


URN: User Requirements Notation

Table of Contents

1. ☰ Overview of URN	1
2. 📖 Principles	1
3. 🎯 Capabilities	1
4. GRL	2
4.1. Basic concepts	2
4.2. Basic notation	2
4.3. Impacts	3
4.4. Benefits	3
5. Use Case Maps	4
5.1. Capabilities	4
5.2. Basic concepts	5
5.3. Basic notation	5
5.4. Benefits	5
Appendix A: Appendices	6
A.1. GRL Notation	6
A.2. UCMs Notation	7
Appendix B: Useful links	8

1. ☰ Overview of URN

- Daniel Amyot, U. of Ottawa 
- ITU-T (Telecom) standardization process

2. 📖 Principles

- **User** vs. Systems requirements
- Several notations:
 - GRL (Goal-oriented Requirement Language)
 - UCMs (Use Case Maps)
- Graphical syntax

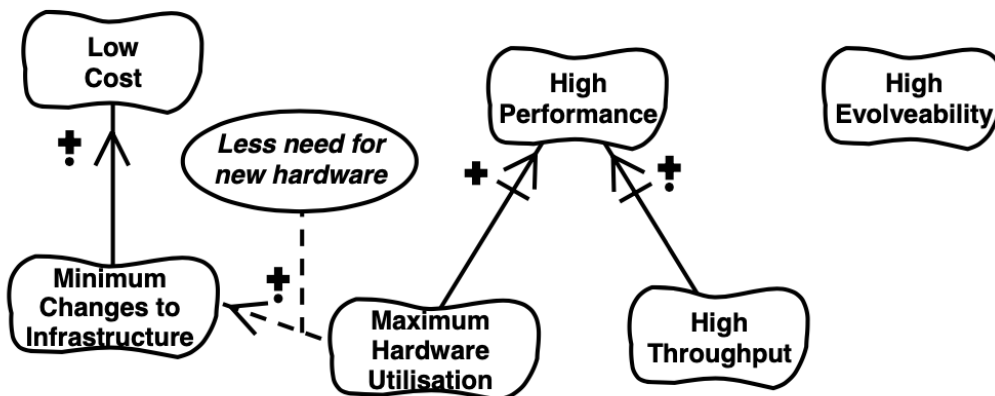
3. 🎯 Capabilities

- Capture user requirements
- Scenarios as 1st class entities

- Focus on elicitation and transition to design
- Allocation of scenario responsibilities to components
- Detection/reasoning of features
- Address goals and NF requirements
- Formal grammar (supports transformations and exchanges)

4. GRL

Example of GRL model (source [here](#))



A **goal** is an objective or concerned used to discover and evaluate requirements.

4.1. Basic concepts

Goals

business or system

Alternative means

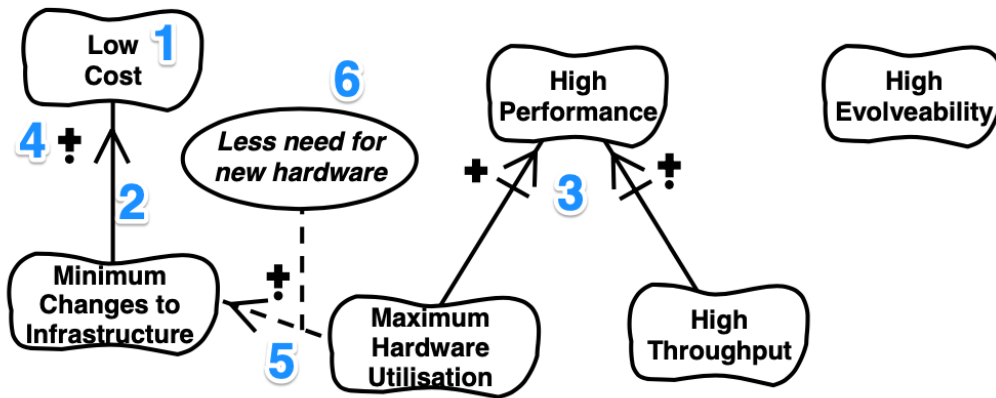
of achieving goals

Rationales

for contributions and decisions

4.2. Basic notation

GRL Notation (source [here](#))



1. Softgoals (fuzzy goals)
2. Contribution link
3. AND composition
4. Impacts
5. Correlation
6. Belief



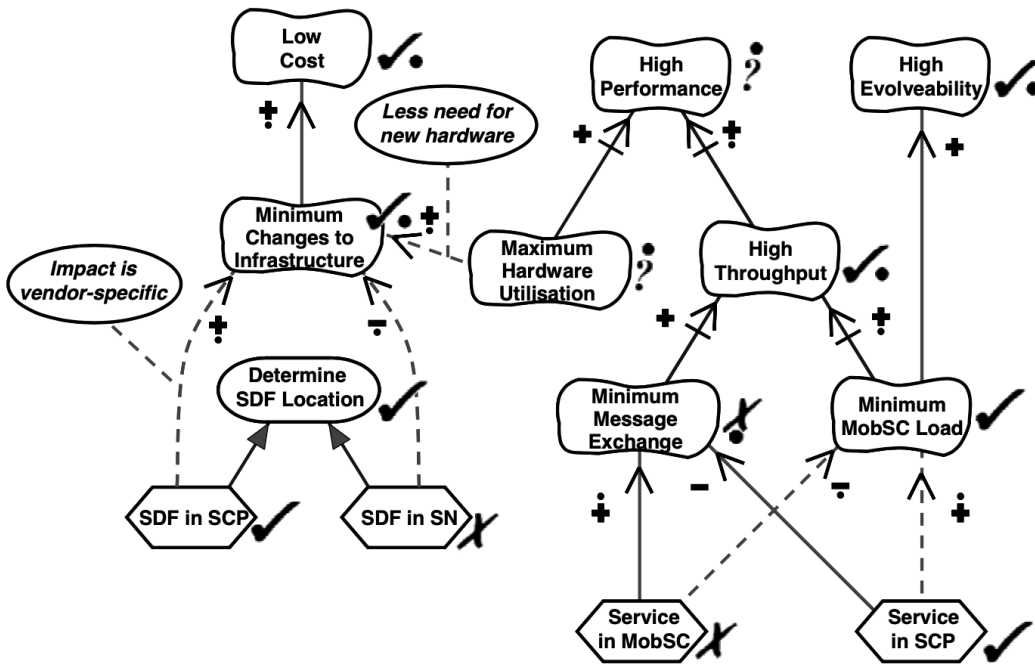
More [here](#).

4.3. Impacts

- positive/negative and sufficient (make/break)
- positive/negative but insufficient (help/hurt)
- unknown positive/negative (some+/some-)

4.4. Benefits

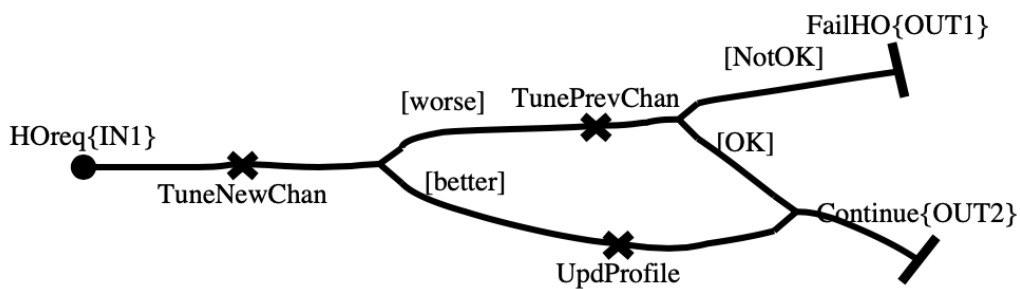
Evaluation of candidate solutions (source [here](#))



- Different level of abstractions
- Tool supported
- Trade-off analysis

5. Use Case Maps

UCMs Notation (source [here](#))



A **functional requirements** is a requirement defining functions of the system.

5.1. Capabilities

- UC capturing and elicitation
- UC validation
- High-level architectural design
- Test case generation

5.2. Basic concepts

Scenario

a partial description of system usage

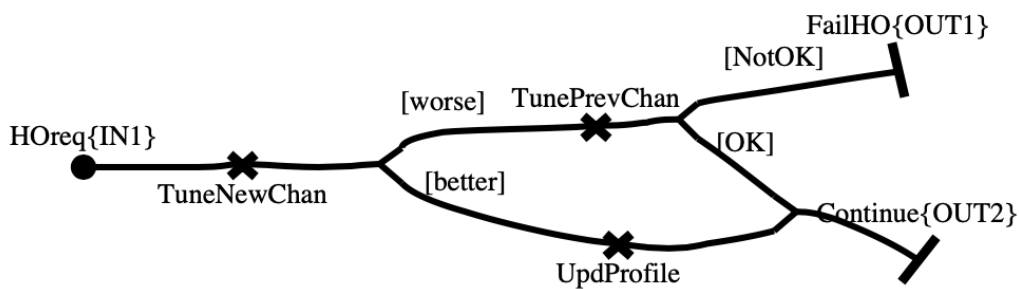
Responsibilities

scenario activities (something to be performed)

Component

entity (software or not) that performs a responsibility

UCMs Notation (source [here](#))



5.3. Basic notation



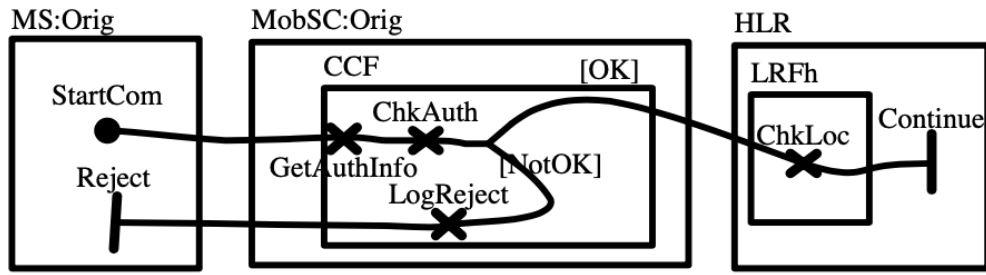
More [here](#).

1. Start point
2. Pre-condition
3. Triggering event
4. Casual paths
5. Responsibilities
6. Fork
7. Condition
8. Endpoint

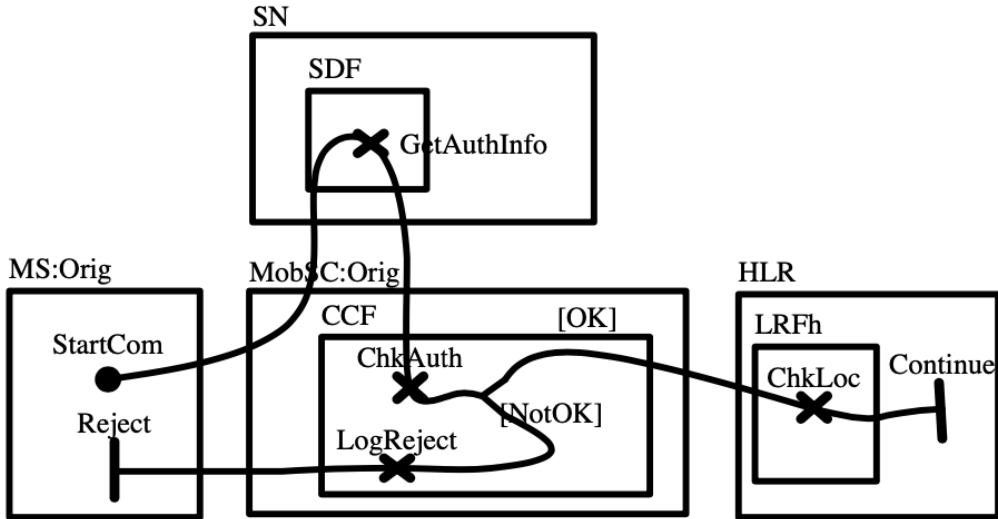
5.4. Benefits

- UCMs can integrate many scenarios
- To analyze potentially conflicting scenarios
- To generate artifacts (MSCs, SD, test cases)
- To analyze alternatives designs

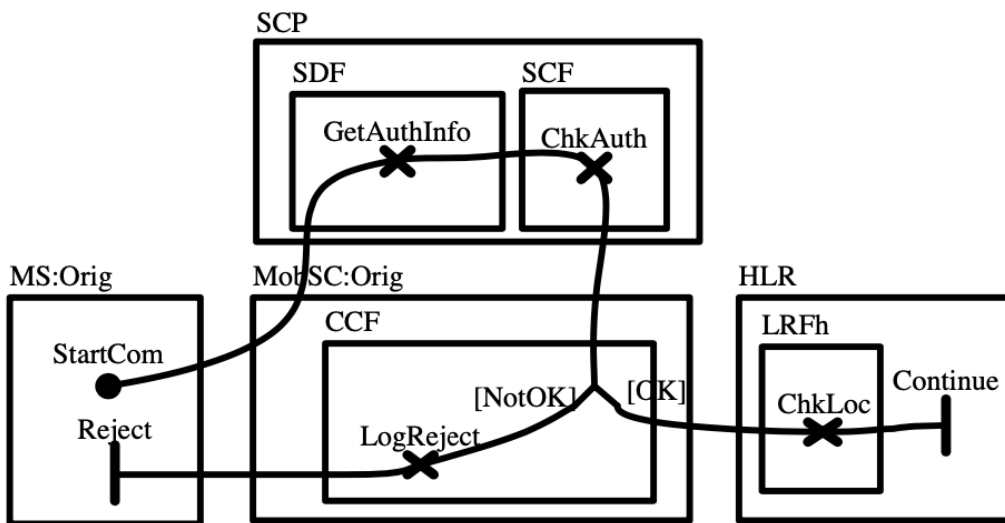
Alternative component structures (source [here](#))



(a) Service in MobSC



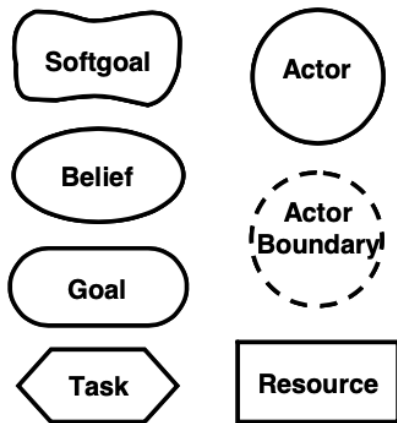
(b) Service in MobSC, SDF in SN



(c) Service and SDF in SCP

Appendix A: Appendices

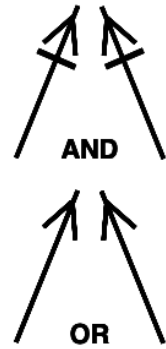
A.1. GRL Notation



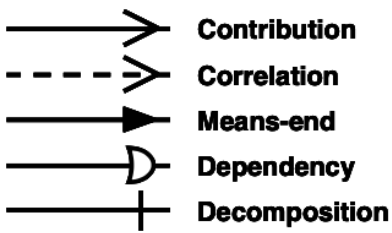
(a) GRL Elements



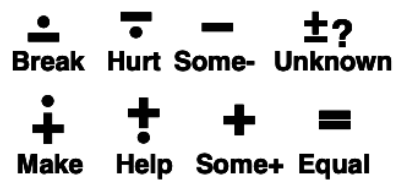
(b) GRL Satisfaction Levels



(c) Link Composition



(d) GRL Links



(e) GRL Contributions Types

Figure 1. GRL Notation (source [here](#))

A.2. UCMs Notation

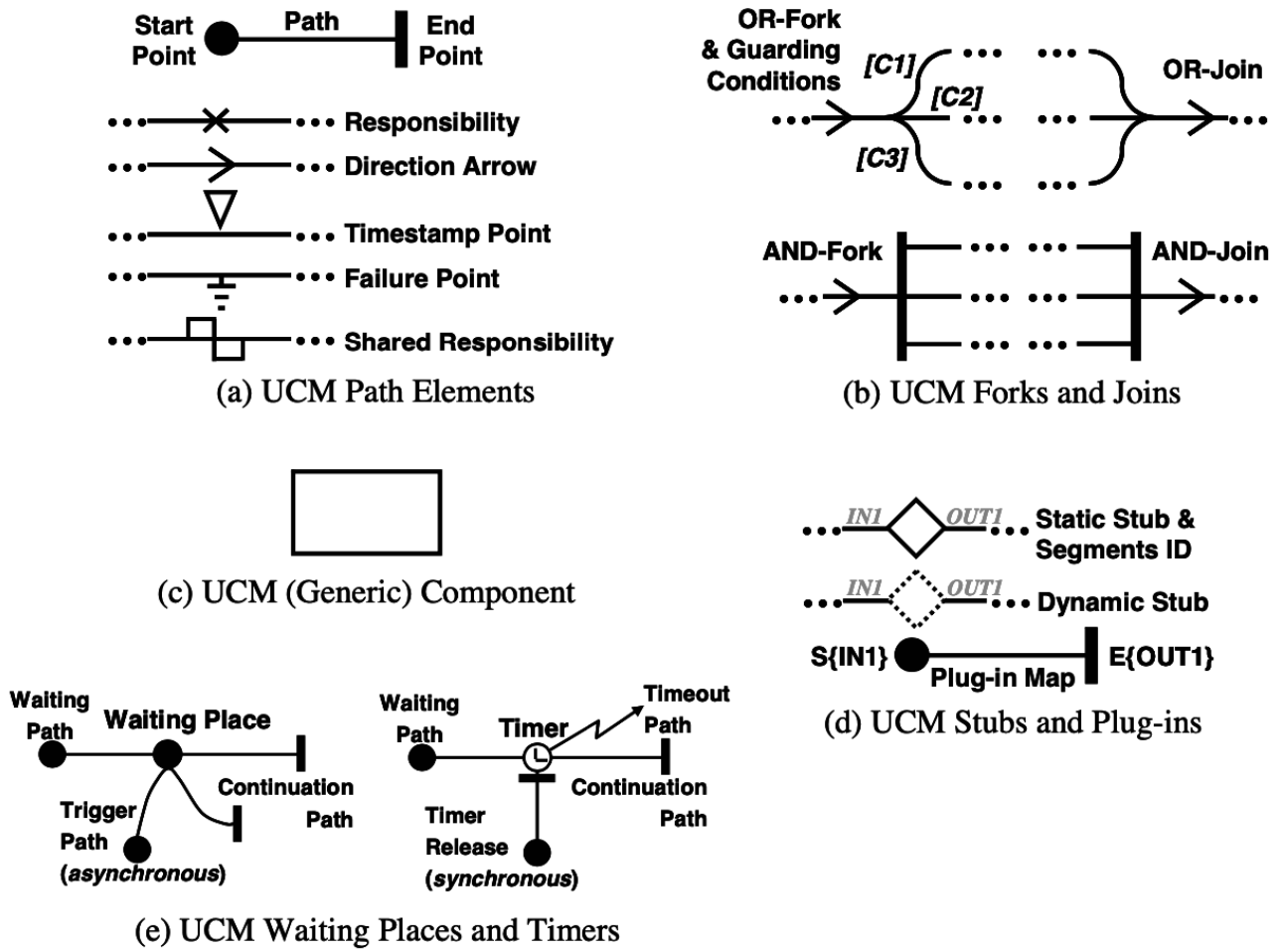


Figure 2. UCM Notation (source [here](#))

Appendix B: Useful links

- [Recent Introduction Article](#)
- [Daniel Amyot web page](#)
- L'outil [jUCMNav](#)