Module 1-7(a)

Model-driven Software Development (MDSE) for the Cloud

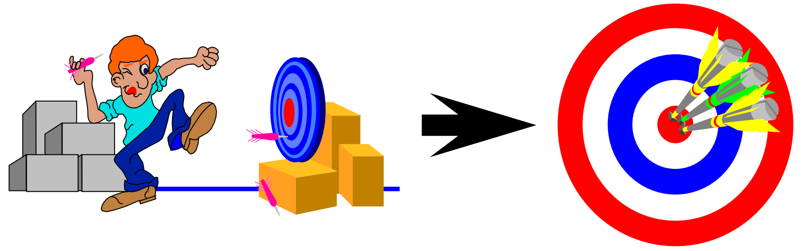
Use Case Modelling – How to capture your requirements as Use Cases

Outline

- Roadmap
- Recommended workflow
- Tasks and corresponding Work products
  - UML use cases
  - Examples
  - Guidelines
- References

Practice: Use case modelling

- Purpose
  - To define system functionality requirements as use cases

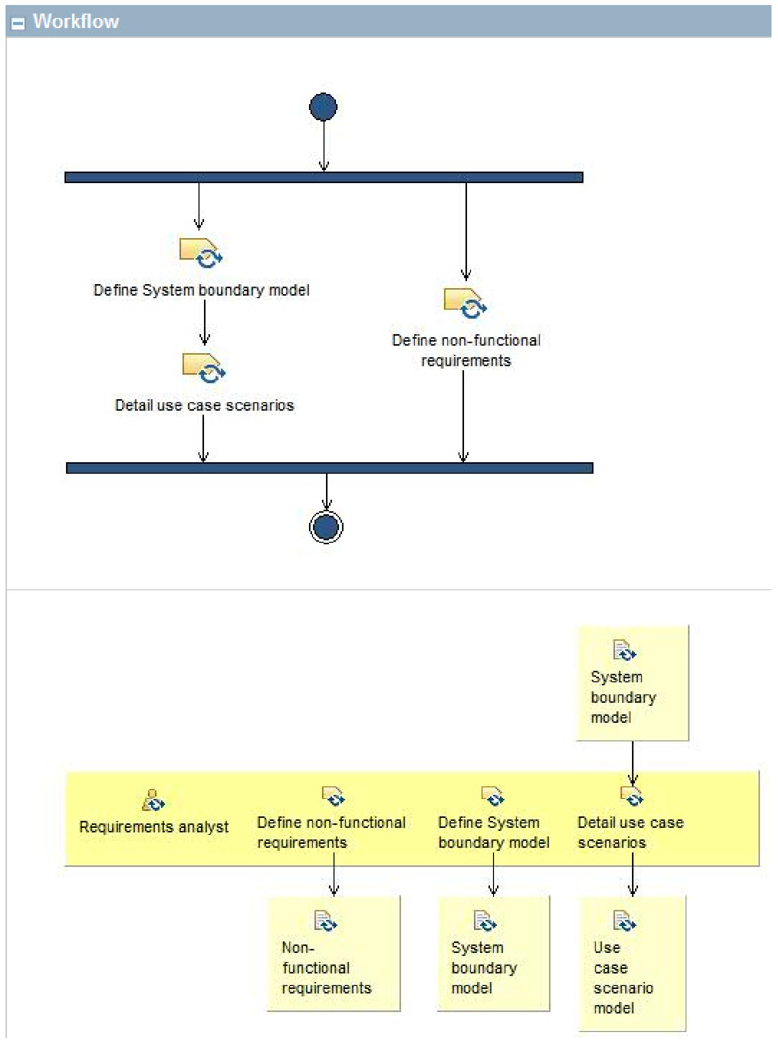


- Contents
  - How to adopt the use case modelling practice
  - Work Products
    - Non-functional requirements
    - System boundary model
    - Use case scenario model
  - Tasks
    - Define non-functional requirements
    - Define System boundary model
    - Detail use case scenarios
  - Use case modelling
  - Requirements analyst
  - Stakeholder
  - Use case template

Roadmap: How to adopt the Use case modelling exercise

- The Requirements model identifies the system requirements. These include functional requirements, Non-functional requirements (quality of service) and constraints.
- Non-functional requirements are statements concerning performance, availability, security, reliability and so forth. There are also other requirements specifying constraints that will have impact on the system to be developed, for instance available resources, special customer preferences, company strategy etc.
- Use cases and scenario descriptions are used to capture the functional requirements. These use cases and scenario descriptions are also used to structure some of the Non-functional requirements and constraints. Figure 1 shows the work products of the Requirements model.

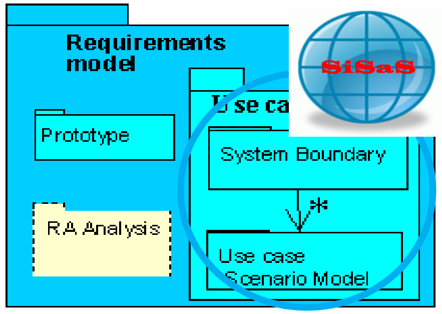
Recommend workflow: Use case modelling


Workflow
Define System boundary model
Define non-functional requirements
Detail use case scenarios
System boundary model
Requirements analyst
Define non-functional requirements
Define System boundary model
Detail use case scenarios
Non-functional requirements
System boundary model
Use case scenario model

Types of requirements

- Functional requirements
- Non-functional requirements. May be added to the functional models or defined separately.
  - Performance, interoperability etc.
- Other requirements towards the
  - technologies and development environment’s ease of use
  - Maturity of tools (performance, consistency, scalability etc.)
  - Compatibility with standards or other software
  - Cost (open source or licenses)
  - User groups and vendor support (now and future)

Requirements Model



- The Use Case Model describes the system in terms of
  - Actors
  - use cases
  - scenario descriptions
- It is defined a use case template to be used as a vehicle for developing the use case model.
- Non functional requirements are part of the use case model as these kinds of requirements are associated with use cases according to the use case template.
- General non functional requirements that applies for the whole system are associated with the system boundary which is also included in the use case model.

Purpose of Requirements Model

- Capture system requirements
- Specify system behavior as “Use Cases”
- Set the system boundary
- Specify system environment as “Actors”
- Describe interactions between System and Actors
- Drive system development process

- A Use Case specifies a sequence of actions, including variants, that the system can perform and that yields an observable result of value to a particular actor.

A Use Case is

- User visible
- User-meaningful
- Easy for a user to confirm (or change)
- Precise enough as specification

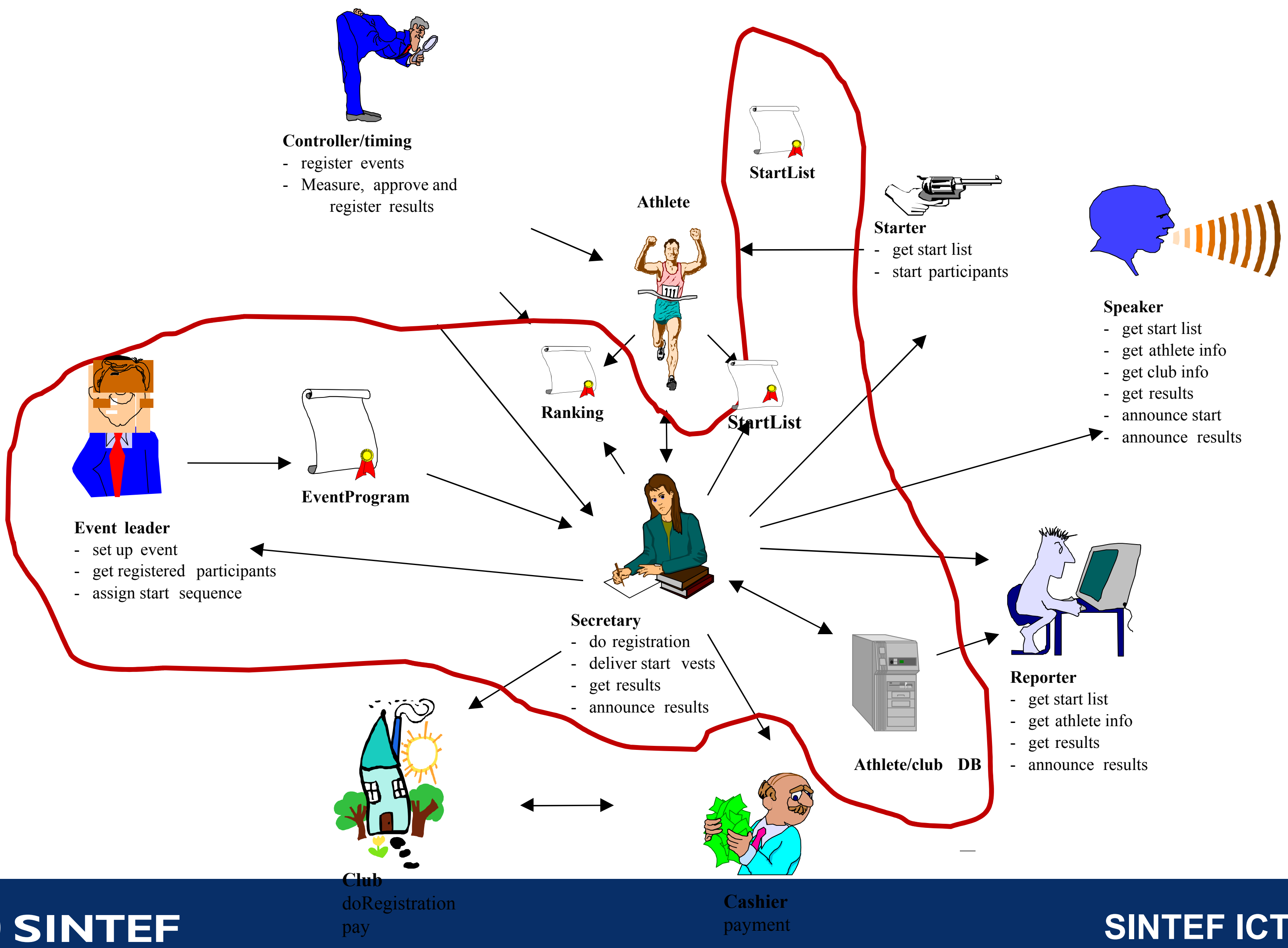
System Boundary Model

- Goal
  - Attain a common understanding of the system content and its purpose
    - Identify and describe the system boundary (AoC)
    - Identify and describe the external actors and their responsibilities
    - Identify and describe the actors main service requirements
- Deliverables
  - Use case diagram
    - System Boundary
    - High-level requirements

Developing a system boundary model

- Information gathering from
  - Rich picture, goal model BIM and BPM
  - Users and domain experts
    - Users playing the roles denoted by identified actors
  - Existing system/competing system
- Identify the system boundary
  - What is Inside the system: **Use case(s)** (this is what to create)
  - What is outside the system: **Actors** (this is what to interface with)

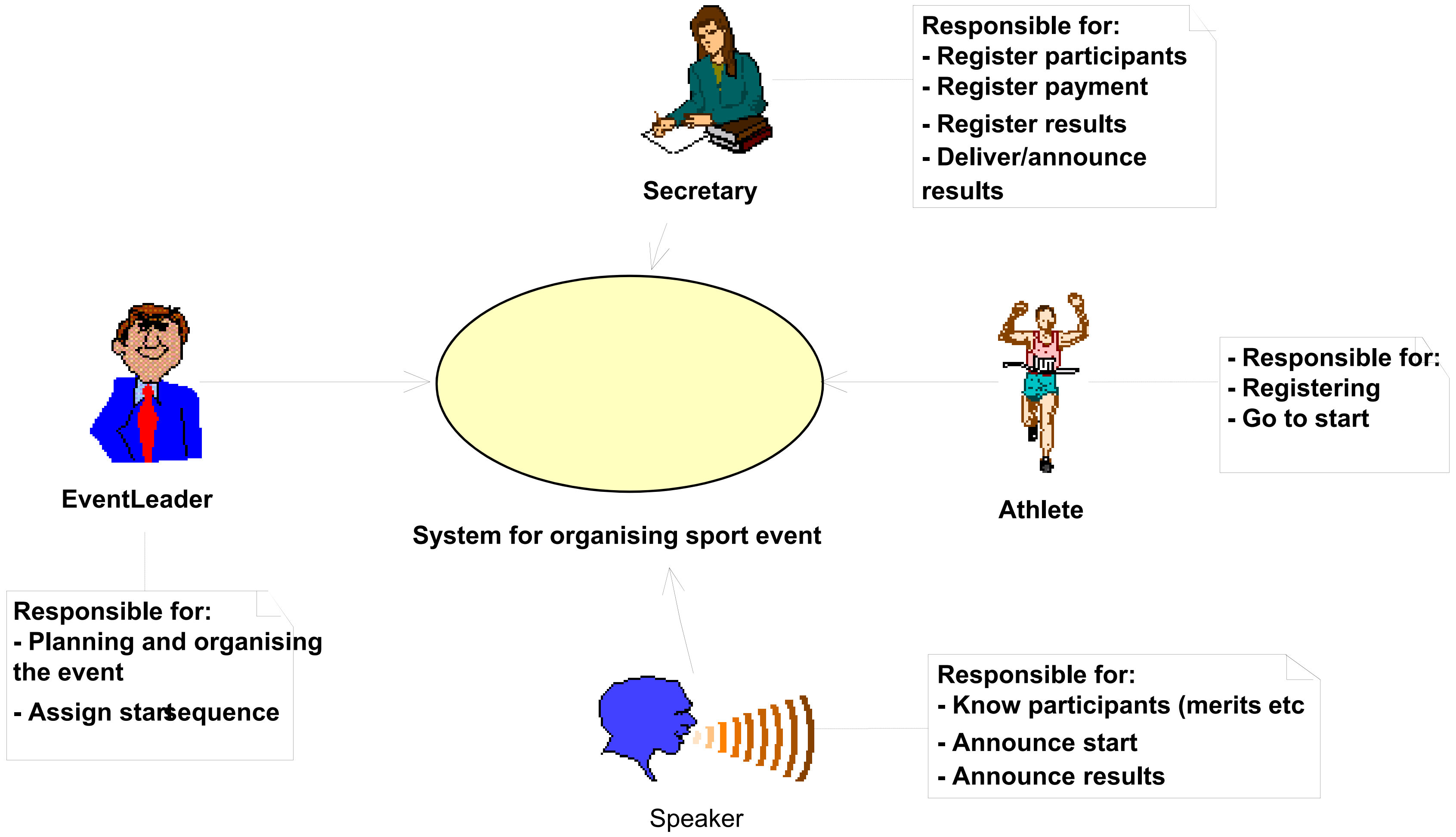
Example, using rich picture



Identifying actors

- Ask:
  - Who uses the system?
  - Who maintains the system?
  - What other systems use this system?
  - Who gets information from the system?
  - Who provides information to the system?
  - Does anything happen automatically at preset times?
  - Who starts and shut down the system?
- Examples: persons/roles, software systems, things/objects (hardware) networks, etc.
- Describe the actors responsibilities (short)

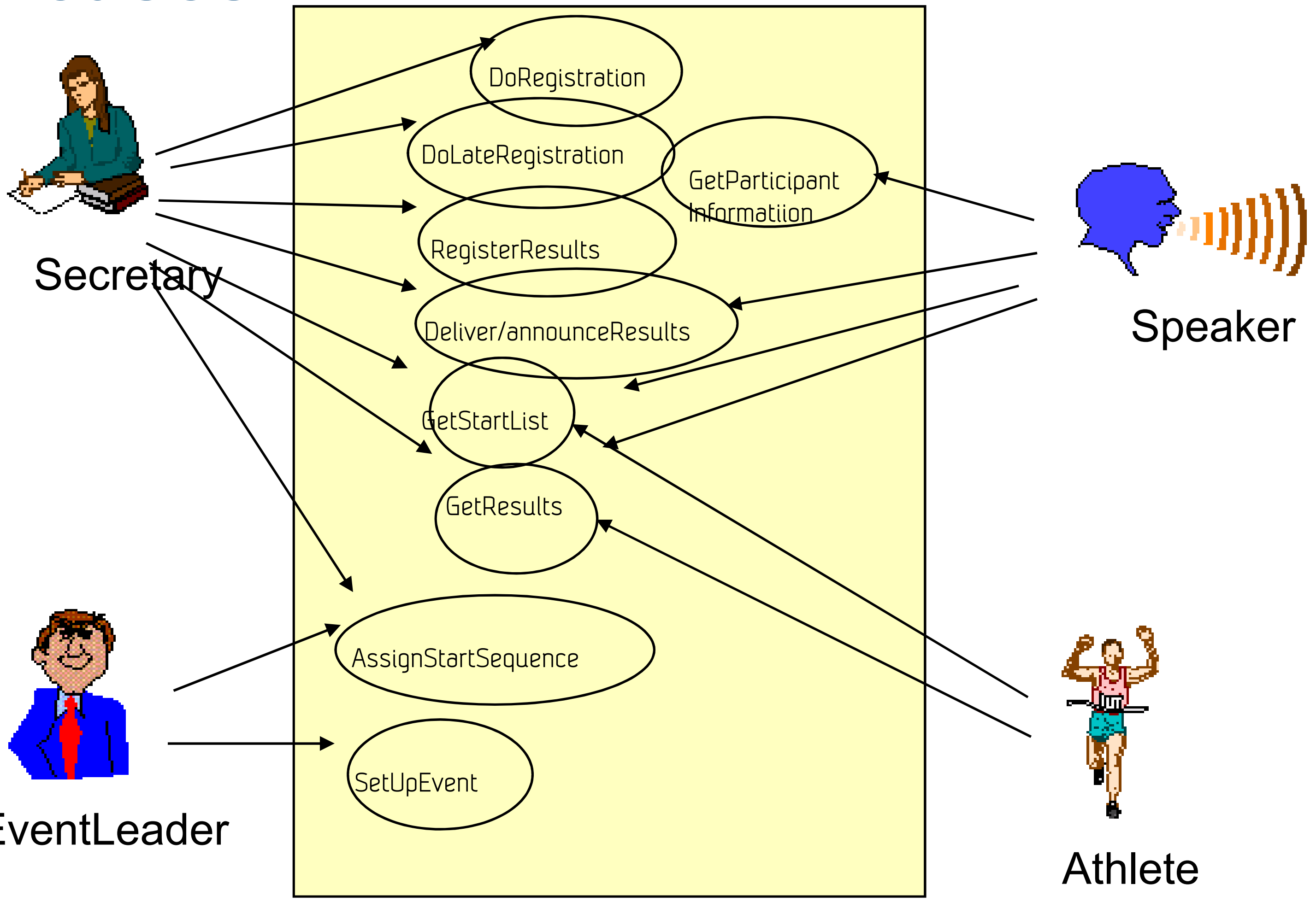
Example: Sport event organizing system SBM



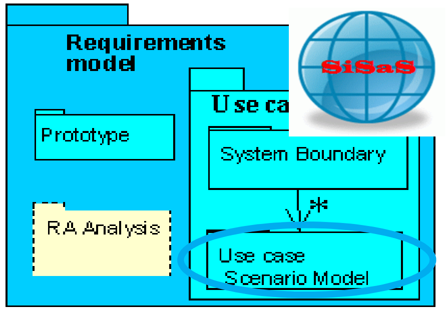
Identify use cases

- Go through the list of actors and identify what services they need to fulfil their obligations
- Answer the following Questions
  - What services do the actor requires from the system?
  - Which events will the actor initiate and which events will the actor be interested in?
  - Which events/notifications occur? (the notifications reaching the actor independent of his interest (e.g. error notifications)

Example: Sport event organizing system use cases



System for organising sport event



Use case Scenario Model

- Description
  - The Use Case Scenario Model digs into the identified use cases and describes these in detail.
  - A use case template is used as a vehicle for this detailing.
- Goal
  - Capture and understand the requirements.
- Deliverables
  - Detailed UML use case diagram and descriptions in accordance to Use case template
- Activities
  - Fill in Use Case template

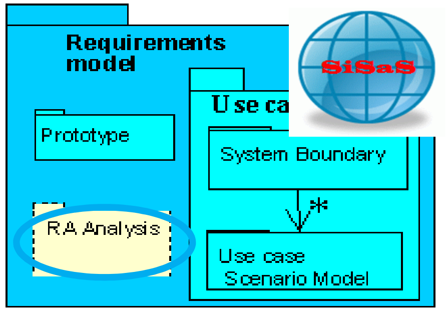
Use case template

Use case <number>	**Use case <name>** **Each use case is identified for later reference.**	
Priority	Priority with respect to implementation, marked 1 – 10.	
<u>Goal</u>	Description of the goal(s) for the use case, derived from the Goal Model in the Business Model.	
Actors	Description of the roles involved in the use case and their responsibilities.	
Pre-conditions	Description of the pre-conditions for the use case.	
Post-conditions	Description of the post-conditions for the use case.	
Façade	Description of methods/operations that the system should provide to realize the use case.	
Quality requirements	Description of extra requirements for this use case. Issues such as uptime, availability, security, performance, etc. can be recorded here.	
Scenario	Description of the use case scenarios. This includes the primary scenario and related secondary scenarios.	
Description	**Step**	**Action**
	1	Step 1 description.
	2	Step 2 description.
	n	Step n description.

Quality / Non-functional requirements

- Non functional requirements
  - Description of non functional requirements for this use case
  - Consider:
    - Performance
    - Uptime
    - Availability
    - Security
    - Scalability
    - Distribution
    - Reliability
    - and more ...

RA Analysis



- Description
  - The Reference architecture analysis is the first step in modeling the architecture, representing an intermediate step between Business Domain Models and the architecture design.
  - The Reference Architecture Analysis is developed using a use case driven approach and provides the basis for the initial development of the Architecture Model.
- Goal
  - The Reference Architecture Analysis should provide the link between analysis and design, in particular the link between the Use Case Model and the Architecture Model.
- Deliverables
  - Detailed use case diagram with subsystem grouping.
  - Updated use case descriptions following the use case template.
  - First version of the Component Structure using the bus architecture pattern (Object Management Architecture pattern).

References