Modelling Scenarios & Use Cases

Emmanuel Letier
Department of Computer Science
University College London
http://www.cs.ucl.ac.uk/people/E.Letier
Scenarios

• **Scenario** = temporal sequence of interactions between agents
  
  - positive scenario (what may happen) vs. negative scenario (what should not happen)
  
  - normal scenario (describe normal sequence of events) vs. exceptional scenario (describe desired sequence of events when one deviates from normal scenarios)

• Notations
  
  - Structured natural language
  
  - UML Sequence Diagrams
Structured natural language description

Sequences of steps: <agent> <performs an action>

1. a member of the Public calls the emergency service
2. a Control Assistant (CA) answers the calls and writes down the call details on a pre-printed form
3. the CA identifies the incident location on a map book
4. when the call is completed, the CA places the form on a conveyor belt
5. the conveyor belt transports the form to the central collection point
6. ...
UML Sequence Diagrams (SD)

- Modelling tips
  - at RE level, don't decompose the Machine into components
  - don't include guards on events, alternative branches, loops
    - a scenario is one example of execution of the system
Consistency between Scenarios and Context Diagrams

- Agents in scenarios must correspond to agents in the context diagram
- Agent interactions in scenarios must correspond to interfaces in the context diagram
Use Cases

- Useful to show the boundary of the Machine
  - unlike context diagrams, it doesn’t show other interactions in the World & is vague about physical interfaces (shared phenomena)
- Contents of use cases must be described!
Use Cases

• Many different definitions of what a use case is
  - Different relations to scenarios
    • 1 use case = 1 scenario
    • 1 use case = set of scenarios
  - Different scopes
    • product-level use cases: represent only interactions with the Machine (i.e. events in the World outside the machine are not considered!)
    • system-level use cases: represent interactions between agents in the World and with the Machine
• Most common definition (A. Cockburn)
  A use case is a set of scenarios describing sequences of interactions with the machine related to a single user’s goal
Example of format for describing use cases

**UseCase** Encode Call Details

**Precondition** an incoming call is waiting to be answered

**Postcondition** a IncidentForm is created and passed to the relevant Allocator

**Main success scenario**
1. Call Handler (CH) responds to call
2. CH encodes incident details
3. CH validates incident location with map gazetteer
4. CH terminates call
5. CAD software assigns incident to relevant Allocator

**Alternative scenarios**
2a. CH determines calls does not require intervention
2b. Caller or Phone network terminates call
   .1 CH determines calls does not require intervention
   .1a CH determines call could require intervention
Use Case Modelling Tips (1)
(based on A. Cockburn, Writing Effective Use Cases, Addison-Wesley, 2000)

- Use active voice to show who performs the action
  
  <actor> <performs action>

  - *Bad*: the call is terminated
  - *Good*: the Call Handler terminates call

- Write use case from the user’s perspective, not from the software perspective

  - *Bad*
    1. Get card
    2. Get PIN
    3. Get transaction type
    4. ...

  - *Good*
    1. costumer inserts card into ATM
    2. costumer encodes PIN
    3. ...
Use Case Modelling Tips (2)

(based on A. Cockburn, *Writing Effective Use Cases*, Addison-Wesley, 2000)

- Show the user’s intent, *not the details of the user interface*
  - *Bad*
    1. System asks for name
    2. User enters name
    3. System prompts for address
    4. User enters address
    5. User clicks “OK”
    6. System presents user’s profile
  - *Good*
    1. User enters name and address
    2. System presents user’s profile
Use Case Modelling Tips (3)
(based on A. Cockburn, Writing Effective Use Cases, Addison-Wesley, 2000)

- Avoid “if” statements
  (by replacing “check whether” by “validate”)

  _Bad_
  1. The user enters its login and password
  2. The system checks whether the password is correct
  3. _If it is_, the system presents the available actions for the user

  _Good_
  1. The user enters its login and password
  2. The system validates that the password is correct
  3. The system presents the available actions for the user
Granularity of Use Cases?

What is the appropriate level of granularity for the use cases for the ambulance despatching system?

- **one big use case**: Respond to emergency call
- **a few medium-sized use cases**
  - encode call, allocate ambulance, send mobilization instructions, …
- **a lot of fine-grained use cases**
  - encode incident details, validate incident location with map gazetteer, …
Cockburn’s Three Levels of Use Cases

- **user goal level**: discrete interactions between primary actor and the Machine in order to achieve a single user goal
  - Cockburn’s heuristics: a single user goal is one that takes a couple of minutes to 30 min for the primary actor to complete

- **summary level**: higher-level use cases involving multiple “user goals”
  - they show how the user goal level use cases fit together

- **subfunction level**: lower-level use cases that are insufficient by themselves to satisfy a meaningful user goal

- The Core set of use cases should be at the user goal level

- Better structuring mechanism: Goal refinement graphs
  (cf. next week)
Benefits of Use Cases & Scenarios

- Extremely useful during elicitation, communication, and validation
- Spontaneously used by stakeholders for
  - explaining how the current system works
  - explaining how the future system could/should work
- Scenarios help formulate stakeholders' needs in a way that is testable
Limitations (1)

- Scenarios can't handle non-functional requirements
- A set of scenarios is *not a specification* (important!)
  - it only gives *examples* of required behaviours
  - Analogy: a set of test cases for a method is not a specification of the method
- Combinatorial explosion problem
  - number of possible scenarios is too large for them to be enumerated completely
    Eg. try to specify all alternative scenarios for the use case "Encode Call Details"
Limitations (2)

- Product-level use cases *focus on interactions with the machine*
  
  -> risk of failing to understand the real problem deeper in the world
  
  -> risks of premature decisions about interfaces and ordering of events

- Required system properties remain implicit

- Scenarios don't distinguish critical from non-critical concerns
  
  - cf. Jackson & Jackson, Separating Concerns in Requirements Analysis: An Example

=> use cases to be complemented by goal model