

Overview of the User Requirements Notation (URN)

Daniel Amyot, Gunter Mussbacher
 Mitel Networks
 Daniel_Amyot@mitel.com

Objectives

- ◆ Give an overview of the User Requirements Notation (URN)
- ◆ Understand how URN fits into use-case driven software development processes
- ◆ Give an overview of the relationship of the two notations URN consists of - the Goal-oriented Requirement Language (GRL) and Use Case Maps (UCMs)
- ◆ Compare and contrast Use Case Maps with other modeling techniques which focus on the same general problem area

Table of Contents

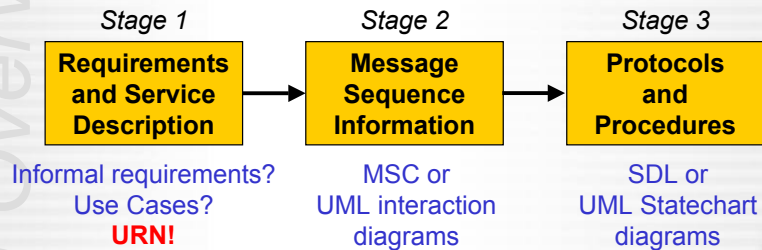
◆ Overview of User Requirements Notation (URN)

- ◆ URN Puzzles

About ITU-T

- ◆ The International Telecommunication Union - Telecommunication Standardization Sector (ITU-T)
- ◆ ITU is a United Nation organization (189 members)
- ◆ 13 Study Groups in ITU-T
- ◆ SG17: Data Networks and Telecommunications Software (now merged with SG7, formerly SG10)
- ◆ 27 questions for study in SG17 (MSC, SDL, UML...)
- ◆ Q18/17: **URN: User Requirements Notation**
 - Create a standard by end of 2003

Motivation for URN (ITU-T SG17 Question 18)



- ◆ Common design and standardisation methodologies already use scenarios
- ◆ Need improvement to cope with new realities of complex, dynamic, and evolving systems

URN - Initial Requirements

- ◆ Focus on early stages of design, with scenarios
- ◆ Capture user requirements when little design detail is available
- ◆ No messages, components, or component states required
- ◆ Reusability of scenarios and allocation to components
- ◆ Dynamic refinement capabilities
- ◆ Modelling of agent systems, early performance analysis, and early detection of undesirable interactions

URN - Additional Requirements

- ◆ Express, analyse and deal with **non-functional requirements** (NFRs)
- ◆ Express the relationship between business objectives/goals and system requirements
- ◆ Capture reusable analysis (argumentation) and design knowledge (patterns) for addressing non-functional requirements
- ◆ Connect to other ITU-T languages
- ◆ **As of now, NOTHING has been cast in stone**

Current Proposal for URN

- ◆ Draft documents for Z.150, Z.151, Z.152
 - <http://www.UseCaseMaps.org/urn/>
- ◆ Combined use of two notations
 - **Goal-oriented Requirement Language (GRL)** for NFRs (<http://www.cs.toronto.edu/km/GRL/>)
 - **Use Case Maps (UCM)** for Functional Requirements (<http://www.UseCaseMaps.org/>)
- ◆ Contribution includes:
 - Refined requirements, traceability relationships, and specifications for both languages
 - Compliance statements, DTD, examples, tutorials
- ◆ Create ITU-T standard by end of 2003 (Z.150-153)

Table of Contents

- ◆ Overview of User Requirements Notation (URN)

◆ URN Puzzles

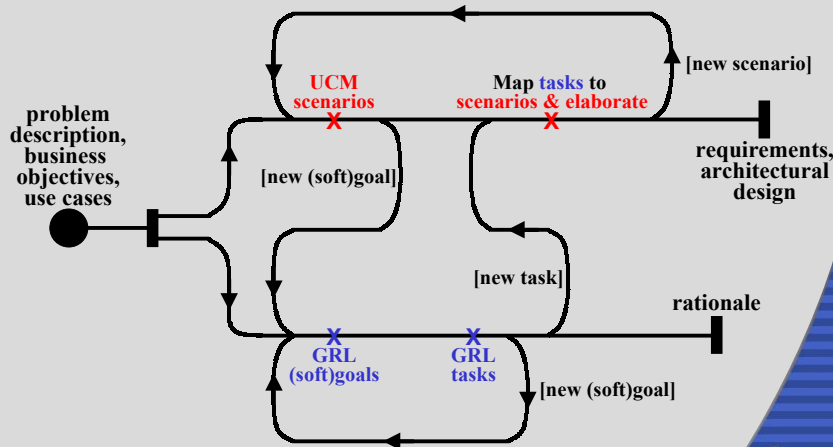


GRL - UCM Relationship

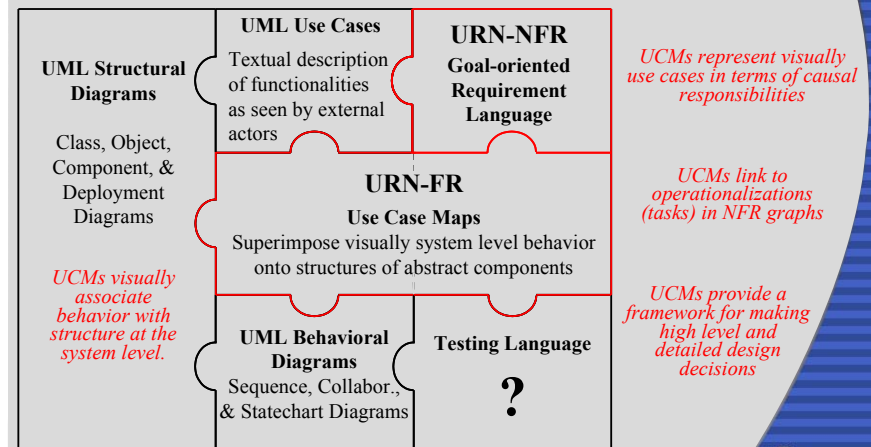
- ◆ Goal-based approach
 - Focuses on answering “why” questions
 - Addresses functional and non-functional requirements
- ◆ Scenario-based approach
 - Focuses on answering “what” questions
- ◆ Goals are **operationalized** into tasks and tasks are elaborated in (mapped to) UCM scenarios
 - Focuses on answering “how” questions



GRL - UCM Relationship



URN - UML Puzzle



UCM - UML

- ◆ UCMs express almost all concepts of use case diagrams and all of activity diagrams
- ◆ Use case diagrams
 - UCMs express all concepts except actors and actor generalization; these concepts, however, could be modeled after minor changes to UCMs
 - UCMs show more precisely the location of and circumstances for extensions to use cases and for common use cases
 - How useful are use case diagrams?

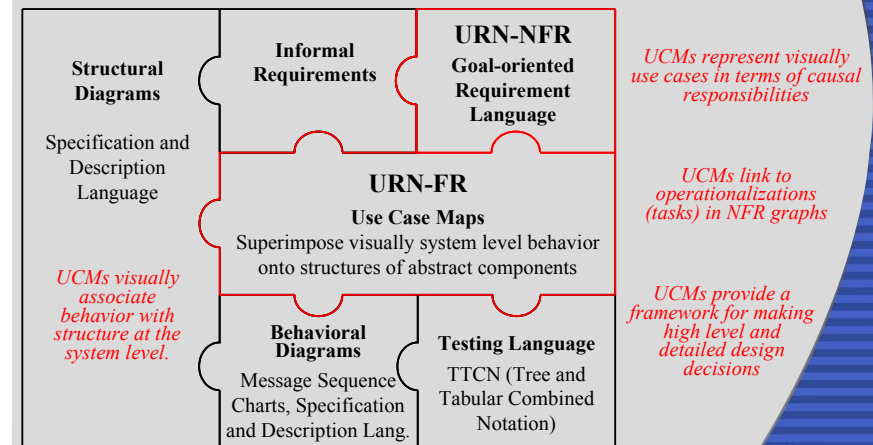
UCM - UML

- ◆ Activity diagrams
 - UCMs provide dynamic stubs & selection policies
 - ◆ Selection policies specify which plug-in(s) to choose dynamically depending on preconditions and whether the plug-ins' execution occurs concurrently or sequentially
 - UCMs provide powerful dynamic capabilities
 - UCMs provide timers and timeout paths
 - UCMs allow several start points per map and several in-paths per stub
 - UCMs have a much more flexible and expressive mapping of structure to behavior

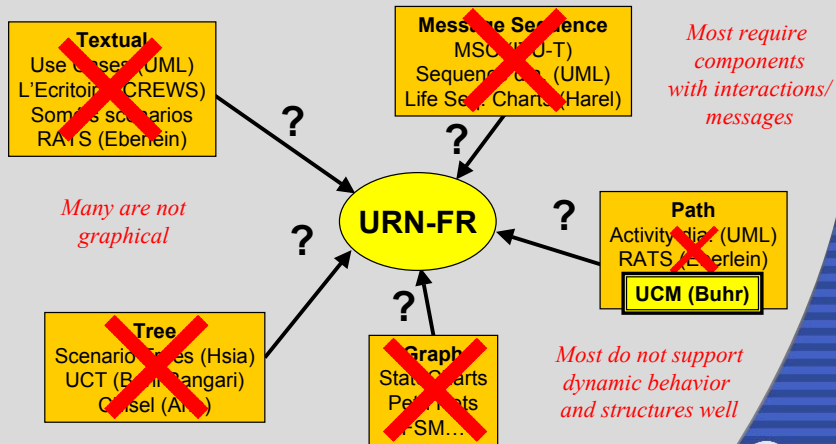
UCM - UML

- ◆ UCMs are at a higher abstraction level than sequence and collaboration diagrams
 - Causality instead of messages
 - Integrated scenarios instead of snapshots
- ◆ UCMs are at a higher abstraction level than statechart diagrams
 - Not focussed on component states and messages
- ◆ UCM components may be mapped to UML structural diagrams

URN - ITU-T SG17 Puzzle



Candidate Scenario Notations For URN-FR



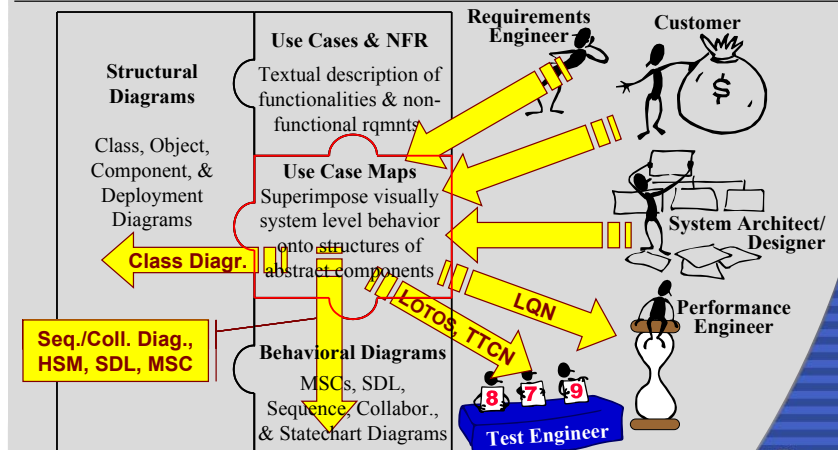
Comparing Scenario Notations

		Scenarios				Components		
		Flow	Concurrency	Integration	Dynamicity	Actors	Structure	Layout
Z-120	MSC	Messages	Yes	No	Timers	No	Flat	Linear
	HMSC	Causal	Yes	Some	No	No	No Components	
UML	Activity Diag.	Causal	Yes	Some	No	No	Flat	Linear
	Collaboration Diag.	Messages	Labels	No	No	Yes	Flat	2D
	Sequence Diag.	Messages	No	No	No	Yes	Flat	Linear
	Use Case Diag.	None	No	Some	No	Yes	No Components	
	Statechart Diag.	Messages	Yes	Yes	No	No	One Component	
	Use Case Map	Causal	Yes	Yes	Yes	No	Hierarchy	2D

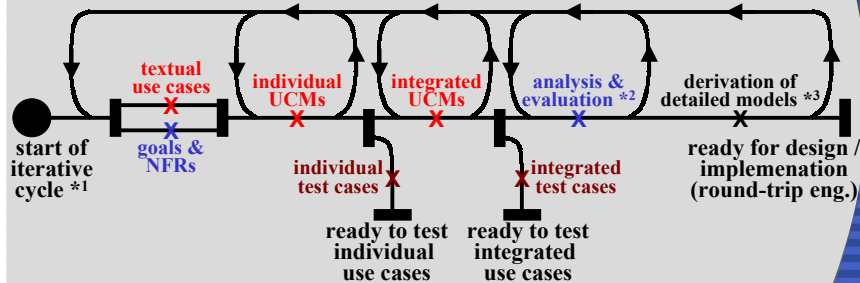
Comparing Scenario Notations

		Scope
Z-120	MSC	Snapshot of System-Level Behaviour and Components
	HMSC	System-Level Behaviour
UML	Activity Diag.	Snapshot of System-Level Behaviour and Components
	Collaboration Diag.	Snapshot of System-Level Behaviour and Components
	Sequence Diag.	Snapshot of System-Level Behaviour and Components
	Use Case Diag.	User-Level Behaviour
	Statechart Diag.	Component-Level Behaviour
	Use Case Map	User and System-Level Behaviour and Components

Scope of Use Case Maps



Sample URN-Based Process



*1 Preconditions depend on previous iterations if applicable and existing systems (UCMs, goals, and NFRs may have to be reverse engineered)

*2 Feature interaction detection, performance analysis, architectural evaluation

*3 e.g. Interaction diagrams, MSCs, class diagrams

Key Points - URN Puzzles

- ◆ Goal-based (e.g. GRL) and scenario-based (e.g. UCMs) notations complement each other
- ◆ GRL and UCMs, as part of the User Requirements Notation (URN), propose to fill a void in methodologies based on ITU-T languages
- ◆ UCMs offer more capabilities than use case diagrams and activity diagrams
- ◆ Compared to other scenario notations, UCMs are graphical, do not require components with interactions/messages, and support dynamic behavior and structures well

Key Points - URN Puzzles

- ◆ URN fits well into scenario-based software development methodologies
- ◆ GRL provides the decision making framework for software engineering activities
- ◆ UCMs become the focal point for 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 (tests, performance, MSC / Sequence Diagrams, LOTOS, and others)

Main References

- ◆ Web sites: <http://www.UseCaseMaps.org/> and <http://www.cs.toronto.edu/km/GRL/>
 - Amyot, D. and Mussbacher, G., *On the Extension of UML with Use Case Maps Concepts*, UML2000, York, UK, October 2000.
 - Buhr, R.J.A., *Use Case Maps as Architectural Entities for Complex Systems*, In: Transactions on Software Engineering, IEEE, Vol. 24, No. 12, December 1998, pp. 1131-1155.
 - Buhr, R.J.A. and Casselman, R.S., *Use CASE Maps for Object-Oriented Systems*, Prentice Hall, 1996.
 - Cameron, D. et al., *Draft Specification of the User Requirements Notation*, Canadian contribution to ITU-T, September 2001.
 - Liu, L. and Yu, E., *From Requirements to Architectural Design – Using Goals and Scenarios*, In: From Software Requirements to Architectures Workshop (STRAW 2001), Toronto, Canada, May 2001.