



INTRODUCTION TO THE USER REQUIREMENTS NOTATION

CSRS'03, May 27, 2003

Daniel Amyot


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

damyot@site.uottawa.ca

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



Overview and Agenda

- 
- Part I: **ITU-T**
 - Part II: **User Requirements Notation (URN)**
 - What is URN?
 - What can we model with URN?
 - What answers can these models provide?
 - What are the typical/potential usages?
 - Break
 - Part III: **Work!**
 - Metamodel concepts
 - Overview of Z.150 and requirements
 - Rationalization of notation elements
 - GRL propagation and modular descriptions
 - UCM actors and traversal



Part I: ITU-T



About ITU-T

International Telecommunication Union

- ITU-T: Telecom Standardization
- ITU-R: Radio-communication
- ITU-D: Telecom Development

United Nations organization, 190 countries

- Industry Canada is our official representative

Hundreds of sector members and others

Headquarters in Geneva



About ITU-T



Web site:

- <http://www.itu.int/>
- <http://www.itu.int/ITU-T/studygroups/com17/index.html> (SG17)

ITU-T Guide for Beginners:

- <http://www.itu.int/ITU-T/promotion/>
- Concise and rather complete coverage of all there is to know about ITU-T
- Structure, roles, processes
- Many useful links to the Web site

ITU-T standards sold on Web site

Free download of 3 standards

- <http://www.itu.int/publications/bookshop/how-to-buy.html#free>



ITU-T: 12 Study Groups



- SG 2 - Operational aspects of service provision, networks and performance
- SG 3 - Tariff and accounting principles including related telecommunications economic and policy issues
- SG 4 - Telecommunication management, including TMN
- SG 5 - Protection against electromagnetic environment effects
- SG 6 - Outside plant
- SG 9 - Integrated broadband cable networks and television and sound transmission
- SG 11 - Signalling requirements and protocols
- SG 12 - End-to-end transmission performance of networks and terminals
- SG 13 - Multi-protocol and IP-based networks and their internetworking
- SG 15 - Optical and other transport networks
- SG 16 - Multimedia services, systems and terminals
- SG 17 - Data Networks and Telecommunication Software**
- SSG - Special Study Group "IMT-2000 and Beyond"



SG17 Questions for Study



There are 5 Working Parties in Study Group 17.

Each WP is responsible for a set of related questions.

WP 1/17 – Data Networks

- Q1-Q6 on QoS, numbering, interworking, Frame Relay, network performance...

WP 2/17 – Open Systems Technology

- Q7-Q11 on IP lower layers, QoS multicast, directory services, security, OSI revision...



SG17 Questions for Study (relevant to WG19)



WP 3/17 – Languages and Notations

- Q12-Q18, and Q28 on ASN.1, SDL, Encoding of SDL Data, MSC, SDL Data in MSC, UML Combined to ITU-T Languages, **URN**, and language coordination

WP 4/17 – Quality and Methods

- Q19-Q23 on testing approaches, testing languages (TTCN), QA, quality aspects of Recommendations, time and performance

WP 5/17 – Open Systems Technology

- Q24-Q27 on ODL, DCL, **ODP**, and middleware



SG 17 Standards



- E.104, E.115 (in conjunction with SG 2)
- F.400-series, F.500-F.549, and F.600-series
- Q.933 and Q.933bis
- X-series, with the exception of those under the responsibility of Study Groups 4 (X.160-series, X.170-series and X.700-series), 15 (X.50-series) and 16 (X.26/V.10 and X.27/V.11).
- Z-series (MSC, SDL, TTCN, eODL, URN...)



Alternative Approval Process (AAP)



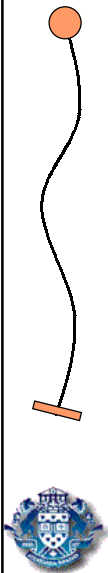
- Mature drafts sent for “consent” to study group
- If accepted, posted on the ITU Web site for a 4-week period for electronic comments
- If no comment, then approved as Recommendation
- However, we need consensus to be approved...



Part II: 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

Terminology...



- A *goal* is an objective or concern used to discover and evaluate functional and non-functional requirements.
- A *functional requirement* is a requirement defining functions of the system under development
- A *non-functional requirement* is a requirement characterizing a system property such as expected performance, robustness, usability, maintainability, etc. NFRs capture business goals/objectives and product quality attributes.
- A *user requirement* is a desired goal or function that a user and other stakeholders expect the system to achieve
- A *scenario* is a partial description of system usage defined as a set of partially-ordered responsibilities a system performs to transform inputs to outputs while satisfying preconditions and postconditions



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 Milestones



Z.150 (URN)

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

Z.151 (GRL) consent

- September 2003

Z.152 (UCM) consent

- September 2003

Z.153 (Methodological Approach) first draft

- September 2003

Z.153 (Methodological Approach) consent

- March 2004 (delayed from Sept. 2003)

Z.159 (UML profile for URN) consent

- 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

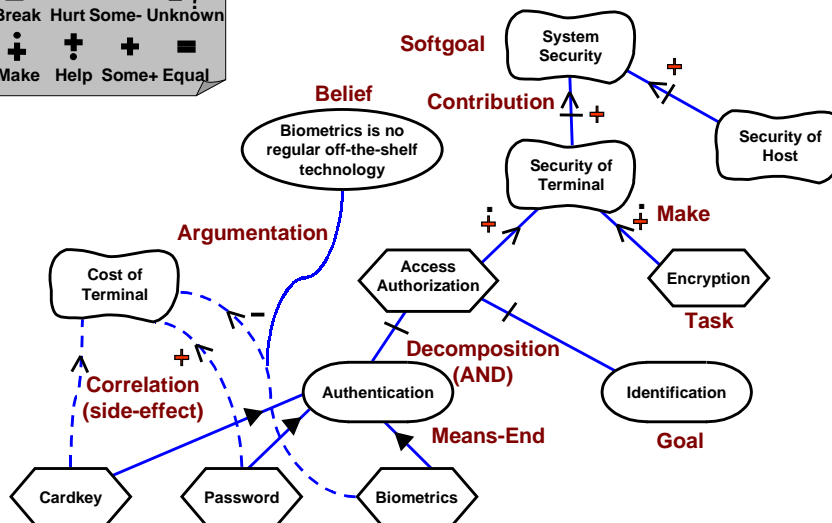
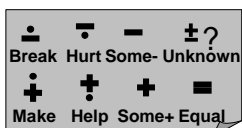


Why GRL?

- Goals become an **important driver** for requirements elaboration. Yet, stakeholders goals and objectives are complex and *will* conflict...
- GRL **expresses and clarifies** tentative, ill-defined and ambiguous requirements
 - Supports argumentation, negotiation, conflict detection & resolution, and in general decisions
 - Captures decision rationale and criteria (documentation!)
- GRL identifies **alternative** requirements and alternative system boundaries
- GRL provides clear **traceability** from strategic objectives to technical requirements
- GRL allows **reuse** of stable higher-level goals when the system evolves
- Nothing like this in UML...



Basic GRL Notation



Basic GRL Notation



Goal

- Quantifiable (often functional)

Softgoal

- Qualifiable but not measurable (often non-functional)

Task

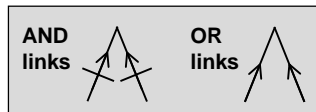
- Solution which achieves goals (means-end) or **satisfice** softgoals (contribution, correlation)

Belief

- Captures rationale

And/Or Link

- Contribution & correlation links may be typed AND or OR



Basic GRL Notation



Contribution

- Link for tasks, softgoals, beliefs, and links
- May be qualified (see symbols to the right)

Correlation

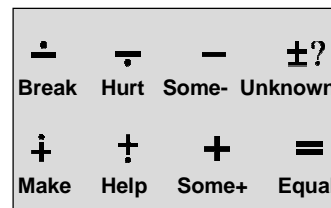
- Same as contribution but indicates side-effect

Means-End

- Link for tasks achieving goals, always OR

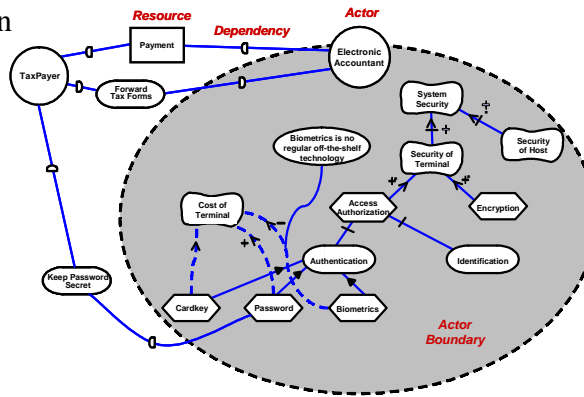
Decomposition

- Defines what is needed for a task to be performed (refinement), always AND



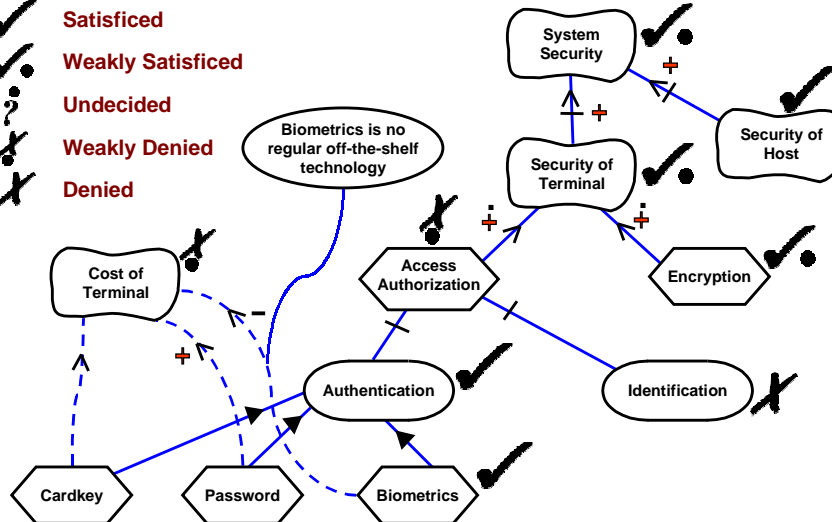
Advanced GRL notation (for your information only)

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



Evaluations with GRL

- ✓ Satisfied
- ✓ Weakly Satisfied
- ⊕ Undecided
- ✗ Weakly Denied
- ✗ Denied



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



UCMs in a Nutshell



Use Case Maps

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

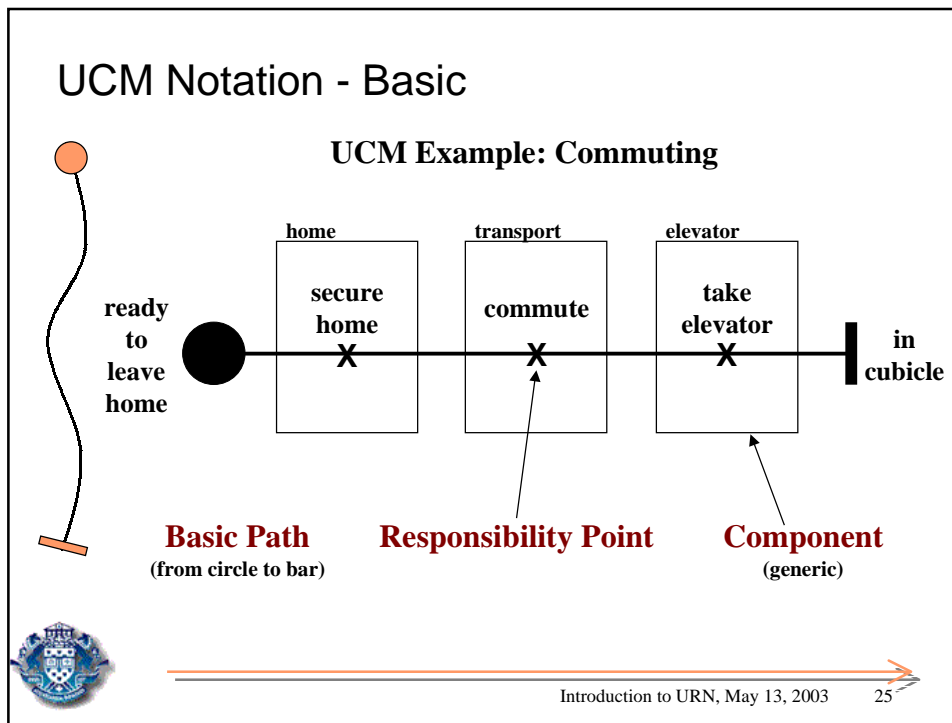
UCMs model the “what” aspects

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

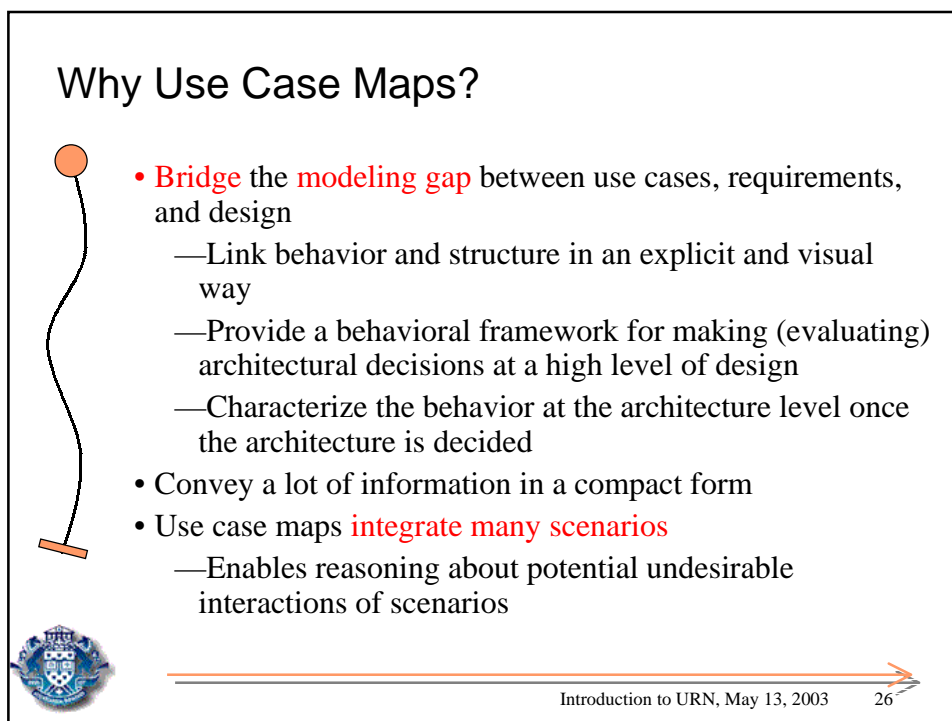
Performance analysis, conflict detection



UCM Notation - Basic



Why Use Case Maps?



Why Use Case Maps?



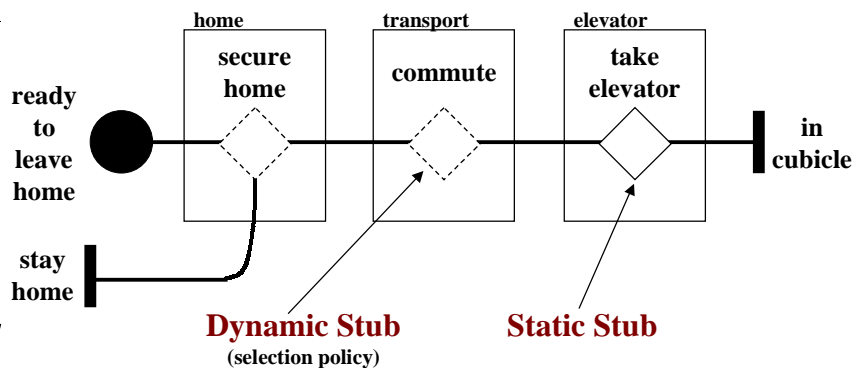
- Provide ability to model **dynamic systems** where scenarios and structures may change at run-time
 - E-commerce applications
 - Telecommunication systems based on agents
- Fairly simple, intuitive, low learning curve
- Document while you design
- Effective learning tool for people unfamiliar with the domain
- May be **transformed** (e.g. into MSC/sequence diagrams, performance models, test cases)



UCM Notation - Hierarchy

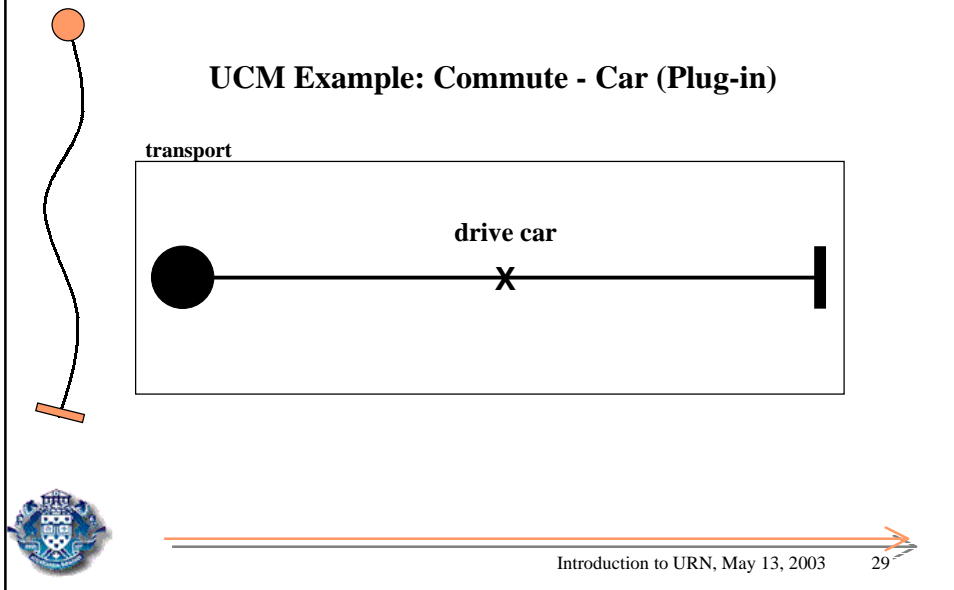


UCM Example: Commuting



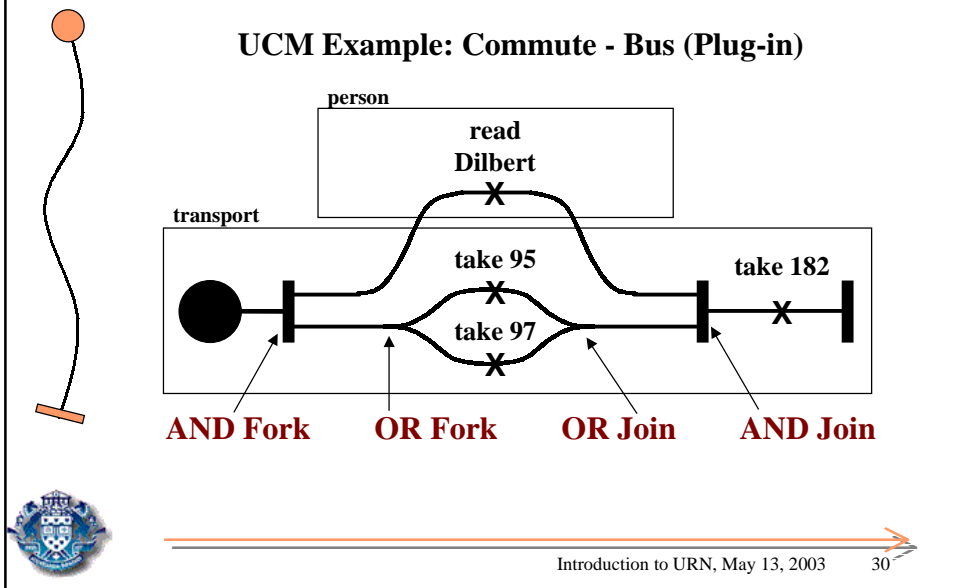
UCM Notation - Simple Plug-in

UCM Example: Commute - Car (Plug-in)

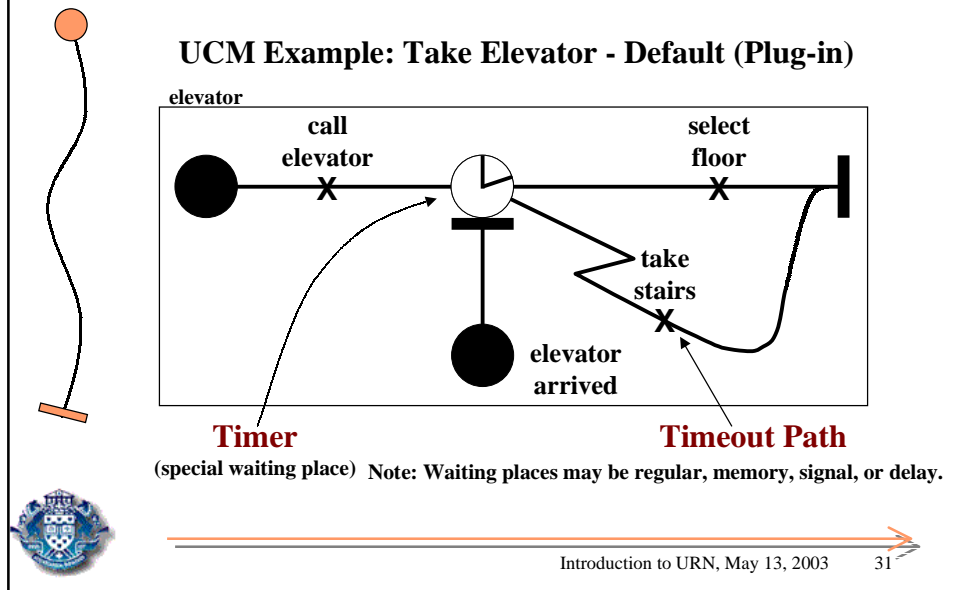


UCM Notation - AND/OR

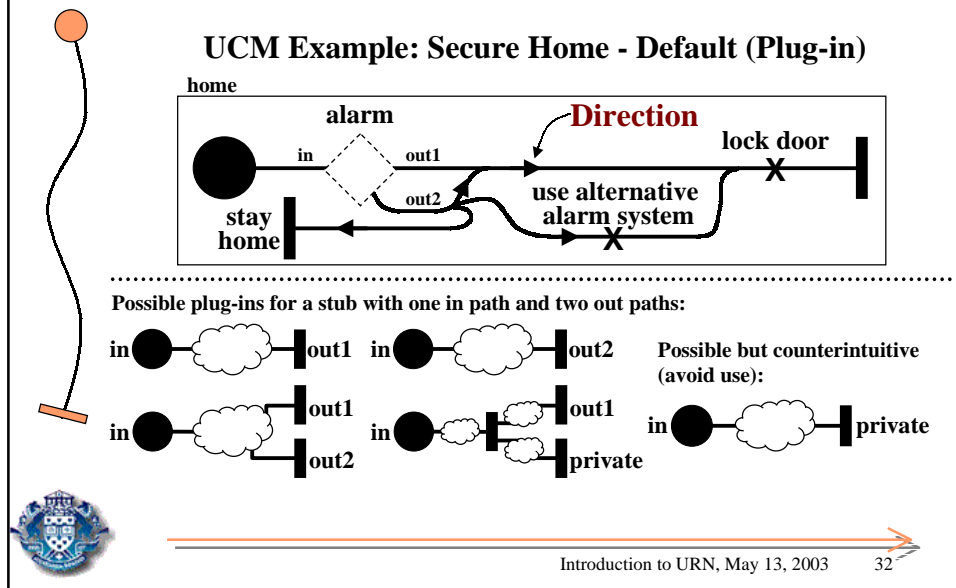
UCM Example: Commute - Bus (Plug-in)



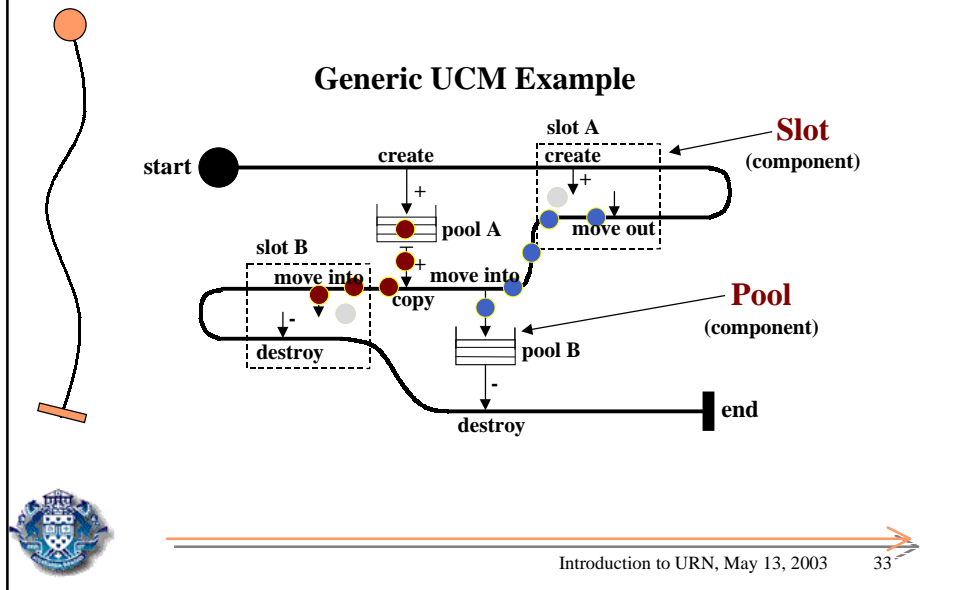
UCM Notation - Waiting Place / Timer



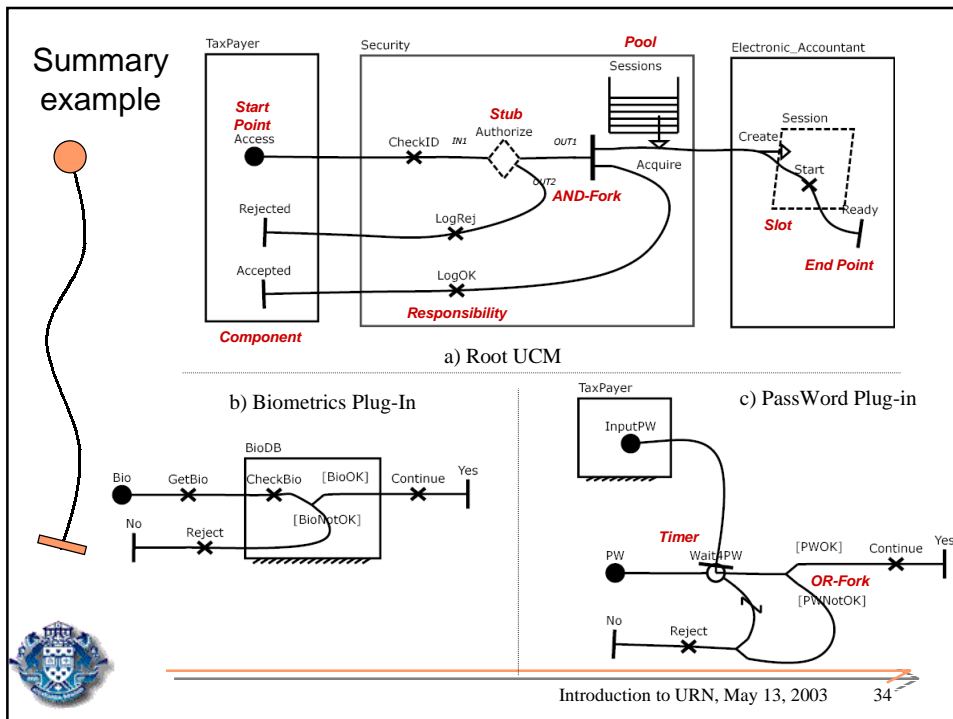
UCM Notation - Simple Plug-in with Stub



UCM Notation – Pools and Dynamic Responsibilities



Summary example



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



Typical Usage of URN



Modelling and documentation

- User and system requirements, rationales

Analysis of business goals

- Evaluations of alternative requirements or solutions
- Discovery of tradeoffs that can optimize the stakeholders' degree of satisfaction for conflicting goals

Architecture analysis

- Based on NFRs and design constraints
- Performance analysis

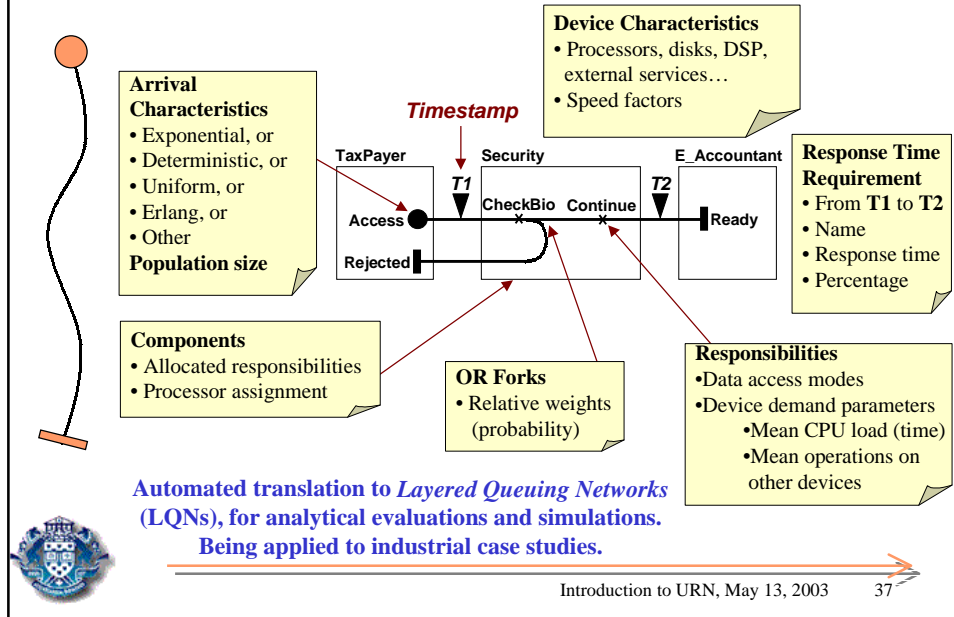
Generation of individual scenarios

- Training, documentation
- Detection of conflicts
- Transformation to MSC and test cases

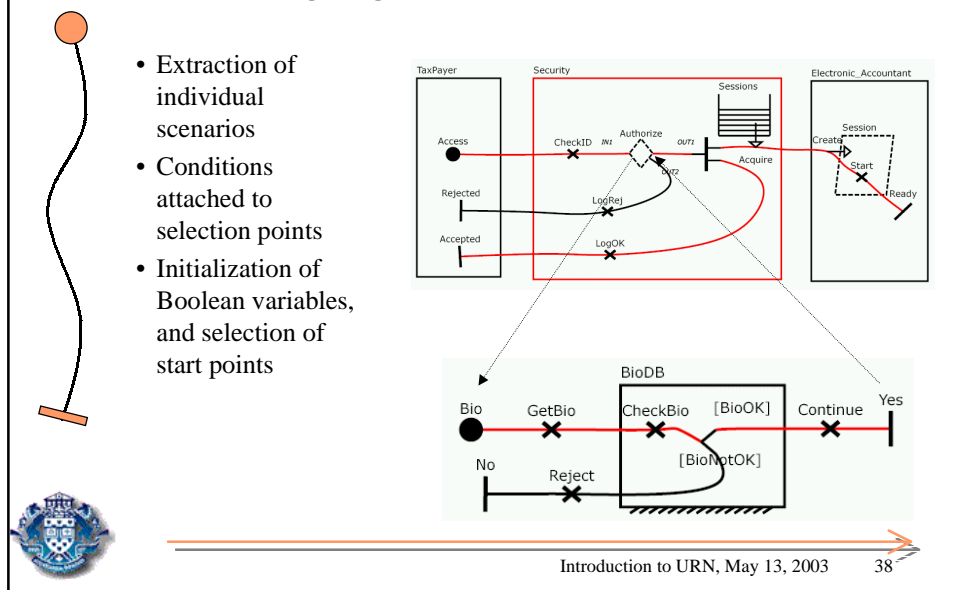
Reverse-engineering



Quantitative Performance Engineering with UCMs



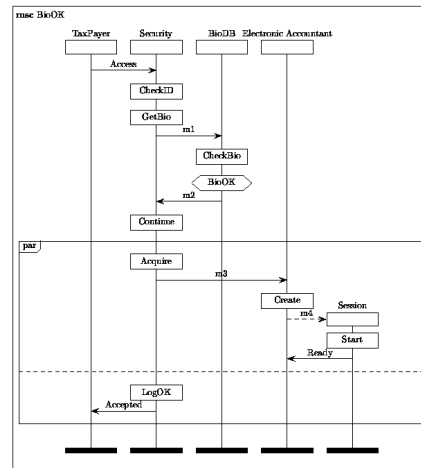
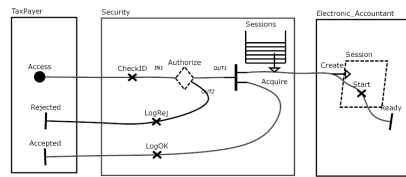
UCM Scenario Definitions and Path Traversal (Highlight)



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...)

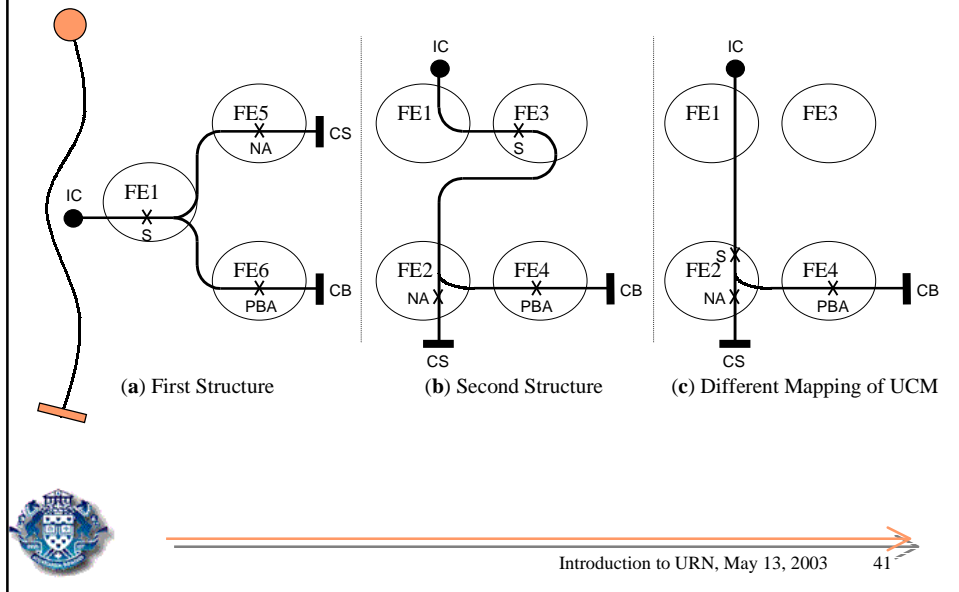


Key Points - Scenario Definitions

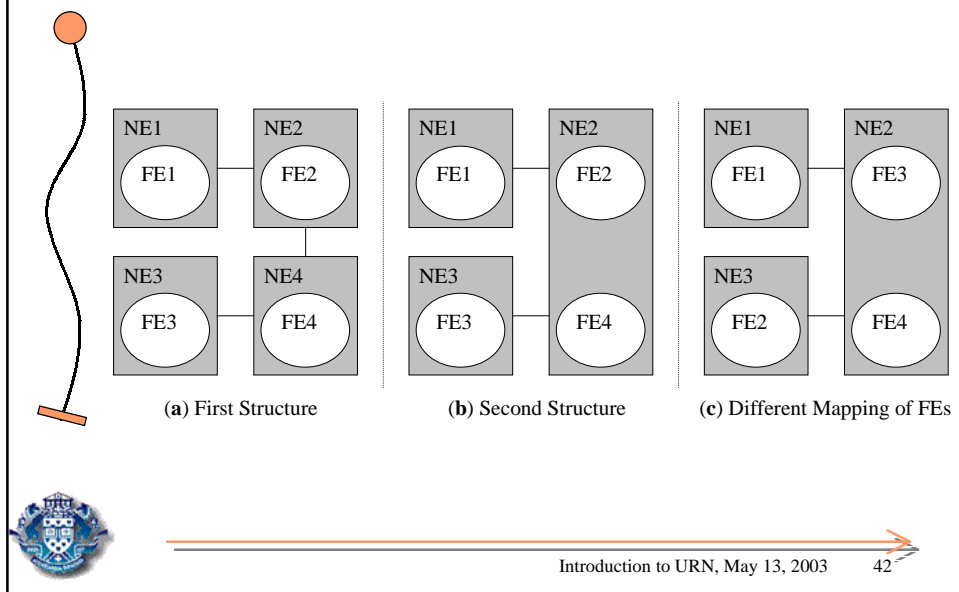
- Path data model is not a problem domain data model
- Improves understanding of scenarios
- Scenario definitions are the foundation for more advanced functionality based on UCM path traversal mechanisms
- Many correct path traversal mechanisms exist because of concurrency that is not well nested
- Scenario definitions and path traversal is instrumental for advanced functionality such as highlighting, animation, and generation of MSC, LQN, TTCN, ...
- Much value in a tool-supported translation



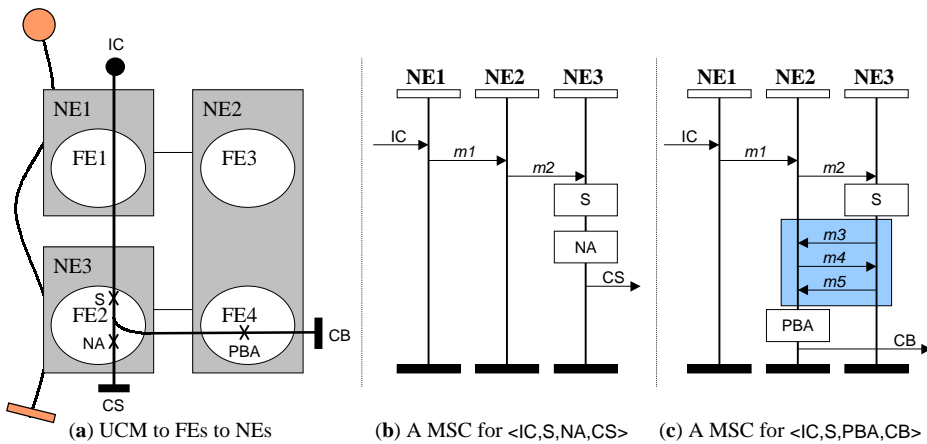
Architectural Alternatives (WIN Scenario)



Binding Functional Entities to Network Entities (WIN)



Refinement with MSCs (WIN)



Tool Support: UCMNav 2.1 (free, + source)

Use Case Map Navigator: SG10demo.ucm

File Components Options Performance Maps Align Utilities Scenarios About

Map Title: Terminating: Default plug-in for ASP

Navigation Mode: Scenario Trace: CHNdisplay (CND)

Component Path Select Scale Editing Mode Decomposition Level

Responsibilities

- busyTreatment: This user is busy and cannot answer. Prepare busy signal.

Scenario Definitions

Scenario Groups

- Basic Call
- OCS
- TeeLine
- CND
- OCS_CND
- TL_TL_CND

Scenarios

- OCS_CHNdisplay

Selected Scenario

Boolean Variables (unreferenced)

- PinValid
- TlActive

Variable Initializations

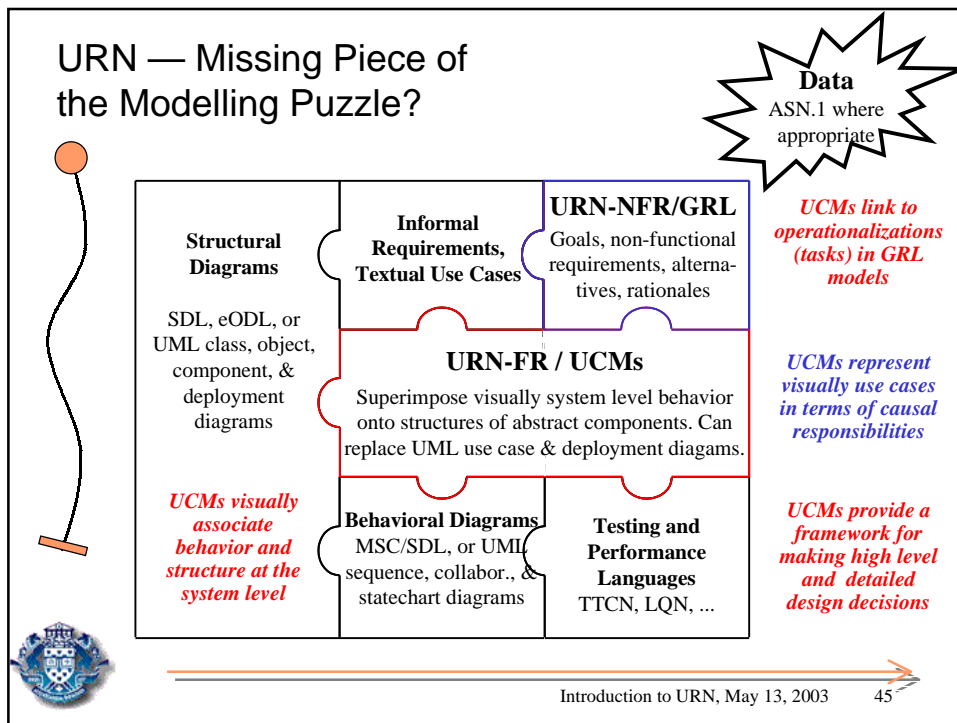
- OnOCSList = F
- subCHD = T
- subOCS = T
- subTL = F
- Busy = F

Scenario Starting Point

- Path Start: req
- Map: root

Buttons: Add Selected Path Start, Generate MSC, Highlight Path, Duplicate Scenario, Remove Dialog

URN — Missing Piece of the Modelling Puzzle?



Key Points - URN Puzzle

- Goal-based (e.g. GRL) and scenario-based (e.g. UCMs) notations complement each other
 - URN fills a void in UML and ITU-T languages
 - UCMs offer more capabilities than UML use case diagrams and activity diagrams
 - URN fits well into scenario-based software development methodologies
 - GRL provides the decision making framework for software engineering activities
 - URN supports early activities in software development, bringing together stakeholders with expertise in many different areas
 - UCMs provide a good basis for design-time feature interaction detection and for model construction
 - UCMs and GRL can be used *iteratively* and *independently*
- Introduction to URN, May 13, 2003 46

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



Part III:



What are the main GRL concepts (metamodel)



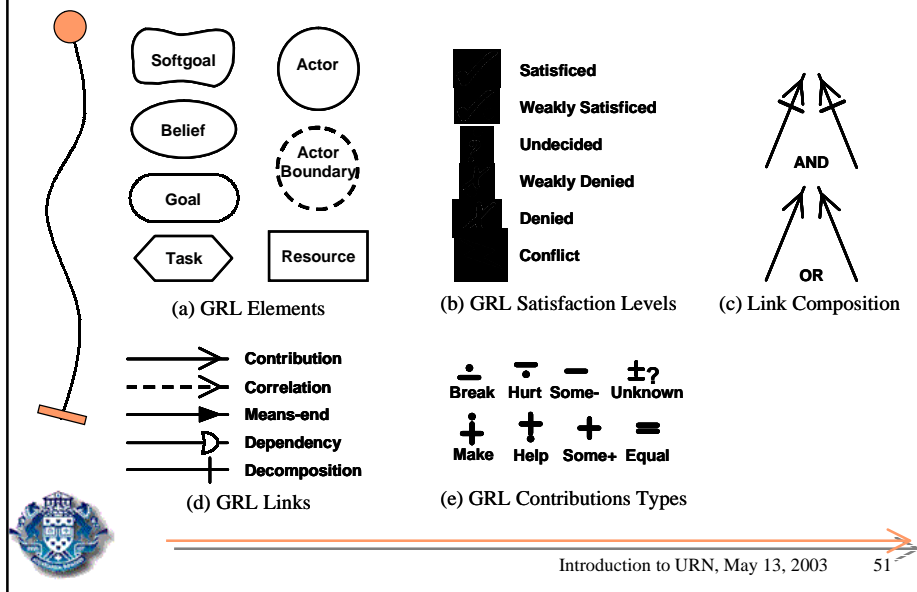
- GRL model
- Intentional elements
 - Goal
 - Operationalisation (task)
 - Belief
 - *Resource and softgoal?*
- Contribution (with weight/type)
- Composition (AND/OR)
- Actor
- Dependency
- Evaluation (argumentation)

Do we need all these GRL symbols?

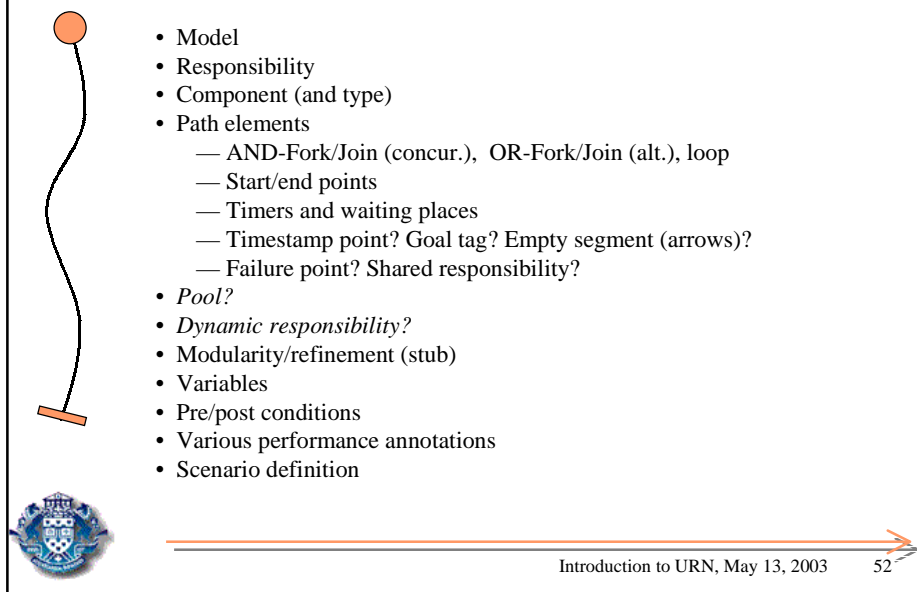


- Goal/softgoals distinction
- Resources, non-intentional element
- Correlation and means-end
- New symbols for contribution types?
- New symbols for weakly denied/satisfied?
- Better way to present link composition (AND-OR)
 - e.g. use of intermediate node for AND (KAOS)
- Criticality (of non-intentional elements)?

Summary of GRL Notation (Partial)



What are the main UCM concepts (metamodel)



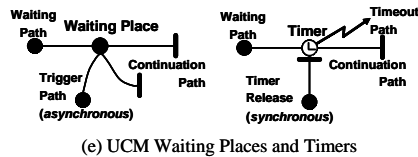
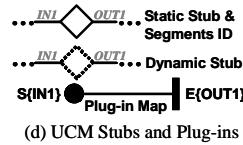
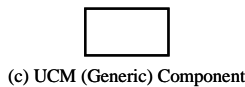
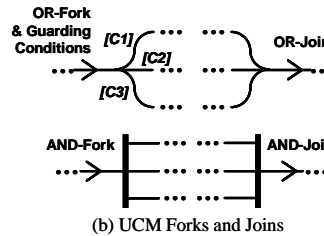
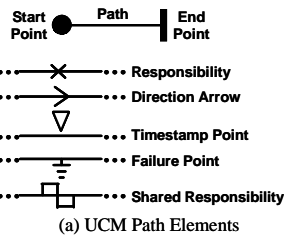
Do we need all these UCM symbols?



- Abort
- Failure point
- Pool
- Dynamic responsibility and dynamic component
- Shared responsibility
- Shared stub



Summary of UCM Notation (Partial)



Are concepts/symbols missing?



GRL

- Obstacle analysis?
- Modular descriptions?
 - At the moment, GRL models are flat models

UCM

- Actor (already a GRL concept)
 - Different symbol(s)? Human/machine?
 - Scenario initiator/participant?
- Bindings attached to start/end points in plug-ins
- Four-variable model?
- Policies (deontic modalities)?



GRL Evaluations



“a new algorithm written by J. Mylopoulos and an Italian group, which is fully automated and does not require user intervention, is now available and will be studied”

May 2002.

Any developments there?



GRL Evaluations



- *Evaluations* of GRL model using
- Generic propagation algorithm with:
 - Evaluation functions for satisfaction level
 - Contributions type/weight
 - Degrees of satisfaction
 - Composition operators
 - Detection of conflicts (interactive/automated propagation)
 - Detection of incomplete initial marking
- Propose two sets of functions (fuzzy and numerical) for
 - Contributions type/weight (help, hurt, etc; -1..1)
 - Degrees of satisfaction (satisfied, denied, etc; -1..1)
 - Composition operators (table; min/max)
- Room for a multi-value logic?

UCM Traversal



- Path traversal guidelines already in place
- Need refinement for synchronization
- Scenario definitions with same start point triggered multiple times,
- Better postconditions in scenario definitions
- More complex path data model (integers? time?)?

Thank you!



Thank you for your precious input!

For more information, please visit

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

