# User Requirements Notation (URN)

Gregor v. Bochmann, University of Ottawa

Based on Powerpoint slides by Gunter Mussbacher(2009)
with material from Amyot

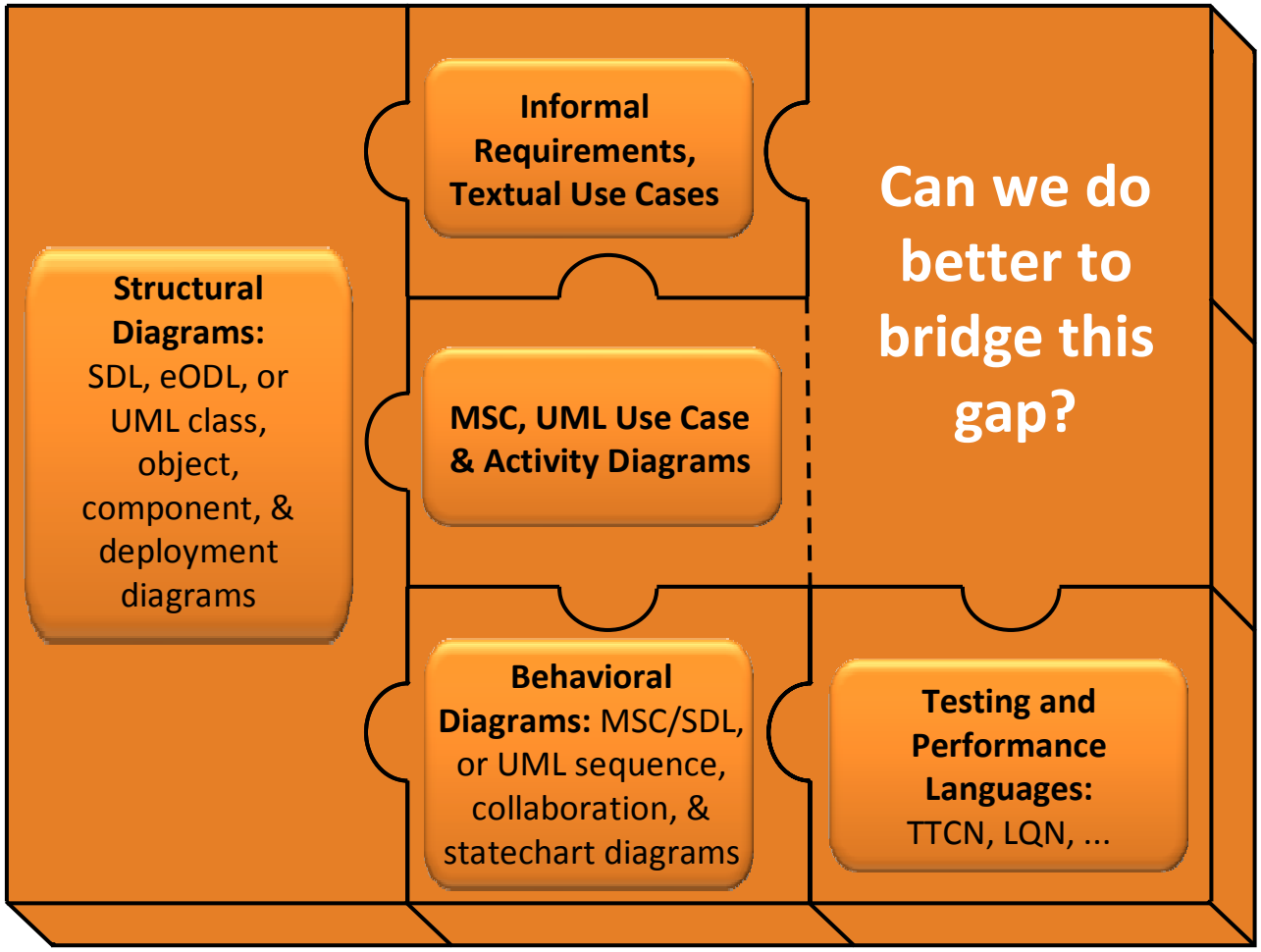
# Table of Contents



- All models are false but some models are useful.[1]

[1] George Edward Pelham Box (1919-)

# Modeling Puzzle – Common View




# Introduction to the User Requirements Notation (URN)

# URN Overview (1)

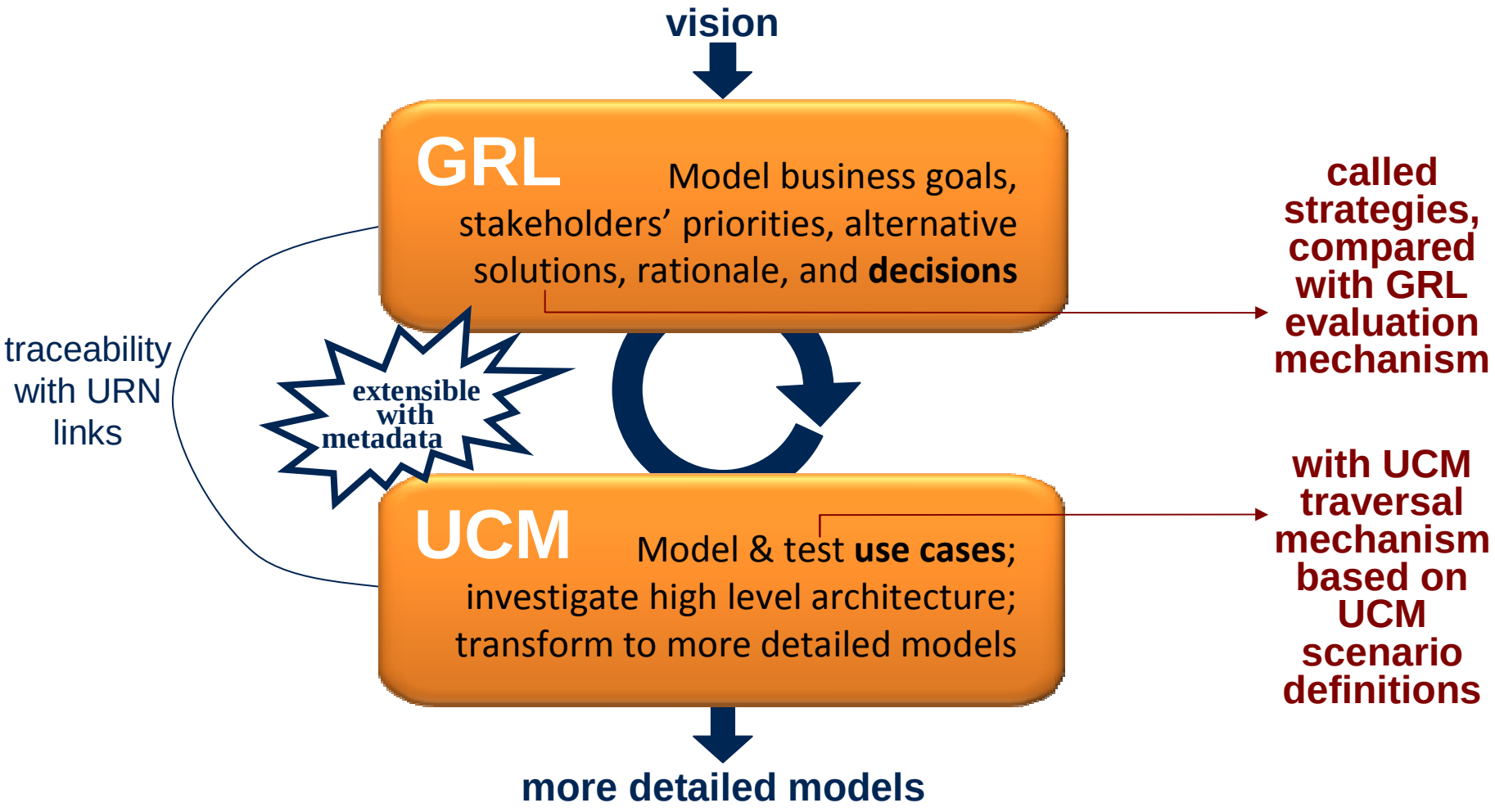
- URN is a semi-formal, lightweight graphical language for modeling and analyzing requirements in the form of goals and scenarios and the links between them
- Combines two existing notations
  - Goal-oriented Requirement Language (GRL) (based on i* & NFR Framework)
  - Use Case Maps (UCMs)
- Support for the elicitation, analysis, specification, and validation of requirements
- Allows systems, software, and requirements engineers to discover and specify requirements for a proposed system or an evolving system, and analyse such requirements for correctness and completeness

# URN Overview (2)

- URN models can be used to specify and analyze various types of reactive systems, business processes, and telecommunications standards

- jUCMNav
  - URN editor & analysis tool
  - Eclipse plug-in
  - Open source project

# URN Overview (3)




A GRL / UCM model visually communicates business objectives and constraints / high-level functional requirements to all stakeholders

# Combination of Goals and Scenarios

- URN combines two complementary parts:
  - The Goal-oriented Requirement Language (GRL)
    - For modelling goals and other intentional concepts (mainly for non-functional requirements, quality attributes, and reasoning about alternatives and tradeoffs)
  - The Use Case Map (UCM) notation
    - For modelling scenario concepts (mainly for operational requirements, functional requirements, and performance and architectural reasoning)
- URN offers concepts for the specification of stakeholders, goals, non-functional requirements, rationales, behaviour, scenarios, scenario participants (actors), and high-level architectural structure

# URN – Main Objectives

- Focus on early stages of development with goals and scenarios
- From user requirements to functional and non-functional system 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
  - Also to LQN, LOTOS, and others

# ITU-T Z.151: URN – Language Definition

- URN is the first and currently only standard which explicitly addresses goals (non-functional requirements with GRL) in addition to scenarios (functional requirements with UCMs) in a graphical way in one unified language
  - International Telecommunication Union (ITU-T Z.150 series)
  - ITU-T Z.150 (02/03):
    User Requirements Notation (URN) - Language requirements and framework
  - ITU-T Z.151 (11/08):
    User requirements notation (URN) - Language definition
- Part of the ITU family of languages: SDL, MSC, TTCN-3…
- Definition of URN in Recommendation Z.151 (approved November 2008)
  - Metamodel, abstract/concrete syntaxes, semantics…

ITU-T Z.150: http://www.itu.int/rec/T-REC-Z.150/en ITU-T Z.151: http://www.itu.int/rec/T-REC-Z.151/en

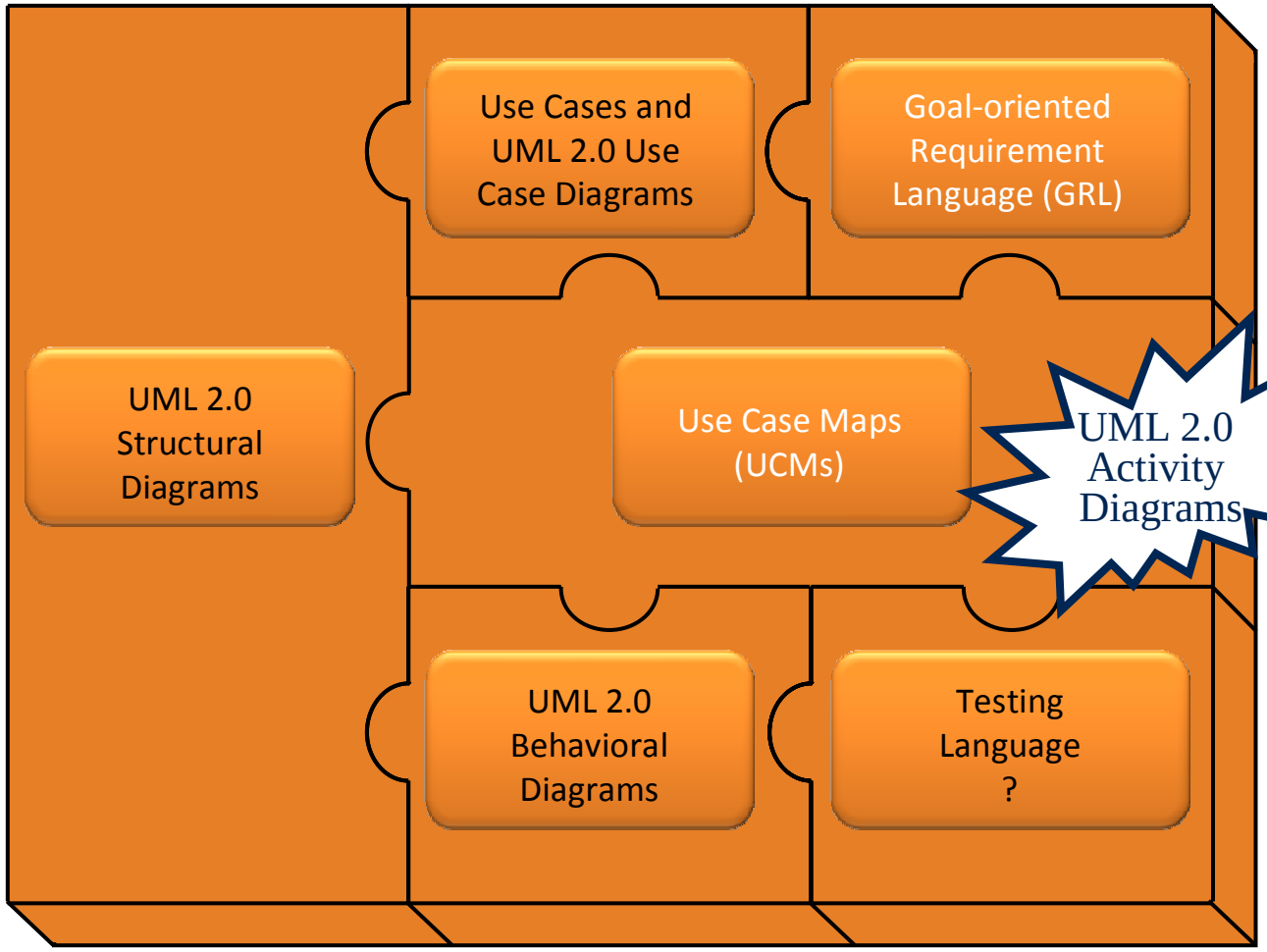
# About ITU-T

- International Telecommunication Union
- United Nations organization
  - 191 countries + hundreds of member companies
- Free access to the definition of many standard languages
  - http://www.itu.int/ITU-T/studygroups/com17/index.html
- Question 13 (Formal languages and telecommunication software) of Study Group 17 (Security)
  - Z.15x User Requirements Notation (URN)
  - Also SDL, MSC, and profiles
- Other international bodies standardize languages
  - ANSI (C, C++), W3C (HTML, XML), IEEE (VHDL), ISO (LOTOS), OMG (UML), OASIS (XACML), etc.

# ITU-T Languages

- SDL: Specification and Description Language
  - For defining and executing complete reactive systems
- MSC: Message Sequence Charts
  - For defining message-oriented scenarios
- ASN.1: Abstract Syntax Notation One
  - For defining data types/packets
- TTCN-3: Testing and Test Control Notation
  - For defining test cases and test environments
- URN: User Requirements Notation
  - Goal-oriented Requirements Language (GRL)
  - Use Case Map notation (UCM)
  - For defining abstract goals, scenarios, and requirements
- UML 2.0 profiles for some of these languages on their way…

# UML Puzzle




*GRL captures stakeholders' business goals, alternatives, decisions, and rationales.*

*Alternatives in GRL are linked with more detailed models described as UCMs.*

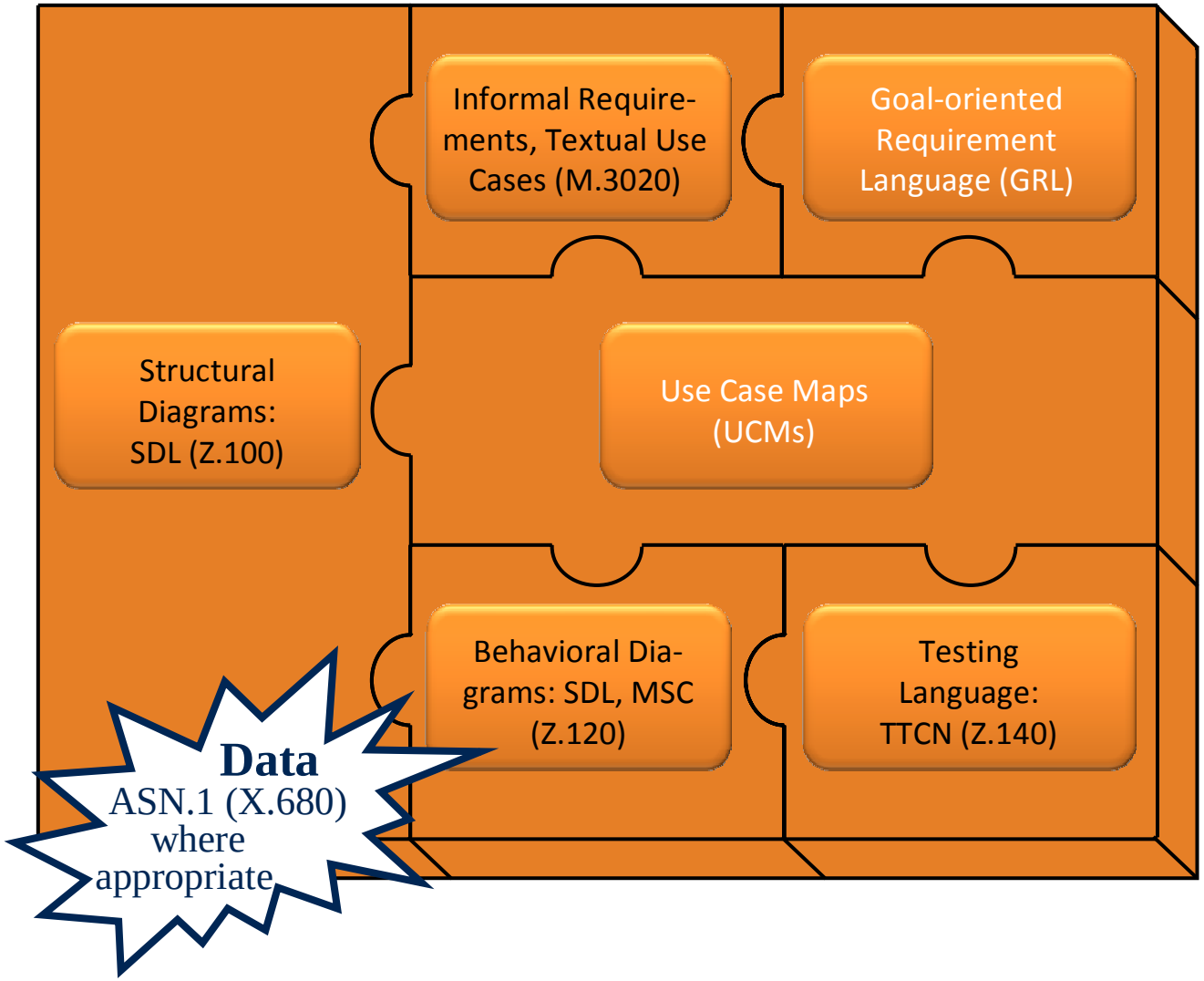
*UCMs represent visually use cases in terms of causal responsibilities.*

*UCMs provide a framework for making high level and detailed design decisions.*

*UCMs visually associate behavior with structure at the system level.*

*UCMs enable functional testing and allow the generation of test purposes.*

# ITU Puzzle




*GRL captures stakeholders' business goals, alternatives, decisions, and rationales.*

*Alternatives in GRL are linked with more detailed models described as UCMs.*

*UCMs represent visually use cases in terms of causal responsibilities.*

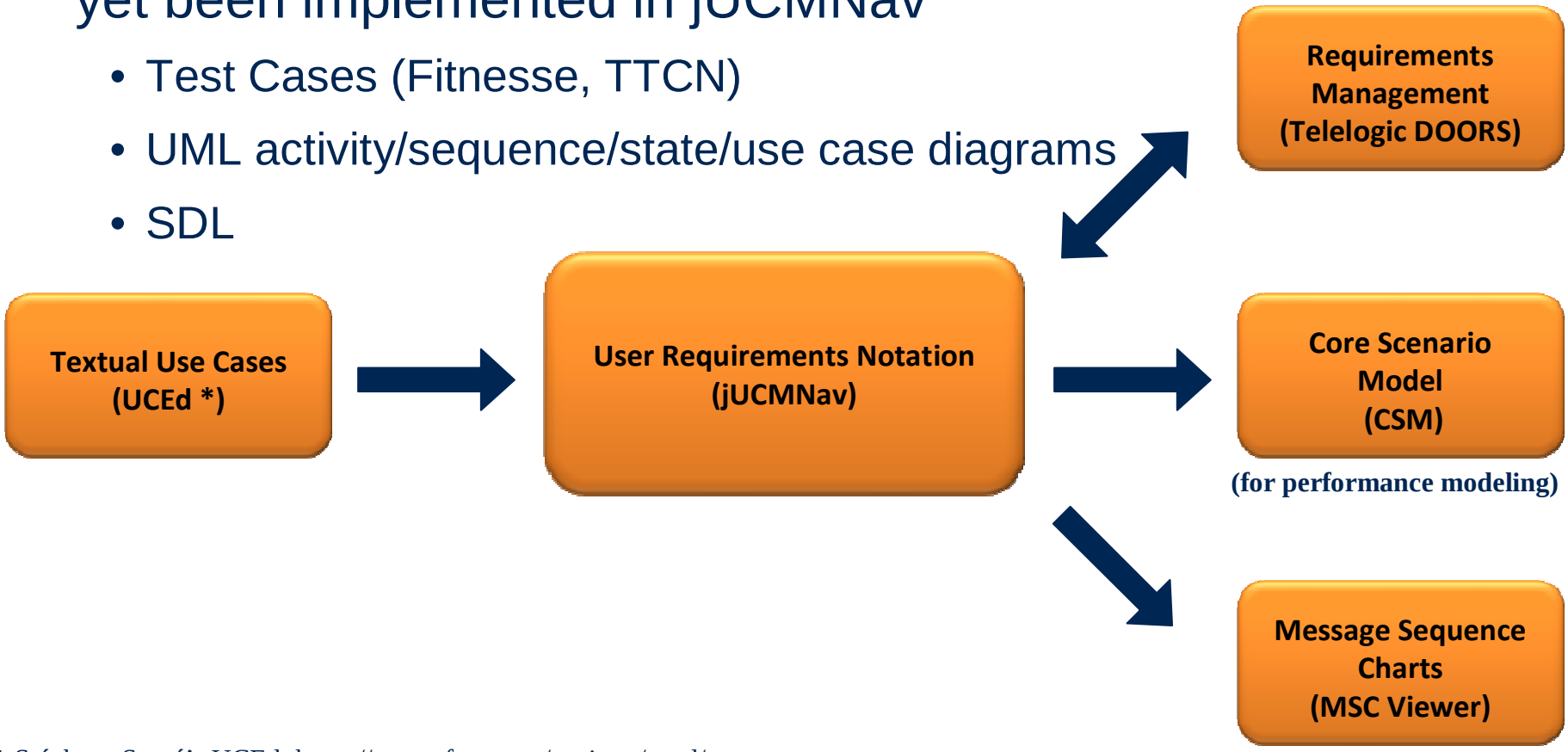
*UCMs provide a framework for making high level and detailed design decisions.*

*UCMs visually associate behavior with structure at the system level.*

*UCMs enable functional testing and allow the generation of test purposes.*

# Transformations

- Transformations connect URN models to other models
- Some transformations discussed in publications or implemented in jUCMNav's predecessor UCMNav have not yet been implemented in jUCMNav
  - Test Cases (Fitnesse, TTCN)
  - UML activity/sequence/state/use case diagrams
  - SDL




* Stéphane Somé's UCEd: https://sourceforge.net/projects/uced/

# Summary

- The User Requirements Notation (URN) is an ITU-T standard
- Modeling with the Goal-oriented Requirement Language
  - Focuses on answering "why" questions
  - Intentions, functional / non-functional requirements, rationales
- Modeling with Use Case Maps
  - Focuses on answering "what" questions
  - Scenarios, services, architectures
- While modelling with URN as a whole, goals are operationalized into tasks, and tasks are elaborated in scenarios (mapped to UCM path elements)
  - Moving towards answering "how" questions
  - Can guide the selection of an architecture or scenarios
- Enables the elicitation/specification of systems, standards, & products, analysis from various angles, & transformations

# URN Tool: jUCMNav – Juice Up Your Modeling!

- GRL
- Strategies
- Evaluation

- UCMs
- Scenario Definition
- Scenario Execution
- Test Suite

- MSC Generation
- Export to DOORS
- Export to CSM


**Pronounced: juicy – em – nav**

http://jucmnav.softwareengineering.ca

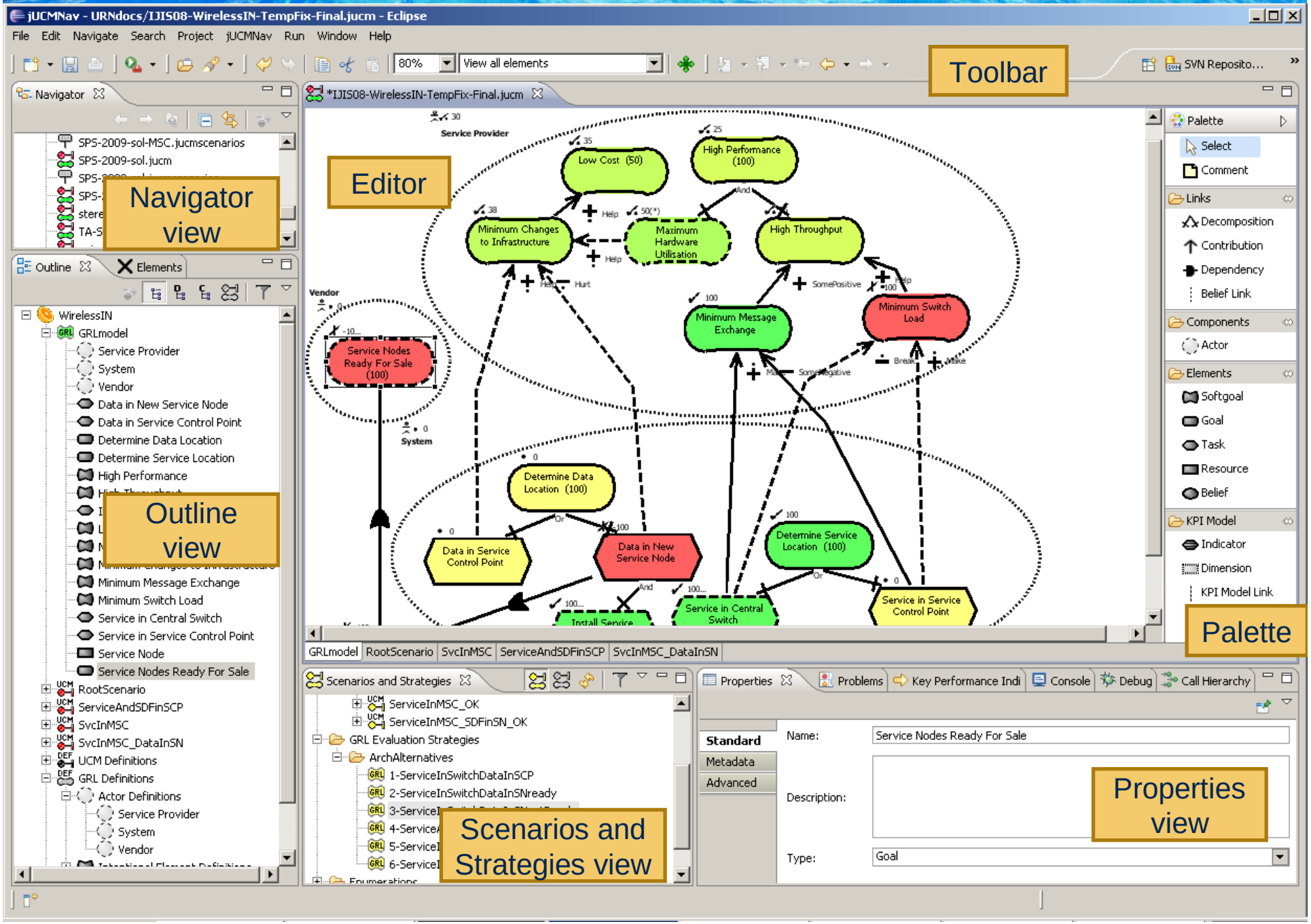
# URN Tool: jUCMNav – Overview (1)

- Free (EPL), open-source plug-in for Eclipse
  - Based on Eclipse Modeling Framework (EMF) and the Graphical Editing Framework (GEF)
- Supports most of GRL and UCM notation elements
- Extensible architecture for customized GRL and UCM analysis techniques
- 4 GRL evaluation algorithms, with color highlight
- 1 UCM path traversal mechanism, with export to flat UCMs and MSCs
  - Integrated MSC viewer

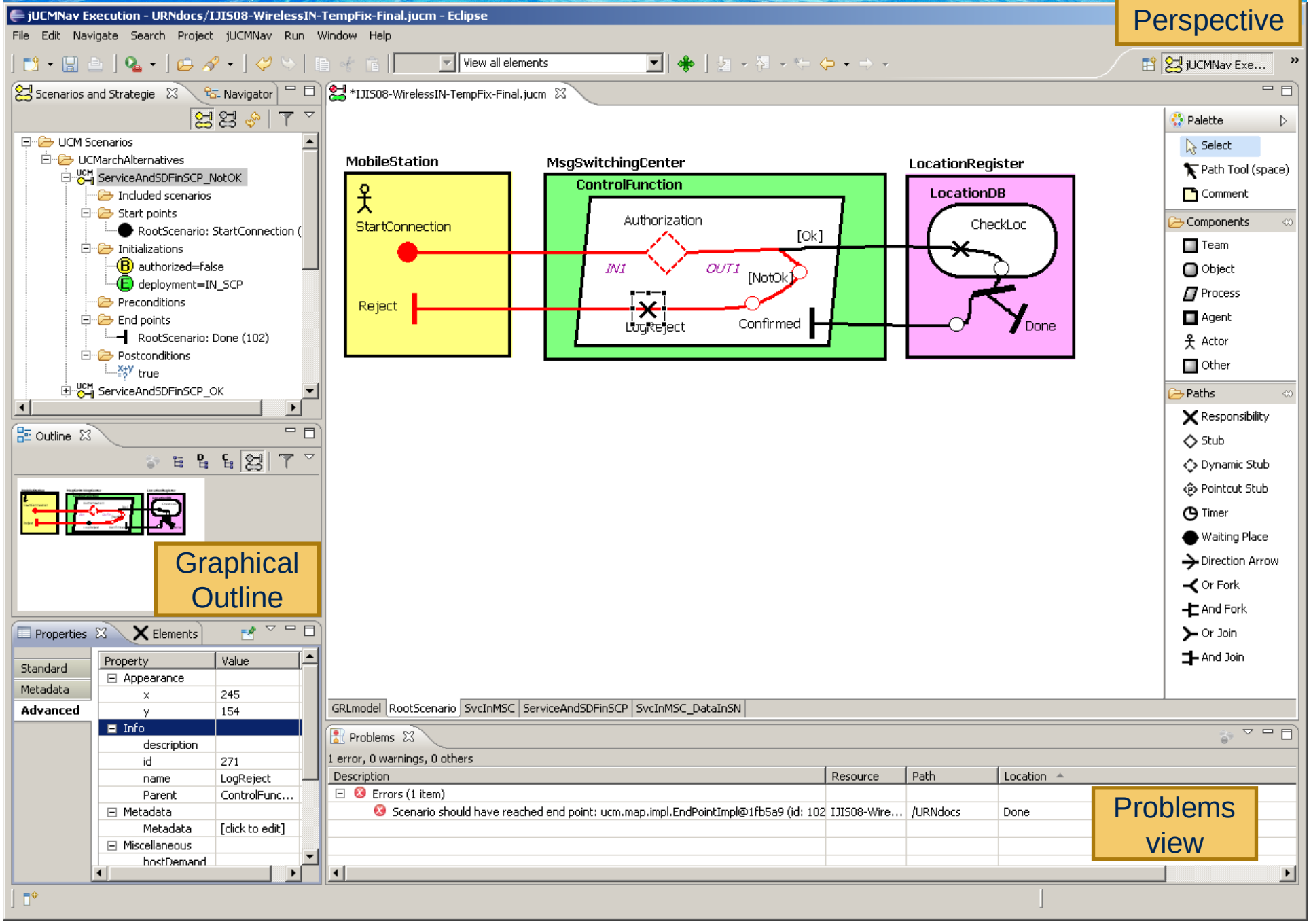
# URN Tool: jUCMNav – Overview (2)

- Other features
  - Export performance models to CSM
  - Integration with DOORS for requirements management
  - Extensions for Business Process Modeling, Monitoring, and Adaptation
  - Extensions for aspect-oriented modeling with AoURN
  - Verification of user-defined (OCL) rules
  - Report generation (PDF, HTML)
  - Export of diagrams to various bitmap formats

# GRL Editor with Strategy Evaluation



# UCM Editor with Scenario Traversal



# Integrated MSC Viewer for Exported UCM Scenarios