Requirements Engineering: Usage Models

CECS 590

Recap time!

- Domain Model
  - What is it?
  - Why do we need it?
  - Which elements does it contain?
  - How to elaborate it?

Visit by Intel: Sarah Gregory

- Next Monday 23rd of March
- 20+ years of experience
- RE specialist and trainer
- Works world-wide
- Active in research

Requirements Engineering – Outline

- WHY do we need Requirements Engineering and what is it?
- Principles: Definitions, process, roles, problem/solution view, artifact orientation
- System Models: Decomposition and abstraction, system views
- Frameworks: What reference structures can I use for requirements?
- Business Case Analysis: Why are we building this system?
- Stakeholders: Who are the people to talk to about requirements?
- Goals and Constraints: What are the major objectives for the system?
- System Vision: What exactly do we want to achieve?
- Domain Models: What are the surrounding systems ours interacts with?
- **Usage Models: How will the system interact with the user?**
- Software quality models: How to determine the quality characteristics?
- Quality requirements: How to specify which qualities need to be met?
- Process requirements: How to specify constraints for development?
- Towards a system specification: How to hand over to design?
- Quality assurance: How to ensure that RE is done in a good way?
- Change management: How to evolve requirements?

Usage Models

- Definition
- Use cases
- Scenarios
- Relation
- Cockburn template
- Elaboration

K Rayker, stock.xchng

Usage Model

- Def.: A usage model describes the system behavior form the point of view of the user („Black box“) by modeling interaction sequences.
- It specifies the use cases (from the system vision)
- Why? Understanding of intended uses by means of the system.
- Notations:
  - Use case overview diagram
  - Structured text (templates)
  - UML activity diagrams
  - Message Sequence Charts

Use cases & Scenarios

- Def.: A use case is a series of system events triggered by an actor that leads to results for the actor.
- Def.: A scenario is an ordered set of interactions between partners, usually a system and a group of external actors.
- A Usage Model in AMDiRE has three parts:
  - Use Case Overview Diagram („bubble“ diagram)
  - Use Case Templates (one per „bubble“)
  - Scenario diagrams (one per use case template)

Use Case Overview Diagram

Another Use Case Overview Diagram

Elaboration of a Usage Model

- Deducing the Use Cases
  - Identification of Use Case by business processes
  - Analysis of business processes
    - Task distribution to actors
    - Identification of usage functions
    - Definition of the role of the system, e.g.:
      - Passive support (data administration),
      - Active support (task performance)
  - Informal start: What are the system features?
- Stepwise description and refinement of the scenarios and their interaction
  - Focus on analysis and modeling of
    - Information flow (for later data modeling)
    - Interaction and control flow at the system border

Exercise: ATM

- Use case overview
- Use case template
- Scenario

[Scott W. Ambler 2007]

USE CASE #		< the name is the goal as a short active verb phrase>
Goal in Context		<a longer statement of the goal in context if needed>
Scope & Level		<what system is being considered black box under design> <one of : Summary, Primary Task, Subfunction>
Preconditions		<what we expect is already the state of the world>
Success End Condition		<the state of the world upon successful completion>
Failed End Condition		<the state of the world if goal abandoned>
Primary,		<a role name or description for the primary actor>.
Secondary Actors		<other systems relied upon to accomplish use case>
Trigger		<the action upon the system that starts the use case>
DESCRIPTION	Step	Action
	1	<put here the steps of the scenario from trigger to goal delivery,and any cleanup afte>
	2	<...>
	3	<...>
EXTENSIONS	Step	Branching Action
	1a	<condition causing branching> : <action or name of sub.use case>
SUB-VARIATIONS		Branching Action
	1	<list of variation s>
RELATED INFORMATION		<Use case name>
Priority:		<how critical to your system / organization>
Performance		<the amount of time this use case should take>
Frequency		<how often it is expected to happen>
Channels to actors		<e.g. interactive, static files, database, timeouts>
OPEN ISSUES		<list of issues awaiting decision affecting this use case >
Due Date		<date or release needed>
...any other management information...		<...as needed>
Superordinates		<optional, name of use case(s) that includes this one>
Subordinates		<optional, depending on tools, links to sub.use cases>

ATM Use Case Overview

Relation: Use Cases and Scenarios

- For each „bubble“ in the overview diagram:
- Use Cases summarize a set of scenarios to a specific usage of the system.
  - Use Case:
    Task, objective, causal relation (pre- and post-conditions)
  - Scenario:
    Sequence of Events (steps, events, interaction)

Iterative Elaboration

(compare to refinement and abstraction of goals in the earlier lecture)

(1) Cluster scenarios to tasks

(2) Elicit task-specific scenarios, analyse and walk through them

Use cases & Scenarios: Cockburn template

- **Use**: Use cases and scenarios complement each other.
- **Techniques**: Structured text and/or sequence/ interaction diagrams
- **Elicitation**: iterative; combine scenarios to tasks, „play out“ task-specific scenarios and analyse

USE CASE #	< the name is the goal as a short active verb phrase>	
Goal in Context	<a longer statement of the goal in context if needed>	
Scope & Level	<what system is being considered black box under design> <one of : Summary, Primary Task, Subfunction>	
Preconditions	<what we expect is already the state of the world>	
Success End Condition	<the state of the world upon successful completion>	
Failed End Condition	<the state of the world if goal abandoned>	
Primary,	<a role name or description for the primary actor>.	
Secondary Actors	<other systems relied upon to accomplish use case>	
Trigger	<the action upon the system that starts the use case>	
DESCRIPTION	Step	Action
	1	<put here the steps of the scenario from trigger to goal delivery,and any cleanup afte>
	2	<...>
	3	<...>
EXTENSIONS	Step	Branching Action
	1a	<condition causing branching> : <action or name of sub.use case>
SUB-VARIATIONS		Branching Action
	1	<list of variation s>
RELATED INFORMATION		<Use case name>
Priority:		<how critical to your system / organization>
Performance		<the amount of time this use case should take>
Frequency		<how often it is expected to happen>
Channels to actors		<e.g. interactive, static files, database, timeouts>
OPEN ISSUES		<list of issues awaiting decision affecting this use case >
Due Date		<date or release needed>
...any other management information...		<...as needed>
Superordinates		<optional, name of use case(s) that includes this one>
Subordinates		<optional, depending on tools, links to sub.use cases>

Example

Use Case: **Smoke detection**

CHARACTERISTIC INFORMATION

Goal in Context: To inform stakeholders of the fire in the house.

Scope: Alarm system.

Level: Primary task

Preconditions: Alarm system is armed and active. Detector is working. Communication means are functioning.

Success End Condition: Stakeholder is informed.

Failed End Condition: Stakeholder are not informed of smoke. Fire destroys monitored property.

Primary Actor: **Smoke detector.**

Trigger: Detection of smoke.

MAIN SUCCESS SCENARIO

<put here the steps of the scenario from trigger to goal delivery, and any cleanup after>

1. One of the smoke detector signals smoke presence.
2. System identifies smoke detector location by its comm. port.
3. System informs stakeholders via phone line and the a/v speaker.

RELATED INFORMATION (optional)

Priority: Top priority.

Performance Target: Stakeholders should be notified within 5 seconds.

Frequency: Rarely. Only in extreme cases of fire, or strong smoke concentration.

Subordinate Use Cases: **Notify Stakeholders**

Channel to primary actor: Simplex, one way, Electric wire.

Secondary Actors: Stakeholders – Authorities and Owners

Channel to Secondary Actors: Phone line, Speaker

OPEN ISSUES (optional)

1. How the system will recognize that someone is cooking food that generates smoke.

SCHEDULE

Due Date: Version 1.0 release.

Exercise: ATM

- Use case overview
- Use case template
- Scenario

[Scott W. Ambler 2007]

USE CASE #		< the name is the goal as a short active verb phrase>
Goal in Context		<a longer statement of the goal in context if needed>
Scope & Level		<what system is being considered black box under design> <one of : Summary, Primary Task, Subfunction>
Preconditions		<what we expect is already the state of the world>
Success End Condition		<the state of the world upon successful completion>
Failed End Condition		<the state of the world if goal abandoned>
Primary,		<a role name or description for the primary actor>.
Secondary Actors		<other systems relied upon to accomplish use case>
Trigger		<the action upon the system that starts the use case>
DESCRIPTION	Step	Action
	1	<put here the steps of the scenario from trigger to goal delivery,and any cleanup afte>
	2	<...>
	3	<...>
EXTENSIONS	Step	Branching Action
	1a	<condition causing branching> : <action or name of sub.use case>
SUB-VARIATIONS		Branching Action
	1	<list of variation s>
RELATED INFORMATION		<Use case name>
Priority:		<how critical to your system / organization>
Performance		<the amount of time this use case should take>
Frequency		<how often it is expected to happen>
Channels to actors		<e.g. interactive, static files, database, timeouts>
OPEN ISSUES		<list of issues awaiting decision affecting this use case >
Due Date		<date or release needed>
...any other management information...		<...as needed>
Superordinates		<optional, name of use case(s) that includes this one>
Subordinates		<optional, depending on tools, links to sub.use cases>

Template

- **Use Case:** <number> <the name should be the goal as a short active verb phrase>
- CHARACTERISTIC INFORMATION
  - Goal in Context: <a longer statement of the goal, if needed>
  - Scope: <what system is being considered black-box under design>
  - Level: <one of: Summary, Primary task, Subfunction>
  - Preconditions: <what we expect is already the state of the world>
  - Success End Condition: <the state of the world upon successful completion>
  - Failed End Condition: <the state of the world if goal abandoned>
  - Primary Actor: <a role name for the primary actor, or description>
  - Trigger: <the action upon the system that starts the use case, may be time event>
- MAIN SUCCESS SCENARIO
  - <put here the steps of the scenario from trigger to goal delivery, and any cleanup after>
  - <step #> <action description>
- EXTENSIONS
  - <put here there extensions, one at a time, each refering to the step of the main scenario>
  - <step altered> <condition> : <action or sub.use case>
  - <step altered> <condition> : <action or sub.use case>
- SUB-VARIATIONS
  - <put here the sub-variations that will cause eventual bifurcation in the scenario>
  - <step or variation # > <list of sub-variations>
  - <step or variation # > <list of sub-variations>
- RELATED INFORMATION (optional)
  - Priority: <how critical to your system / organization>
  - Performance Target: <the amount of time this use case should take>
  - Frequency: <how often it is expected to happen>
  - Superordinate Use Case: <optional, name of use case that includes this one>
  - Subordinate Use Cases: <optional, depending on tools, links to sub.use cases>
  - Channel to primary actor: <e.g. interactive, static files, database>
  - Secondary Actors: <list of other systems needed to accomplish use case>
  - Channel to Secondary Actors: <e.g. interactive, static, file, database, timeout>
- OPEN ISSUES (optional)
  - <list of issues about this use cases awaiting decisions>
- SCHEDULE
  - Due Date: <date or release of deployment>

Example Use Case ATM

- **Use Case:** 1 withdraw money
- CHARACTERISTIC INFORMATION
  - Goal in Context: user withdraws money from the ATM
  - Scope: ATM
  - Level: Primary task
  - Preconditions: user has an ATM card and has access to ATM
  - Success End Condition: user gets money
  - Failed End Condition: user doesn't get money
  - Primary Actor: customer (= user)
  - Trigger: ATM card entered by user
- MAIN SUCCESS SCENARIO
  1. User enters card
  2. System prompts for PIN
  3. User enters PIN
  4. System prompts options for withdrawal / transfer / deposit money
  5. User selects withdraw
  6. System prompts for amount
  7. User enters amount
  8. System returns money
- EXTENSIONS
  - 5. *condition* selection of different account: *action* Withdraw from different account
  - <step altered> <condition> : <action or sub.use case>
  - <step altered> <condition> : <action or sub.use case>
- SUB-VARIATIONS
  - 4. *condition* user entered wrong PIN: *action* system displays error message
  - 8. not enough money: system displays error message
  - <step or variation # > <list of sub-variations>
- RELATED INFORMATION (optional)
  - Priority: critical
  - Performance Target: one minute
  - Frequency: very often (depends on location of ATM)
  - Superordinate Use Case: <optional, name of use case that includes this one>
  - Subordinate Use Cases: <optional, depending on tools, links to sub.use cases>
  - Channel to primary actor: interactive
  - Secondary Actors: <list of other systems needed to accomplish use case>
  - Channel to Secondary Actors: <e.g. interactive, static, file, database, timeout>
- OPEN ISSUES (optional)
  - <list of issues about this use cases awaiting decisions>
- SCHEDULE
  - Due Date: May 2014

Use Cases vs. Scenarios vs. Functional Requirements

Use Case: Set of all scenarios

Scenario: Concrete interaction sequence

Functional requirement (roughly):
A (system-)reaction to a stimulus

Use Case

USE CASE #	<the name is the goal as a short active verb phrase>	
Context of Use	<a longer statement of the context of use if needed>	
Scope	<what system is being considered black box under design>	
Level	<one of summary, primary task, subfunction>	
Primary Actor	<a role name for the primary actor, or a description>	
Stakeholder and Interests	Stakeholder	Interest
	<stakeholder name>	<put here the interest of the stakeholder>
	<stakeholder name>	<put here the interest of the stakeholder>
Preconditions	<what we expect is already the state of the world>	
Minimal Guarantees	<the interests as protected on any exit>	
Success Guarantees	<the interests as satisfied on a successful ending>	
Trigger	<the action upon the system that starts the use case>	
Description	Step	Action
	1	<put here the steps of the scenario from trigger to goal delivery and any cleanup after>
	2	<...>
	3	
Extensions	Step	Branching Action
	1a	<condition causing branching> : <action or name of sub use case>
Technology and Data Variations		
	1	<list of variations>

Scenario

Functional requirement

Example: Scenario Purchase Item

Def.: A scenario is an ordered set of interactions between partners, usually a system and a group of external actors.

How-to: Scenarios in a diagram

- UML Sequence Diagram: for user interaction
- UML Activity Diagram: for system-internal interaction

→ Depends on your system what works better.

Sequence Diagram

FIGURE A1.7

A sample sequence diagram

Sequence Diagram: frame

An empty UML 2 frame element

http://www.ibm.com/developerworks/rational/library/3101.html

Sequence Diagram: in/out messages

A sequence diagram that has incoming and outgoing messages

http://www.ibm.com/developerworks/rational/library/3101.html

Sequence Diagram: Lifelines

An example of the Student class used in a lifeline whose instance name is freshman

http://www.ibm.com/developerworks/rational/library/3101.html

Sequence Diagram: Messages

An example of messages being sent between objects, including return messages

http://www.ibm.com/developerworks/rational/library/3101.html

Sequence Diagram: help method

The system object calling its determineAvailableReports method

http://www.ibm.com/developerworks/rational/library/3101.html

Sequence Diagram: asynch/synch

A sequence diagram fragment showing an asynchronous message being sent to instance2

http://www.ibm.com/developerworks/rational/library/3101.html

Sequence Diagram: guard

A segment of a UML 1.x sequence diagram in which the addStudent message has a guard

http://www.ibm.com/developerworks/rational/library/3101.html

Sequence Diagram: alternative

A sequence diagram fragment that contains an alternative combination fragment

http://www.ibm.com/developerworks/rational/library/3101.html

Sequence Diagram: option

A sequence diagram fragment that includes an option combination fragment

http://www.ibm.com/developerworks/rational/library/3101.html

Sequence Diagram: loop

An example sequence diagram with a loop combination fragment

http://www.ibm.com/developerworks/rational/library/3101.html

Sequence Diagram: reference

A sequence diagram that references two different sequence diagrams

http://www.ibm.com/developerworks/rational/library/3101.html

Collaboration "Rent Item", scenario "unavailable"
c:Customer
:Employee
:Rental
i:Item
r:Reservation
rent(this, i, p)
new(c, i, p)
isValid()
alt
[adult movie]
isRentableTo(c)
isAdult()
see business rule BR_IRC
true
true
[else]
isRentableTo(c)
true
isRentableDuring(p)
*[all] getPeriod()
period
false
false
see business rule BR_IRP
getReasonInvalid()
"unavailable"
cannotRent("unavailable")

Exercise: ATM

- Use case overview
- Use case template
- Scenario

[Scott W. Ambler 2007]

USE CASE #		< the name is the goal as a short active verb phrase>
Goal in Context		<a longer statement of the goal in context if needed>
Scope & Level		<what system is being considered black box under design> <one of : Summary, Primary Task, Subfunction>
Preconditions		<what we expect is already the state of the world>
Success End Condition		<the state of the world upon successful completion>
Failed End Condition		<the state of the world if goal abandoned>
Primary,		<a role name or description for the primary actor>.
Secondary Actors		<other systems relied upon to accomplish use case>
Trigger		<the action upon the system that starts the use case>
DESCRIPTION	Step	Action
	1	<put here the steps of the scenario from trigger to goal delivery,and any cleanup afte>
	2	<...>
	3	<...>
EXTENSIONS	Step	Branching Action
	1a	<condition causing branching> : <action or name of sub.use case>
SUB-VARIATIONS		Branching Action
	1	<list of variation s>
RELATED INFORMATION		<Use case name>
Priority:		<how critical to your system / organization>
Performance		<the amount of time this use case should take>
Frequency		<how often it is expected to happen>
Channels to actors		<e.g. interactive, static files, database, timeouts>
OPEN ISSUES		<list of issues awaiting decision affecting this use case >
Due Date		<date or release needed>
...any other management information...		<...as needed>
Superordinates		<optional, name of use case(s) that includes this one>
Subordinates		<optional, depending on tools, links to sub.use cases>

ATM Scenario in UML Sequence diagram

Assignment Usage Model

Usage model assignment is online

(due Friday March 27)

Monday: Intel.

Wednesday: No class. So, meet in your teams and get assignments done.