A8: Deriving Use Cases from Organizational Modelling

Victor F.A Santander, Jaelson F.B. Castro
RE’02

Connection from i* to other techniques (A7, A8): motivation revisited

- i* (GRL) represents actors and goal, and their dependencies and interrelationships
  - but not complete processes.
  - i* for early RE (analysis)
  - Must be connected to other representations
- A7. Connection to use case maps (UCM) (last week)
- A8: Connection between i* → use cases is regarded as useful
- A9-A10: Examples of connection to other approaches (Wednesday)
System development

Developer
- Analysis
- Specification
- Design
- Coding
- Test
- Maintenance
- Adaptation, integration

User
- Needs
- Requirements
- Procurement
- Installation
- Use
- Interactive models

Framework for different model-driven approaches

Process Spectrum
- Static
- Dynamic

Process Models
- Hardcoded
- Predefined
- Evolving
- Implicit

Model-driven solutions:
- Analysis
  - Enterprise model
- Specification
  - Platform Independent Model
- Design
  - Platform Specific Model
- Detailed d.
  - PIM EE
  - PSM EE
  - Code
- Coding
  - Operating system, runtime execution environment (EE)
  - Enterprise Model EE
  - EE
Briefly on Use Cases

- Very simple process modeling language (external actors, use cases (tasks within the boundary of automation of future system), some possible aggregation mechanism (includes, extends)
- Often accompanied by a template/pattern for explaining the normal and exceptional behavior (scenarios)
- Packaged with UML, important in UML methodology (e.g. RUP, use-case driven one of four main principles)
- When introduced, regarded (in the OO-world) as a big improvement for the possibility of specifying requirements as external observable behavior in OO-projects
- Used extensively to structure requirements
  - "Isn’t this THE way to do this?" (statement from a senior systems analyst in Telenor on a project some years ago)
- Some problems with its use
  - Can easily end up with many use cases, what are the essential ones?
  - Proved difficult to use the simple aggregation mechanisms
  - Informal
  - Gap between Use-case and OO-thinking
- But very beneficial for communication that the diagrams are so simple

From i* to use-cases

- Suggested overall methodology:
  - Use i* to understand the problem domain (early RE)
  - Use cases to represent functional requirements (late RE)
  - Carry soft-goals into non-functional requirements

- Steps
  - First develop the i*-model (Strategic Dependency SD and Strategic Rational SR)
  - Use this to come up with use cases

Figure 3. Steps of the integration process between i* and Use Cases in UML.
**Example**

- **SD - Fig 1**
- **SR - Fig 2**
- **Step 1: Discover actors**
  - Guidelines 1-4, discover 3 actors
- **Step 2: Discover use cases**
  - Guideline 5, find 4 use cases (Tab 3)
  - Guideline 6, find another use case
  - Guideline 7, classification of use cases (Tab 4)
- **Step 3: describe use cases**
  - Eks. "MeetingBeScheduled"
- **Summary UCD in Fig 4**

---

**Figure 1. Strategic Dependency Model for the Meeting Scheduling Problem.**
i* → use cases. Guidelines step 1:

1. Every i*-actor is a potential Use Case actor
2. i*-actor should be external to the intended software system
3. If the i*-actor is external to the system, check if they are relevant to the system looking at dependency-relationships
4. ISA relationships between i*-actors becomes generalization relationships in the Use-Case diagram
i* → use cases. Guidelines for step 2:

5. For every actor (discovered in step 1), find all dependencies where this actor is the dependee
   1. Goal-dependency → use case goal?
   2. Task dependency → use case?
   3. Resource dependency → use case?
   4. Soft-goal dependency → non-functional requirement

6. Analyze situations where the system depends on an external actor.
   - Will often results in use-cases where a person initiate the work of the system

7. Classify use case
   - In accordance with the classification scheme of Cockburn (business goal, summary goal, user goal, sub-function level goal)

i* → use cases. Guidelines for step 3:

- Analyze each actors relations in the Strategic Rationale Model
- Find information useful for describing steps in the use case scenario
- Specifically look at:
  - means-ends-relationships
  - task decomposition-relationships
Summary, from i* to other models (A07, A08)

- i* is primarily for early RE/analysis
  - Actors, actor dependencies
  - Goal hierarchies
  - To understand the problem domain and the organization
  - Both for modeling as-is and to-be situation

- Can be connected to
  - Later requirements specification (use cases)
  - Evaluation of design-alternatives (Use Case Maps)

- More experiences needed

- The examples so far pedagogical toy-examples, on Wednesday some experiences from a larger project trying to use these techniques
A8: Deriving Use Cases from Organizational Modelling

Victor F.A Santander, Jaelson F.B. Castro
RE’02