



STANDARDIZING REQUIREMENTS NOTATIONS: URN, AND WHAT ELSE?

CSRS, May 26, 2003

Daniel Amyot


ITU-T Q.18/17 Rapporteur
SITE, University of Ottawa, Canada

damyot@site.uottawa.ca

<http://www.UseCaseMaps.org/urn/>



What is a Standard?

- 
- Guideline documentation that reflects *agreements on products, practices, or operations* by nationally or internationally recognized industrial, professional, trade associations or governmental bodies,
 - or is accepted *de facto* by industry or society.

Source: F. Coallier, *An Introduction to International IT Standardization*, 2002



Types of Standards



Organization Standards

- Such as internal company standards

Market Standards (De Facto)

- Such as Microsoft Windows

Professional Standards

- Developed by Professional organizations (e.g. IEEE)

Industry Standards

- Developed by industrial Consortia (e.g. OMG)

National Standards

- Developed by national standards organization (e.g. CSA)

International Standards

- Developed by formal international standard organization (e.g. ITU, ISO, IEC, ...)



Added Value of International Standards



In addition to the Brand:

- They represent an international **consensus** attained through a very rigorous and uniform process
- They represent sets of conventions and/or technical requirements or practices that are **relatively stable**
- The development process makes it relatively difficult and costly for special interests to take over a given standardization project, especially if the topic is controversial.
- They are beneficial to users, companies, and tool vendors



Requirements Engineering and Standards



Focus mainly on templates and processes!

- IEEE 830-1998: Software Requirements Specifications
- IEEE 1233: Guide for Developing System Req. Specs.
- ISO/IEC 12207: IT - Software Life Cycle Processes
- ESA PSS-05-02, 03: Guide to the User (resp. Software) Requirements Definition Phase

Languages and notations

- OMG: Unified Modeling Language 1.5
- ITU-T: MSC, SDL
- ISO/IEC: LOTOS, Estelle

But are these really for RE ???



URN Standardization Effort at ITU-T



- Question 18 of Study Group 17 is the **User Requirements Notation (URN)**
- Z.15x family of standards
- URN Focus Group
 - Progress on URN between SG17 meetings
 - <http://www.UseCaseMaps.org/urn/>



URN – Main objectives



- Focus on early stages of development with **goals** and **scenarios**
- From user requirements to system functional and non-functional requirements
- No messages, components, or component states required
- Reusability
 - of argumentations (goal patterns and analysis)
 - of scenarios (patterns and architectural alternatives)
- Early performance analysis
- Traceability and transformations to other languages
 - Particularly MSC, SDL, TTCN, and UML



Proposal for URN



Combined use of two complementary notations:

- **Goal-oriented Requirement Language** (GRL)
 - for goals and non-functional requirements
 - <http://www.cs.toronto.edu/km/GRL/>
- **Use Case Maps** (UCM)
 - for functional requirements
 - <http://www.UseCaseMaps.org/>



URN Family of Standards

Z.150 URN

- *Recommendation Z.150, User Requirements Notation (URN) – Language Requirements and Framework.*
- Approved in February 2003.

Z.151 GRL

Z.152 UCM

Z.153 Methodological Approach

Z.159 UML 2.0 profile for URN

To be available by March 2004...



GRL in a Nutshell

Goal-oriented Requirement Language

- graphical notation
- connects requirements to business objectives
- allows reasoning about (non-functional) requirements

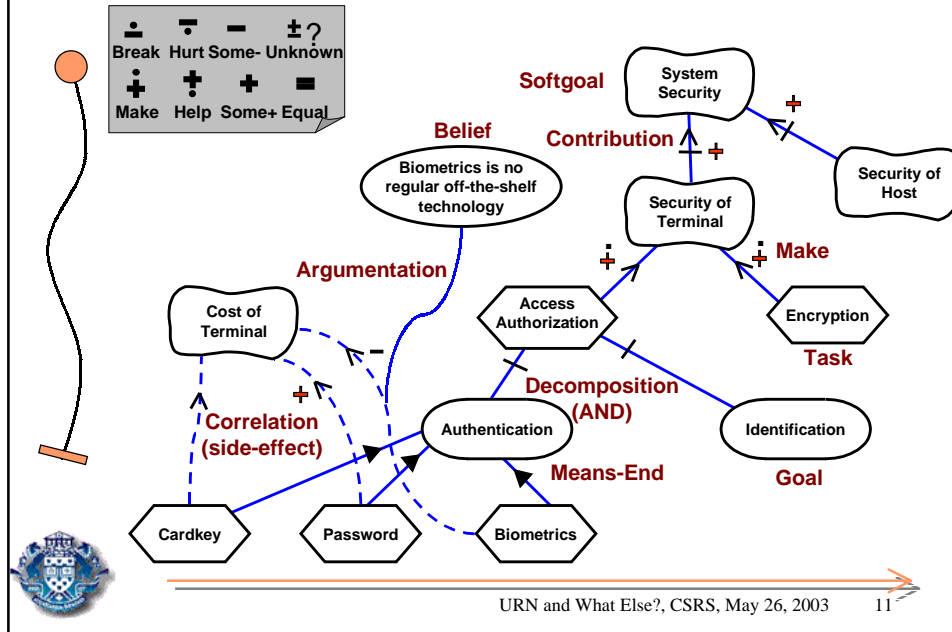
GRL models the “why” aspect

- objectives, alternatives, as well as decision rationale
- no operational details

Supports goal analysis and evaluations

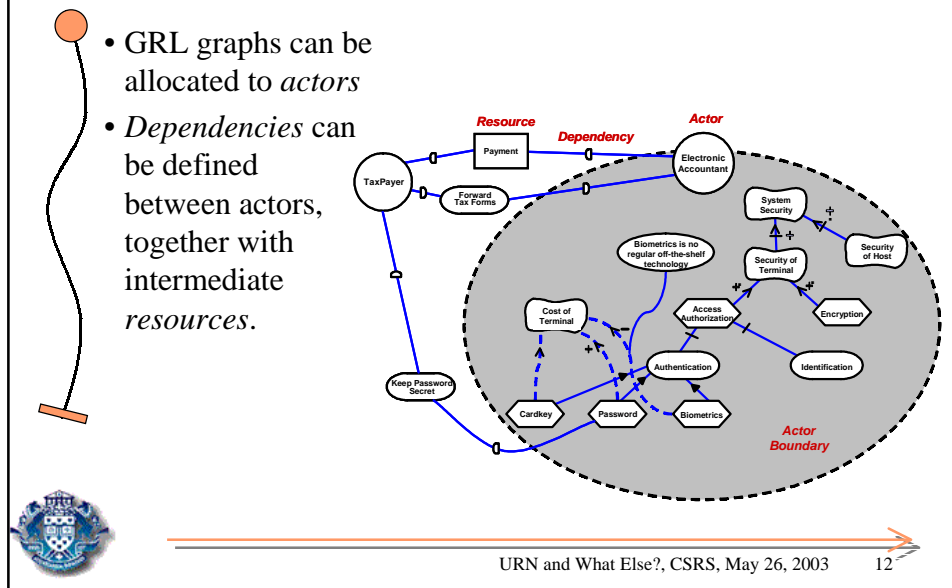


Basic GRL Notation

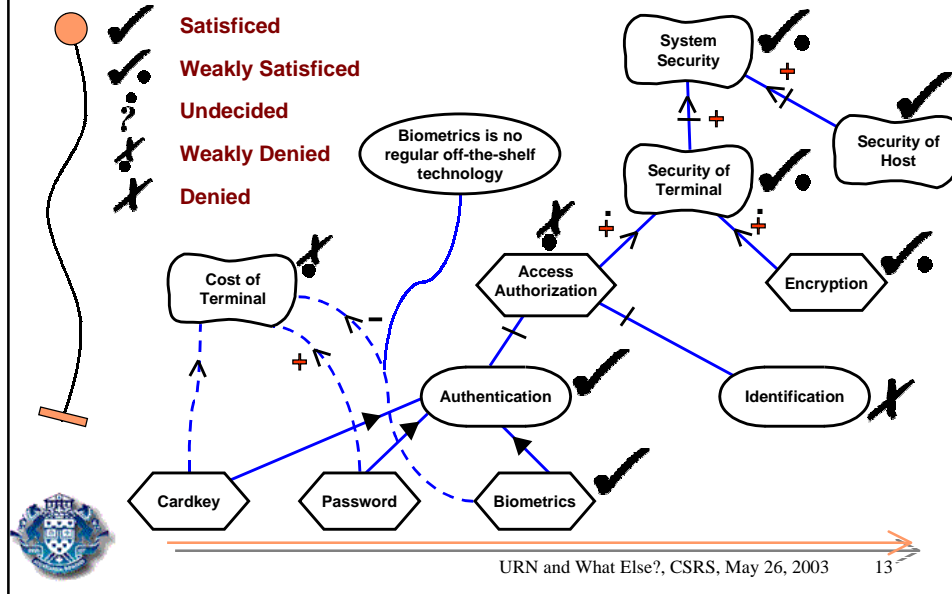


Advanced GRL notation (for your information only)

- GRL graphs can be allocated to *actors*
- *Dependencies* can be defined between actors, together with intermediate *resources*.



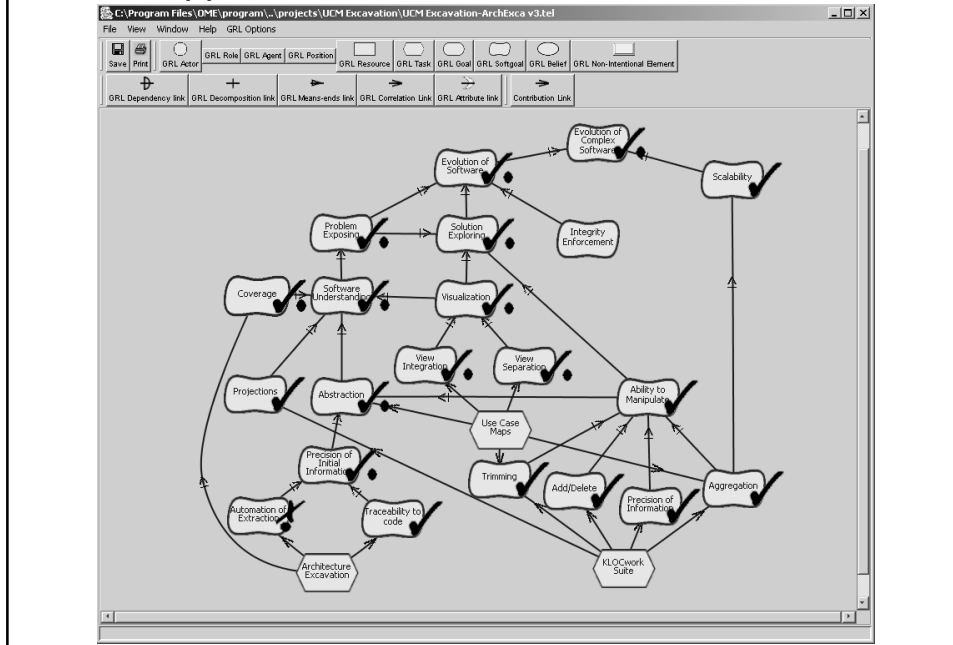
Evaluations with GRL



Evaluations with GRL

- *Evaluations* of GRL graphs show the impact of qualitative decisions on high level softgoals
 - *Propagation* is usually bottom-up
 - Fuzzy evaluation of *satisfaction level*
 - Takes into consideration the contributors:
 - Contributions and correlations (help, hurt, ...)
 - Degrees of satisfaction (satisfied, denied, ...)
 - Composition operators (AND, OR)
 - One could use numerical values and functions instead of qualitative (fuzzy) values
- URN and What Else?, CSRS, May 26, 2003 14

Tool Support: OME 3



UCMs in a Nutshell



Use Case Maps

- graphical scenario notation
- causal relationships between responsibilities
- scenario elements may (optionally) be allocated to components

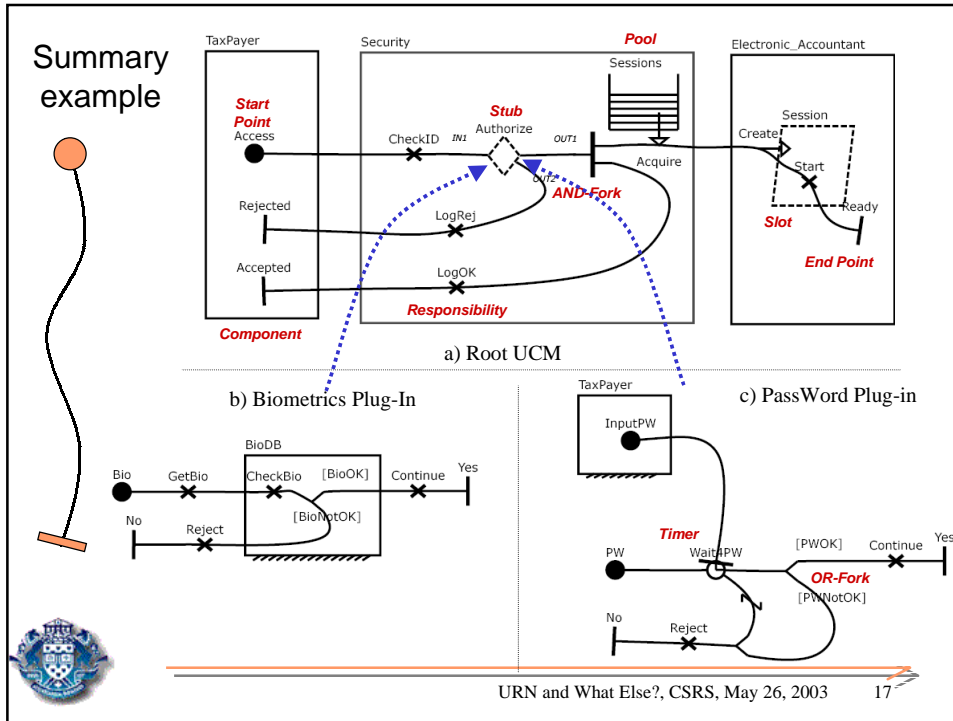
UCMs model the “what” aspects

- functional requirements as scenarios
- integration and reusability of scenarios
- guidance for architecture and detailed behaviour

Performance analysis, conflict detection



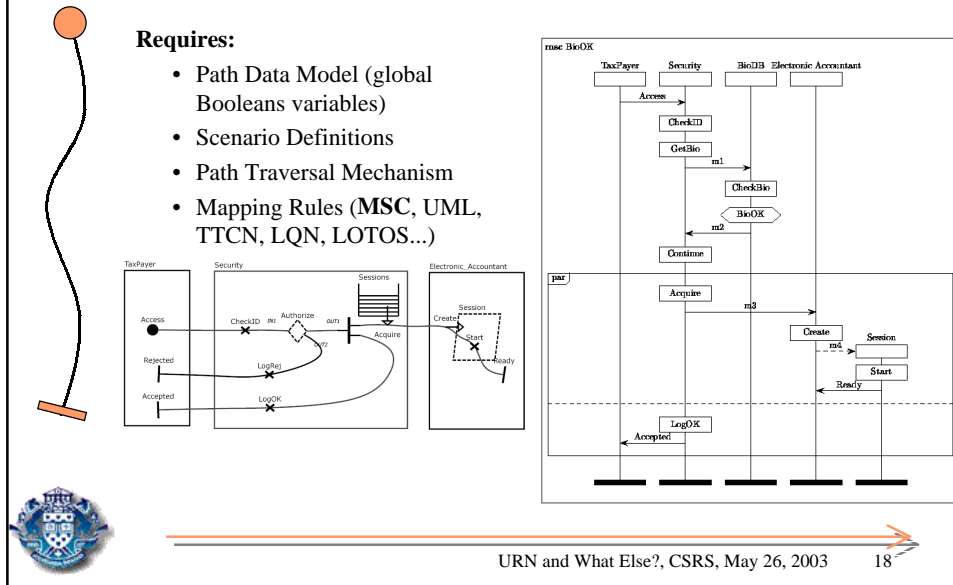
Summary example



From UCM Requirements to More Detailed Design Models

Requires:

- Path Data Model (global Booleans variables)
- Scenario Definitions
- Path Traversal Mechanism
- Mapping Rules (MSC, UML, TTCN, LQN, LOTOS...)



GRL - UCM Relationship



Goal-based approach

- Focuses on answering “why” questions

Scenario-based approach

- Focuses on answering “what” questions

Goals are *operationalized* into tasks and tasks are elaborated in (mapped to) UCM scenarios

- Focuses on answering “how” questions

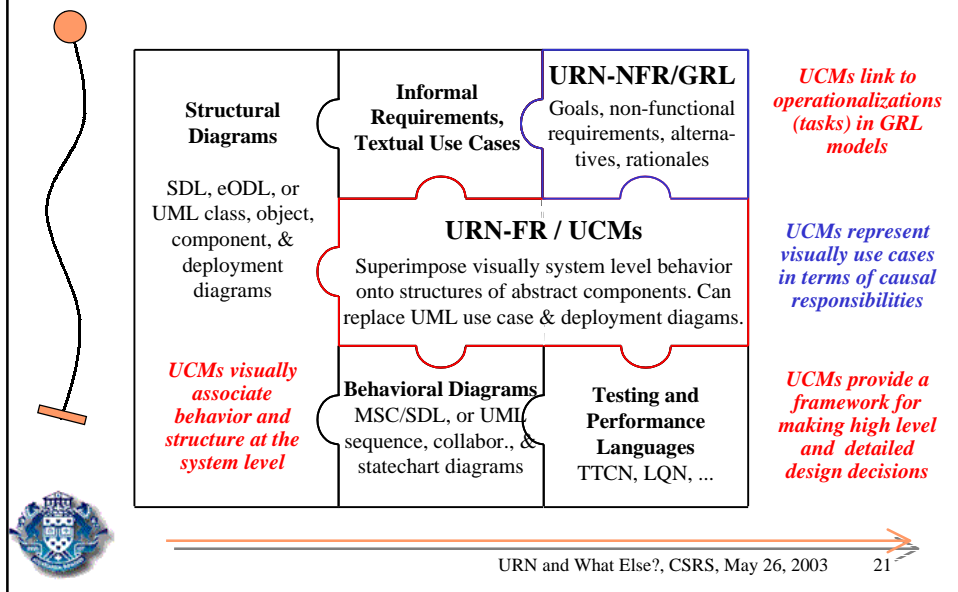
GRL goals can guide the selection of a particular architecture for the UCM scenarios



Tool Support: UCMNav 2.1 (free, + source)

The screenshot displays the UCMNav 2.1 software interface. The main window shows a Use Case Map (UCM) diagram with nodes like 'start', 'fail', 'reportSuccess', 'busyTreatment', 'ringingTreatment', 'Sdisplay', 'OUT1', 'OUT2', 'disp', 'ringTreatment', and 'success'. The interface includes a menu bar (File, Components, Options, Performance Maps, Align, Utilities, Scenarios, About), a toolbar with navigation and editing tools, and a 'Scenario Definitions' dialog box on the right. The dialog box has tabs for 'Scenario Groups' and 'Scenarios', and sections for 'Boolean Variables (unreferenced)', 'Variable Initializations', and 'Scenario Starting Point'.

URN — Missing Piece of the Modelling Puzzle?



Conclusions

URN

- Allows engineers to specify or discover requirements for a proposed system or an evolving system, and review such requirements for correctness and completeness.
- Is usable in industry and in standardization bodies
- Combines goals and scenarios
- Helps bridging the gap between **informal** and **formal** concepts, and between requirements models and design models
- Big benefits for little modelling investment, even when used informally

GRL

- For incomplete, tentative, (non-functional) requirements
- Capture goals, objectives, alternatives and rationales

UCM

- For operational and functional requirements
- Enables analysis and transformations
- Architectural alternatives and dynamic systems

Ongoing Work on URN at U. of Ottawa



- URN Meta-model and Semantics
- UCM Scenarios to MSC, UML, TTCN
- URN and Requirements Management (DOORS)
- URN for Reverse Engineering
- URN and Requirements-based Design (synthesis of SDL and LOTOS specifications from UCMs)
- URN and Performance Engineering (UCM2LQN)
- UCM for Feature Interaction Detection
- Tool Development
- Case Studies



Related Work on RE Languages/Notations



ITU-T

- MSC, SDL, eODL, UML profiles

OMG

- UML 2.0

ISO/IEC JTC1 SC7

- High-Level Petri Nets
- UML 1.4.1
- UML profile for EDOC, UML for ODP viewpoints

FIPA

- Tropos and UML for multi-agent systems



Two events...

URN Focus Group Meeting

Tomorrow morning, 9:00-12:00

McMaster U., ITB 225



7th Feature Interaction Workshop

Ottawa, June 10-13, 2003

<http://www.site.uottawa.ca/fiw03/>

