Requirements capture in UML: Developing Use Cases

CS3: SEOC1

Note 3

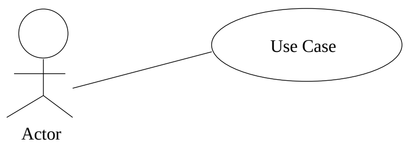
Developing Use Case Models

Use Case models represent the actors (human and non-human) who gain benefit from interacting with the system and the interactions they expect the system to support. This is an important plank of the requirements analysis for any system.

Use Case Models Help:

- Capture requirements
- Plan iterations of the development (prioritise what should be done next).
- System validation.

Use case diagrams



- Actor are represented by actor icons: actors represent *roles* rather than individuals.
- Use case names are contained in oval icons
- Line between an actor and a use case indicates an association with the use case. The association is not inevitable – the actor may be associated in some circumstances.

Example

- An example actor for some security system might be *Control Room Guard*. This does not indicate a particular individual but a role many individual guards might undertake.
- *Evacuate Building* might be a use case with whom the *Control Room Guard* is associated but in certain circumstances (e.g. some failure modes of a system with a remote control room) the *Control Room Guard* may not be involved in an evacuation.
- It is quite possible that more than one actor may be involved in a use case. In some cases no actor may be associated with a use case.

Hospital Information System (HIS)

An NHS hospital wishes to implement a system which controls the allocation of patients to hospital beds. The system may be assumed to have an accurate record of all wards in the hospital and how many beds each ward has. The system will hold the nursing records for hospital patients. At present new patients are admitted to a particular hospital ward. The duty nurses for that ward create a new record for each patient and allocate them to a particular bed. The duty nurses also record when a patient is discharged from hospital, thereby freeing up the relevant bed. On occasion nurses are required to designate some number of free beds as "reserved" (and thus no longer free). The patient record initially contains a patient reference number, a note of their ailment, and the date of their admission. Nurses should be able to add to these records any notes concerning nursing care or change in patients health. The hospital managers should be able to check the avalibility of beds throughout the hospital.

Actors

Beneficiaries: If an actor benefits from a use case there should certainly be an association. If not then associating an actor who does not benefit may be taking a design decision.

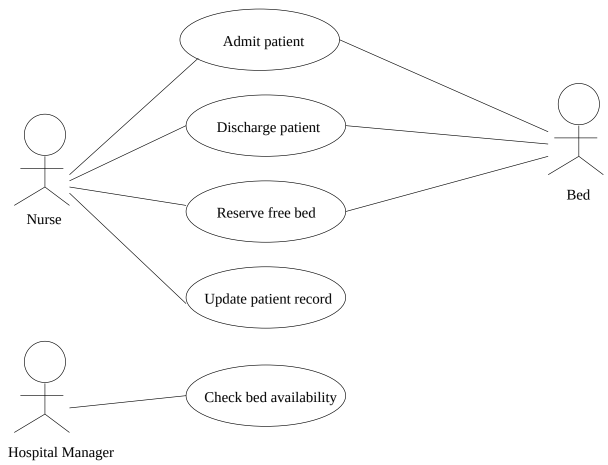
Identifying Actors: look at potential real people interacting with the system. Then look at them from the systems point of view. If they are "the same" as far as the system is concerned then we see them as being the same actor.

Non-human Actors: Different views, count if:

- There is an interaction
- If the outside system initiates contact
- If the outside system gains benefit from the interaction

For example: In the HIS, should "patient" and "record" be actors?

A Use Case Diagram for HIS



Use Cases

Scenario: An instance of a use case: "Mr Brown presents the Budget later this year.". This is easily gathered data, close to concrete experience. Might be an instance of the use case: "The chancellor presents the budget."

Generalisation: The process of generalisation can be hard because the same use case may have a quite different outcome depending on the instantiation.

Validation: The use case should cover all the scenarios seen as instances.

Documenting use cases

By itself, the use case diagram is insufficient. The use cases should also be individually documented. For example:

Admit patient: A duty nurse creates a new record for the patient and allocates them to a bed.

Discharge patient: A record is made of the discharge and the relevant bed becomes free.

Update patient record: A nurse adds to a record any change in patient health, or a note concerning nursing care.

What Use are Use cases?

Requirement Capture: We identify actors then we identify their needs from the system and how they facilitate others needs. We can also prioritise use cases according to strength of needs. Need to identify any use cases with no associated actor.

In Development:

- **Planning:** Effort estimation, scheduling, project risk analysis and risk reduction can all be structured around use cases
- **Organisational Aspects:** to keep the project alive, plan to deliver early iterations that influence project sponsors/funders. Having use case analysis helps this.

What Use are Use cases? (cont)

In Development (cont):

- **Technical Aspects:** High project risk use cases should be restructured to mitigate against risks. Avoid offering risky things, use risk reduction to restructure risky use cases.
- **Validation:** Validate against the use case throughout the life-cycle. Can use use case to structure system tests, usability trials, selection of alpha and beta sites from candidates.

Problems

- Can build a non-OO system by concentrating too closely on needs.
- It is easy to get design decisions into the use cases.
- Some requirements aren't directly associated with user needs.

Common Use Case Mistakes

- Failing to consider all scenarios
- Ignoring system response; focusing only on user interactions
- Descriptions too terse
- Describing attributes and methods instead of usage
- Writing functional requirements instead of usage scenario texts
- Failing to consider user interface

Summary

- Identify actors
- Identify use cases.
- Make associations between actors and use cases.
- Use the use case to help throughout the life-cycle.