URN: User Requirements Notation

Table of Contents

1. Overview of URN . . . 1
2. Principles . . . 1
3. Capabilities . . . 2
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
6. Mapping with the Unified Framework . . . 6
    6.1. The 7 kinds of class . . . 6
    6.2. References between classes . . . 7
    6.3. Applied to URN . . . 7
Appendix A: Appendices . . . 8
    A.1. GRL Notation . . . 8
    A.2. UCMs Notation . . . 8
Appendix B: Useful links . . . 9

1. Overview of URN

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

2. Principles

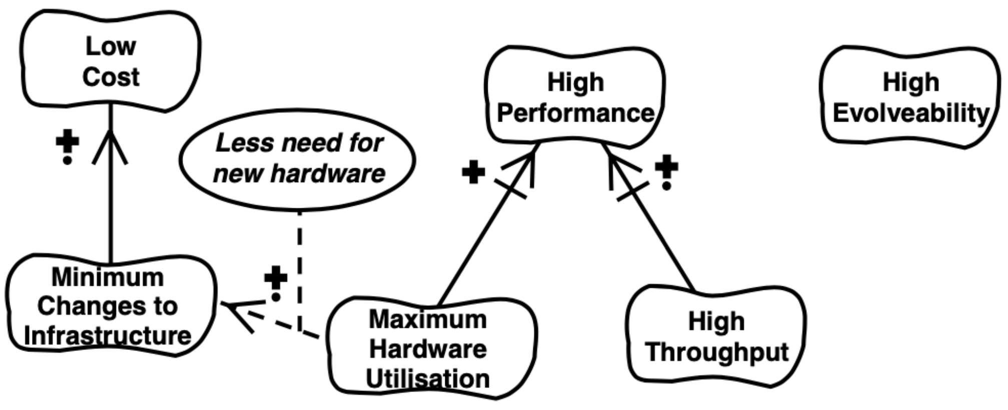
- **User** vs. Systems requirements
- Several notations:
    - GRL (Goal-oriented Requirement Language)
    - UCMs (Use Case Maps)
- Graphical syntax

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)


Low Cost
Less need for new hardware
High Performance
High Evolveability
Minimum Changes to Infrastructure
Maximum Hardware Utilisation
High Throughput
+
+
+
+
+

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

4.1. Basic concepts

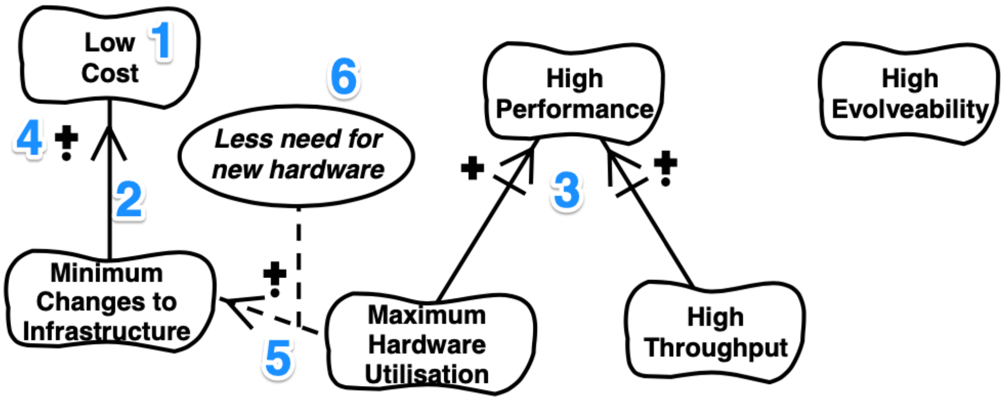
Goals
: business or system

Alternative means
: of achieving goals

Rationales
: for contributions and decisions

4.2. Basic notation

GRL Notation (source here)


Low Cost
1
6
Less need for new hardware
High Performance
High Evolveability
4 +
2
+
3
+
Minimum Changes to Infrastructure
+
5
Maximum Hardware Utilisation
High Throughput

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

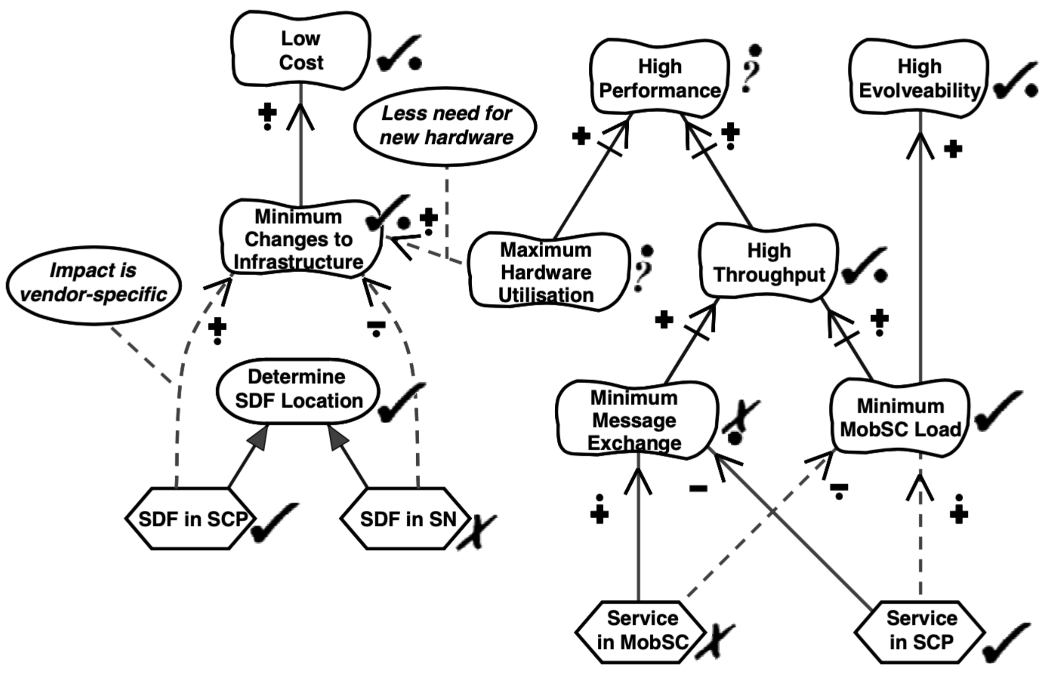
More here.

4.3. Impacts

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

4.4. Benefits

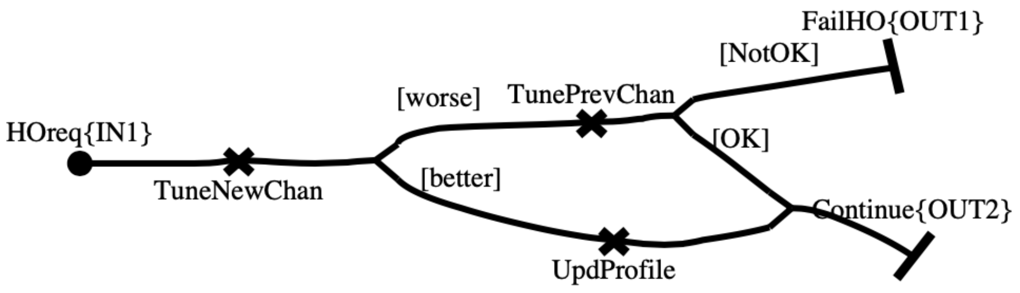
Evaluation of candidate solutions (source here)



- 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

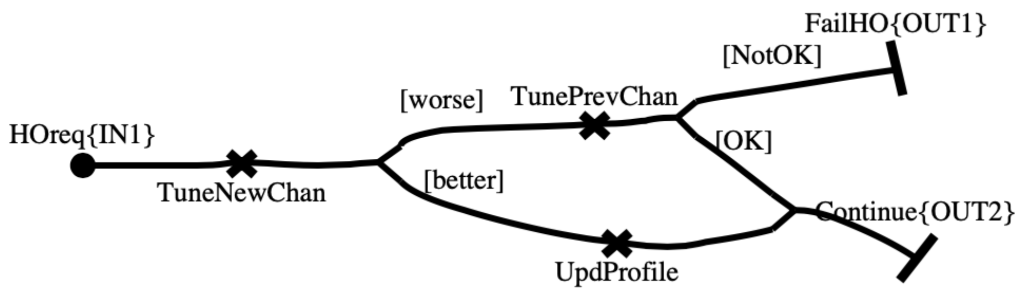
Responsibilities

scenario activities (something to be performed)

Component

entity (software or not) that performs a responsibility

UCMs Notation (source here)



5.3. Basic notation

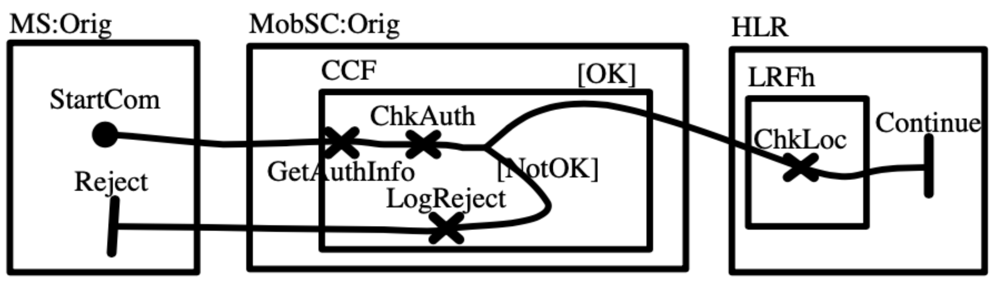
More here.

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

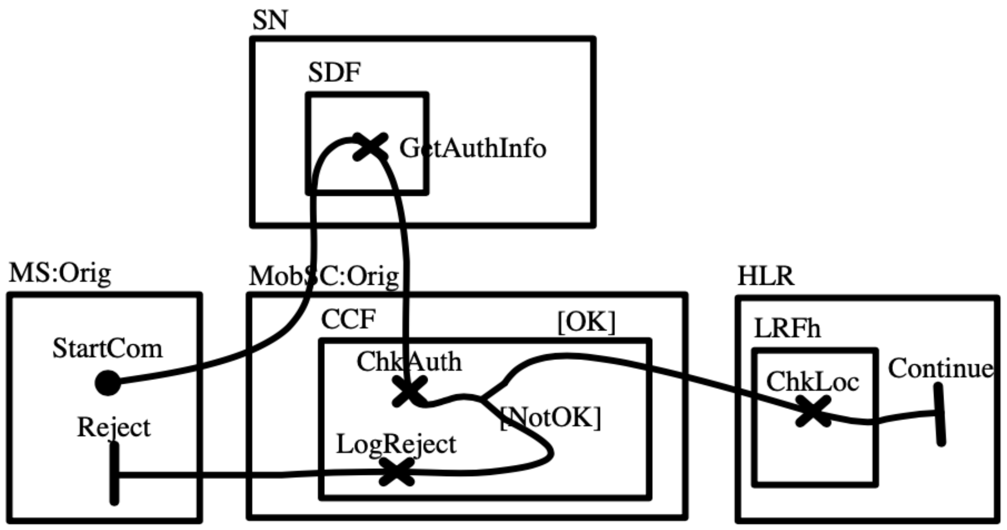
5.4. Benefits

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

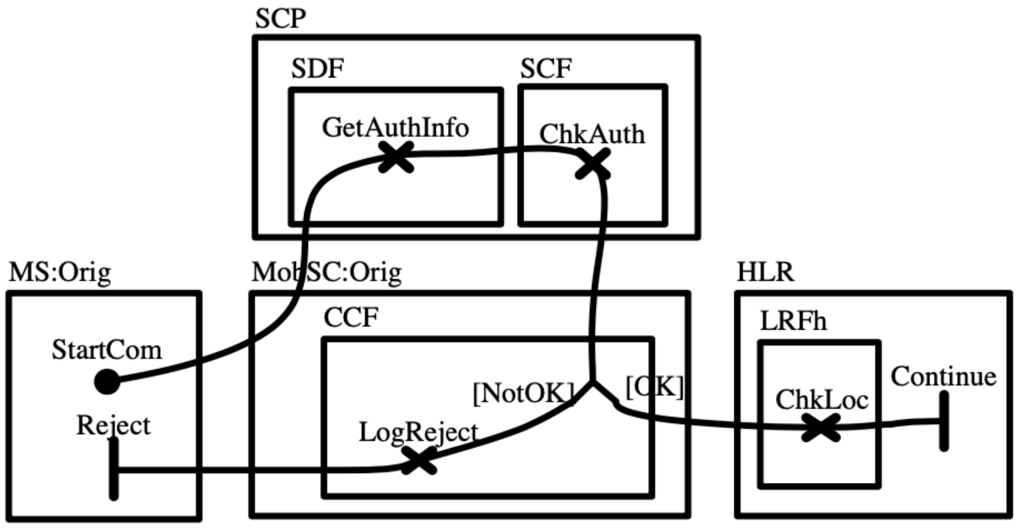
Alternative component structures (source here)



(a) Service in MobSC



(b) Service in MobSC, SDF in SN



(c) Service and SDF in SCP

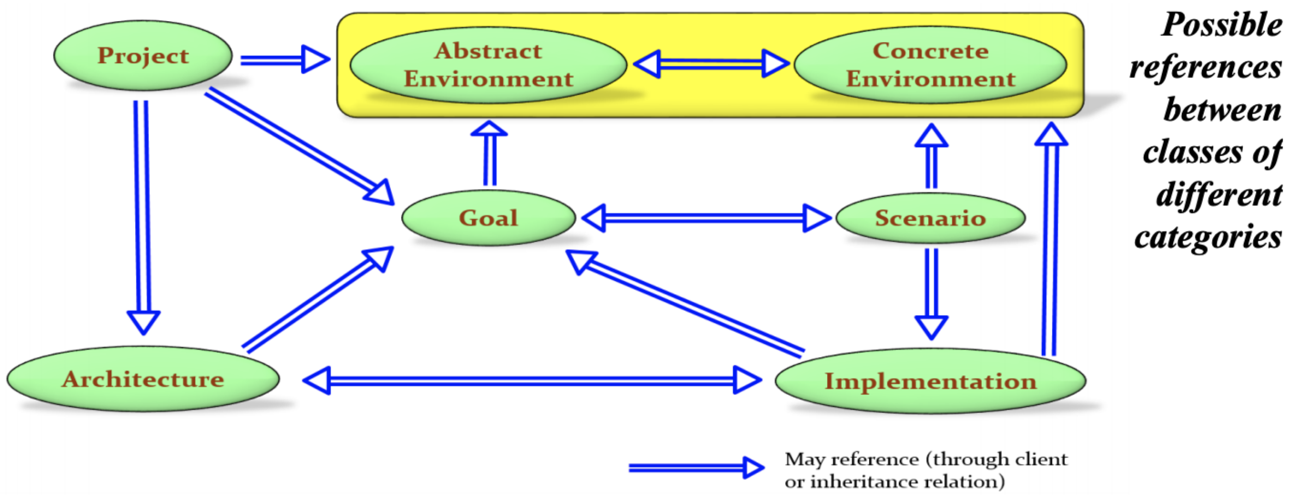
6. Mapping with the Unified Framework

6.1. The 7 kinds of class

- Requirements
    - Concrete Environment

    - Abstract Environment
    - Scenario
- Design and implementation
    - Architecture
    - Implementation
- Goals and project
    - Goal
    - Project

6.2. References between classes



Project
Abstract Environment
Concrete Environment
Goal
Scenario
Architecture
Implementation
Possible references between classes of different categories
May reference (through client or inheritance relation)

Figure 1. References between classes (source here)

6.3. Applied to URN

Kind of class	URN Concept	Details
Concrete Environment	Actor, Resource	
Abstract Environment	Actor Boundary, Belief?	
Scenario	UCM Paths	
Architecture		Out of scope
Implementation		Out of scope
Goal	Goal, Softgoal, Belief?	
Project	Task, Resource?	

Appendix A: Appendices

A.1. GRL Notation

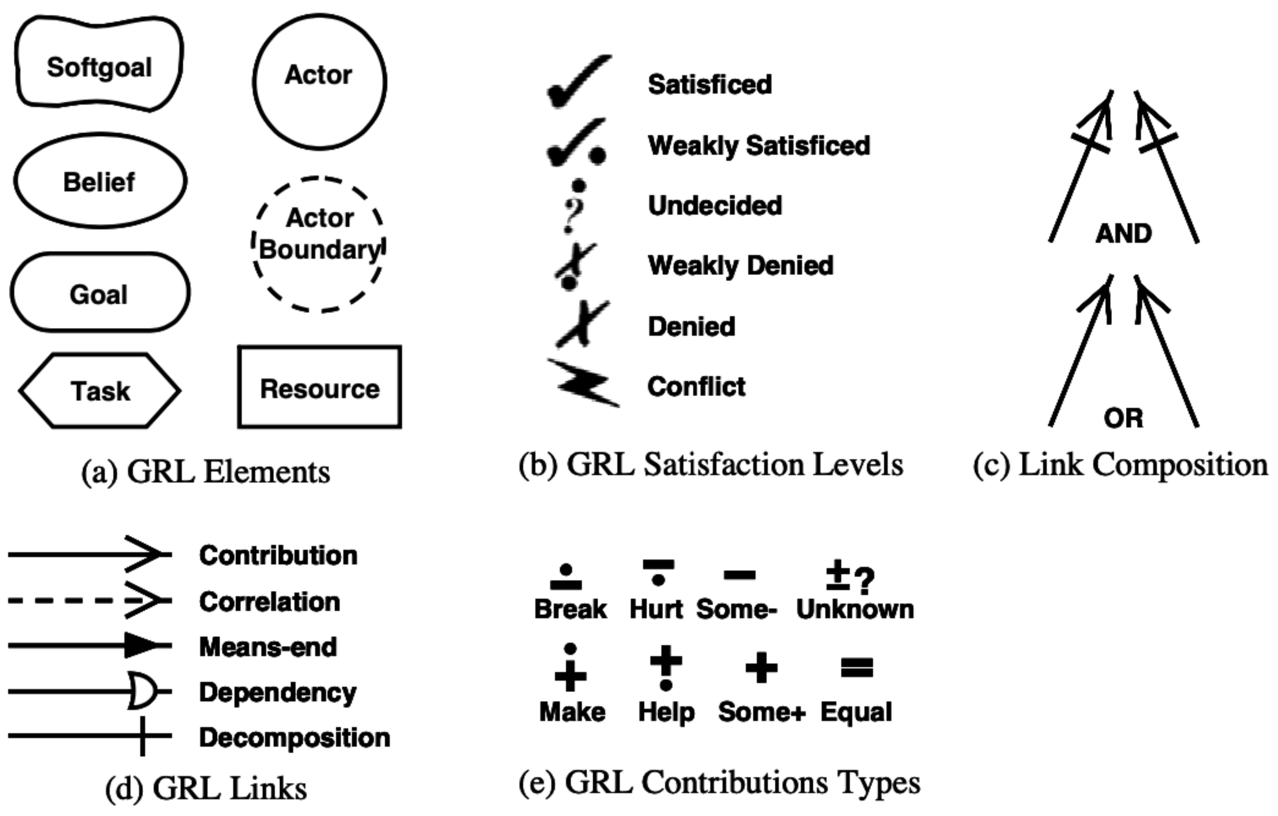


Figure 2. GRL Notation (source here)

A.2. UCMs Notation

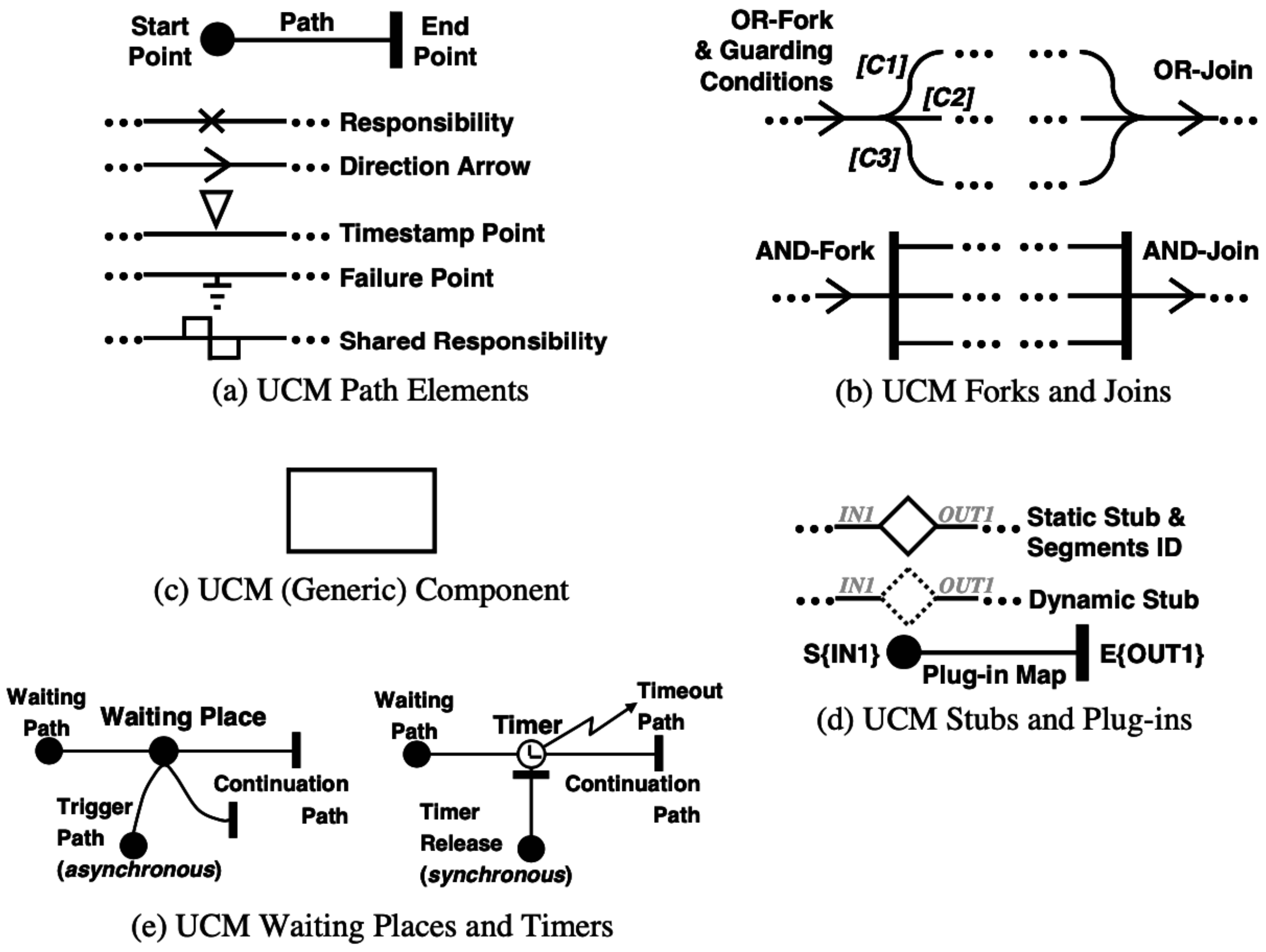


Figure 3. UCM Notation (source here)

Appendix B: Useful links

- Recent Introduction Article
- Daniel Amyot web page
- L’outil jUCMNav