building block
 - Derived from existing UML feature
 - But specific for current problem
- Pre-defined and user-defined stereotypes
- UML even allows you to provide a new icon!
- Syntax: Above name add <<stereotype>> inside guillemets (French quotes)
- Again, used to provide extra info about some UML modeling construct

2/6/03 C-26