



Enhanced Use Case Map Traversal Semantics

<http://jucmnav.softwareengineering.ca>

<http://www.jasonkealey.com/thesis>

Jason Kealey (jkealey@shade.ca)

Daniel Amyot (damyot@site.uottawa.ca)

University of Ottawa, Ottawa, Ontario, Canada

Enhanced Use Case Map Traversal Semantics Presentation Overview

- **Background**
 - Use Case Maps and jUCMNav
- **New Scenario Traversal Semantics**
 - Improved tool support for model analysis
- **New Scenario Export Mechanism**
 - Improved tool support for model transformation
- **Semantic Variations, Workflow Limitations**
 - Lessons learned
- **Conclusions and Future Work**

Enhanced Use Case Map Traversal Semantics Presentation Overview

- **Background**
 - **Use Case Maps and jUCMNav**
- **New Scenario Traversal Semantics**
 - Improved tool support for model analysis
- **New Scenario Export Mechanism**
 - Improved tool support for model transformation
- **Semantic Variations, Workflow Limitations**
 - Lessons learned
- **Conclusions and Future Work**

User Requirements Notation (URN)

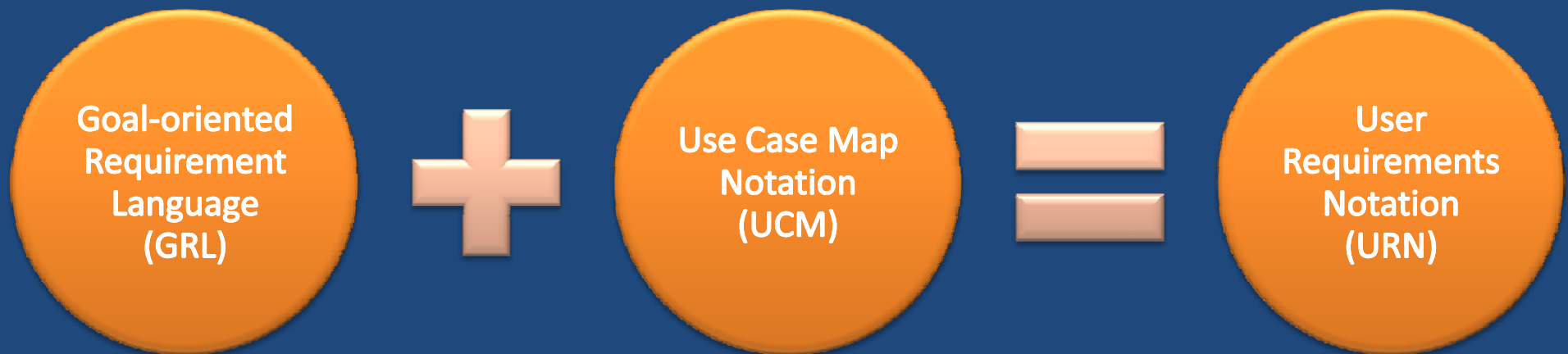
- The **first** standardization effort * which explicitly addresses both:



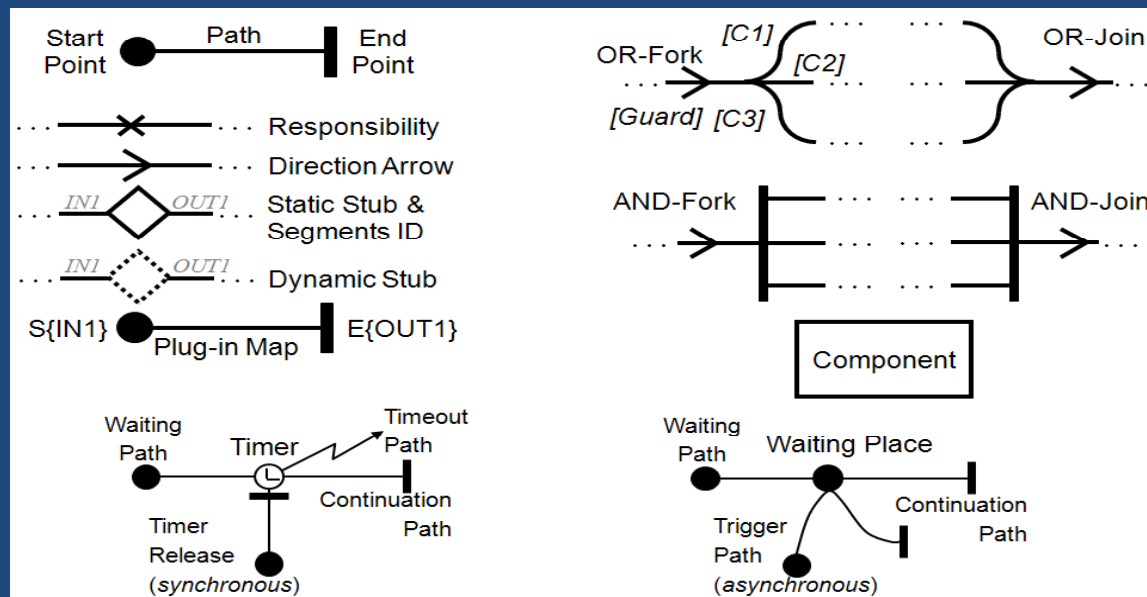
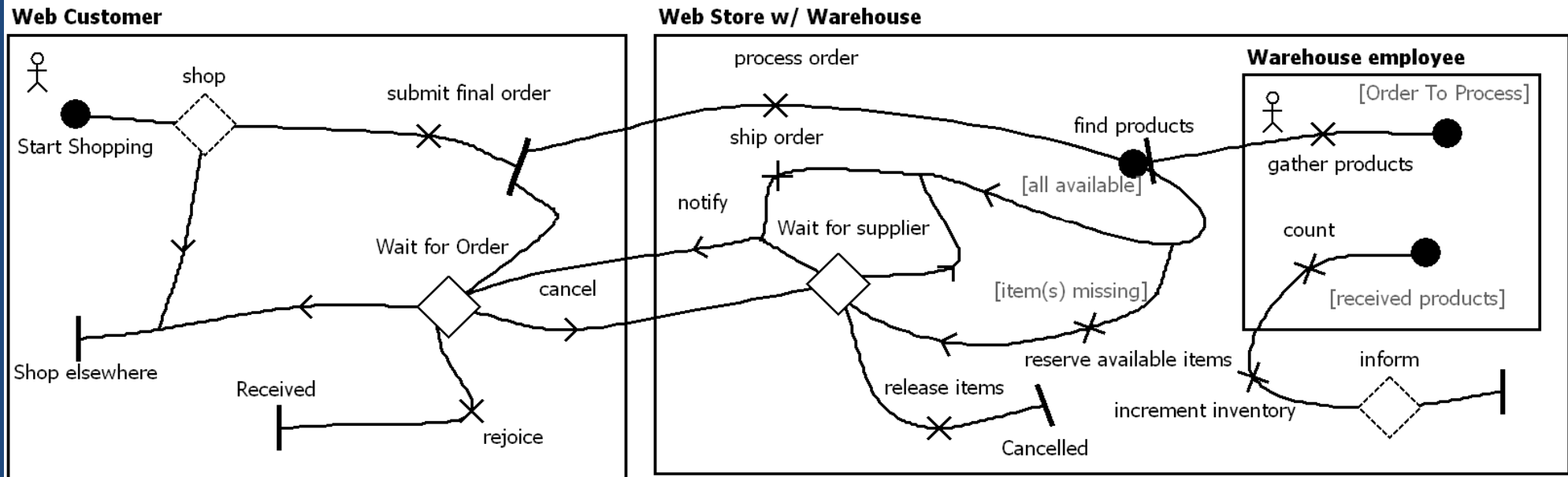
Non-functional requirements with goals



Functional requirements with scenarios



Use Case Map Notation



jUCMNav – URN Editor

The screenshot displays the jUCMNav URN Editor interface. The main workspace shows a sequence diagram with four lifelines: **OriginatingUser** (yellow), **OriginatingAgent** (yellow), **TerminatingAgent** (green), and **TerminatingUser** (green). The diagram includes messages such as **req**, **notify**, **busy**, **ringing**, **ring**, **display**, and **fwd_sig**. Red arrows highlight a specific path of messages. The interface also features an **Outline** on the left, a **Scenarios and Strategies** view at the bottom, and a **Properties** view on the right.

The **Scenarios and Strategies** view shows a tree structure for **BasicCall (5)**, including **BCbusy (2416)** with sub-elements like **Included scenarios**, **Start points**, **Initializations**, **Preconditions**, **End points**, and **Postconditions**. An orange arrow points to the **Initializations** section with the text "Added Scenario Traversal".

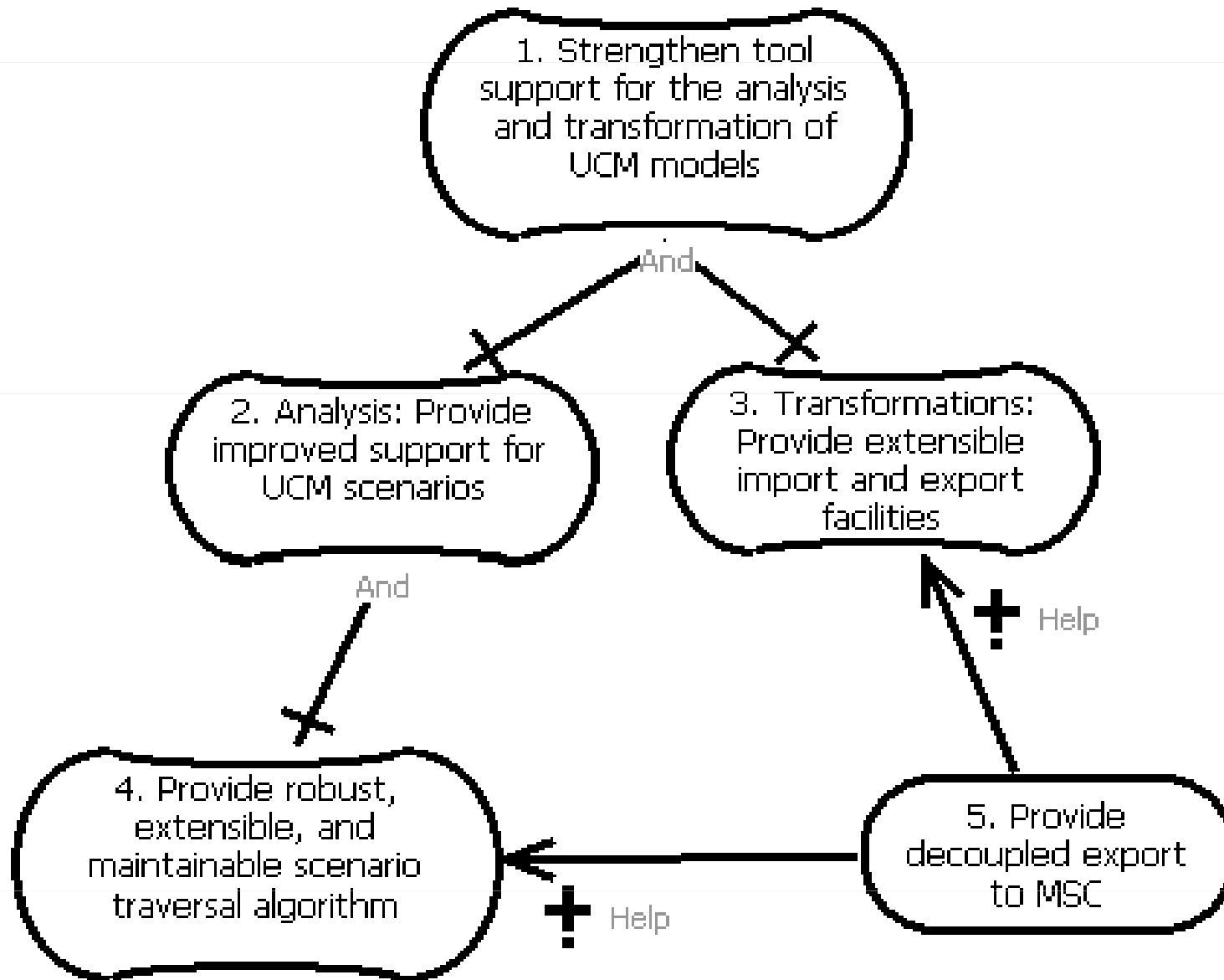
The **Properties** view shows the following table:

Property	Value
Info	
description	Call connection n
id	2
name	SimpleConnection
Metadata	
Metadata	[click to edit]

Scenario Traversal Mechanism

- Provides **operational semantics for UCMs** based on an execution environment.
- Given an initial context (**scenario definition**), the mechanism determines which paths will be followed until no progress is possible.
- Applications:
 - Model understanding and **scenario visualization**
 - **Model analysis**
 - Test goal generation
 - MDA-like transformations

Goals

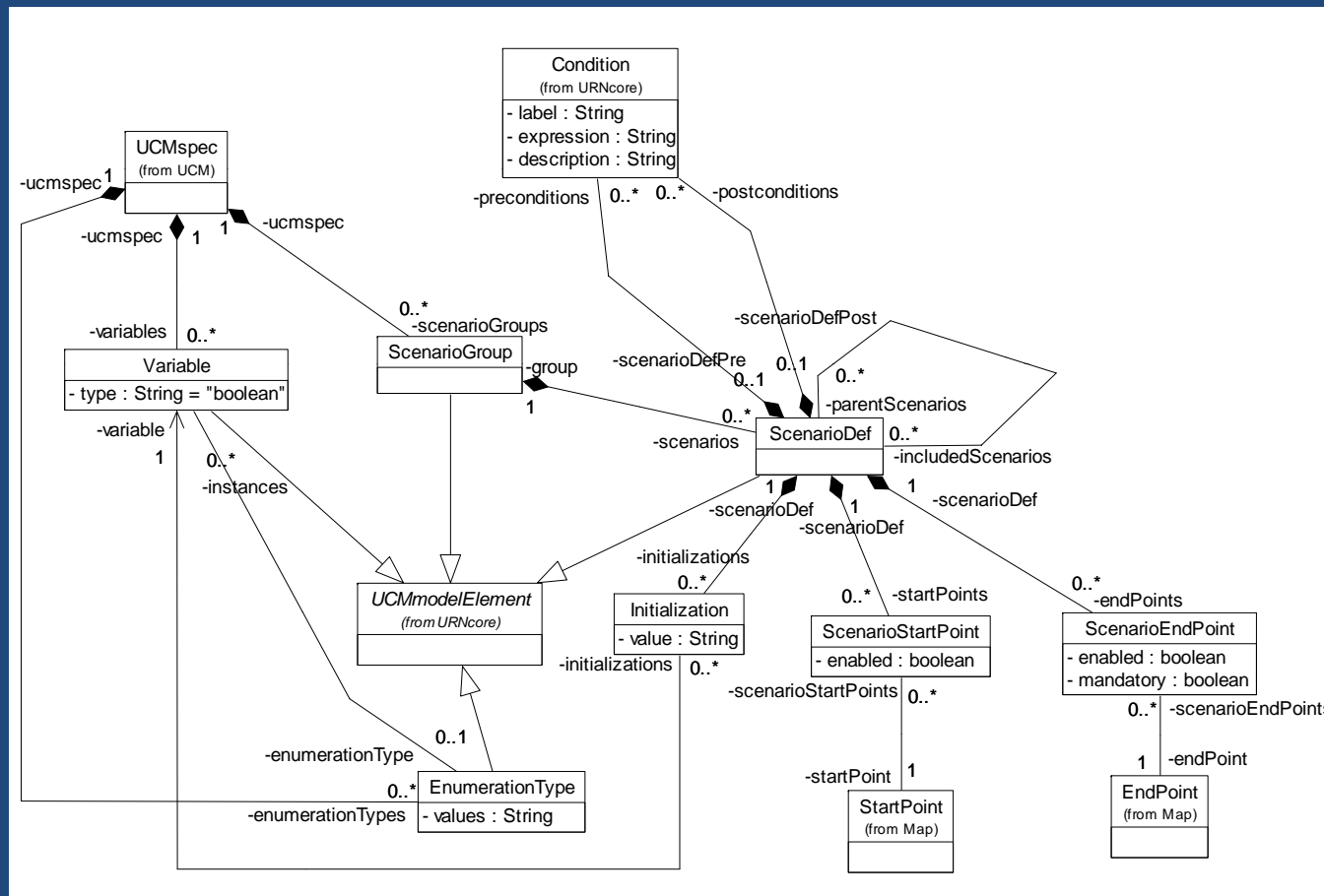


Enhanced Use Case Map Traversal Semantics Presentation Overview

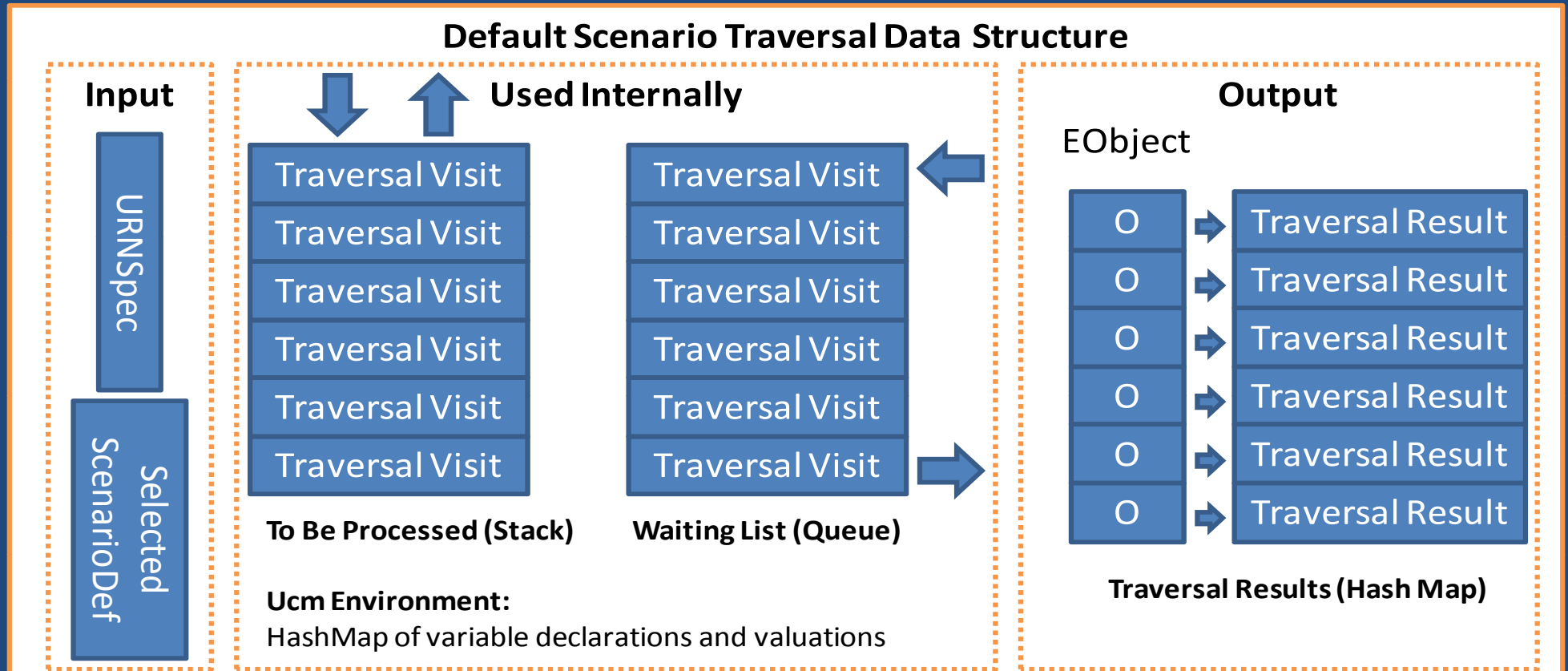
- Background
 - Use Case Maps and jUCMNav
- **New Scenario Traversal Semantics**
 - **Improved tool support for model analysis**
- **New Scenario Export Mechanism**
 - Improved tool support for model transformation
- **Semantic Variations, Workflow Limitations**
- **Conclusions and Future Work**

New Traversal Semantics

- Metamodel enhancements solve many previous limitations (**new data types and scenario re-use**)



Execution Environment Overview



Object Legend

URN Specification

- Use Case Maps
- Variable Definitions
- Enumerations
- [...]

Scenario Definition

- Start/ End Points
- Variable Initializations
- Pre/post conditions
- Included scenarios

Traversal Visit

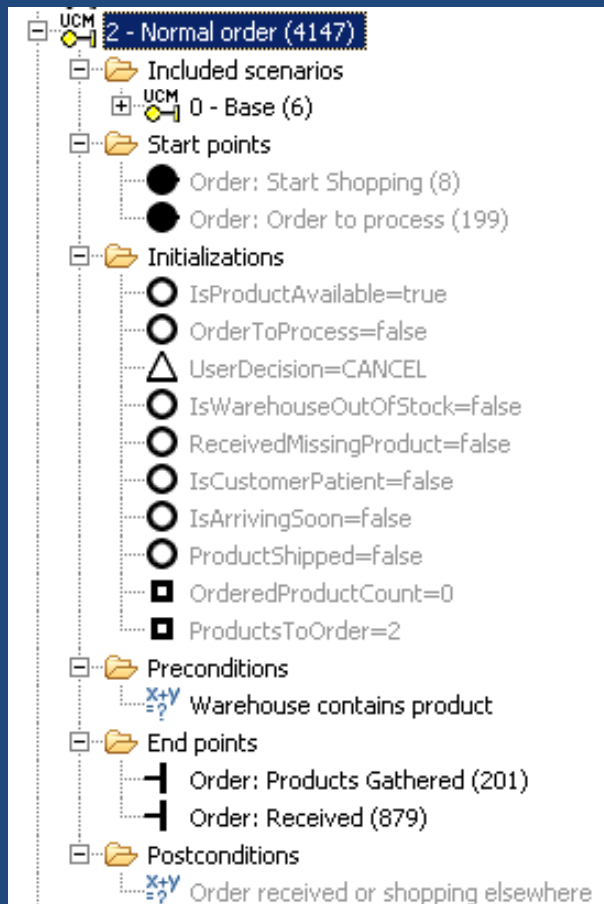
- Path Node To Visit
- Incoming Node Connection
- Traversed Plug-in Bindings
- Thread ID

Traversal Result

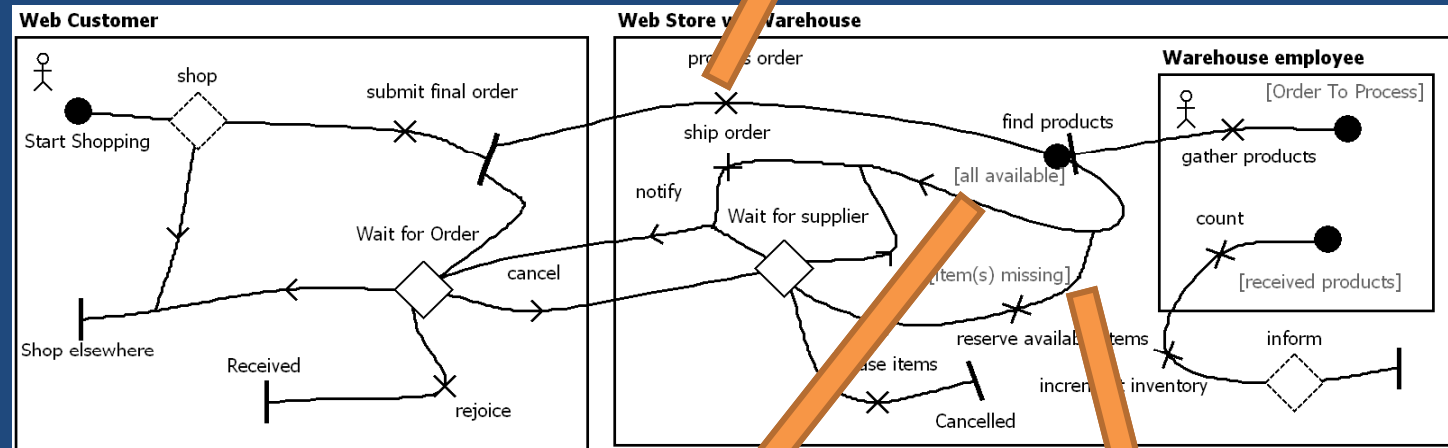
- Internal Hit Count (for infinite loops)
- External Hit Count (for visualization)

(EObject is any EMF object)

Example UCM Model with Scenarios



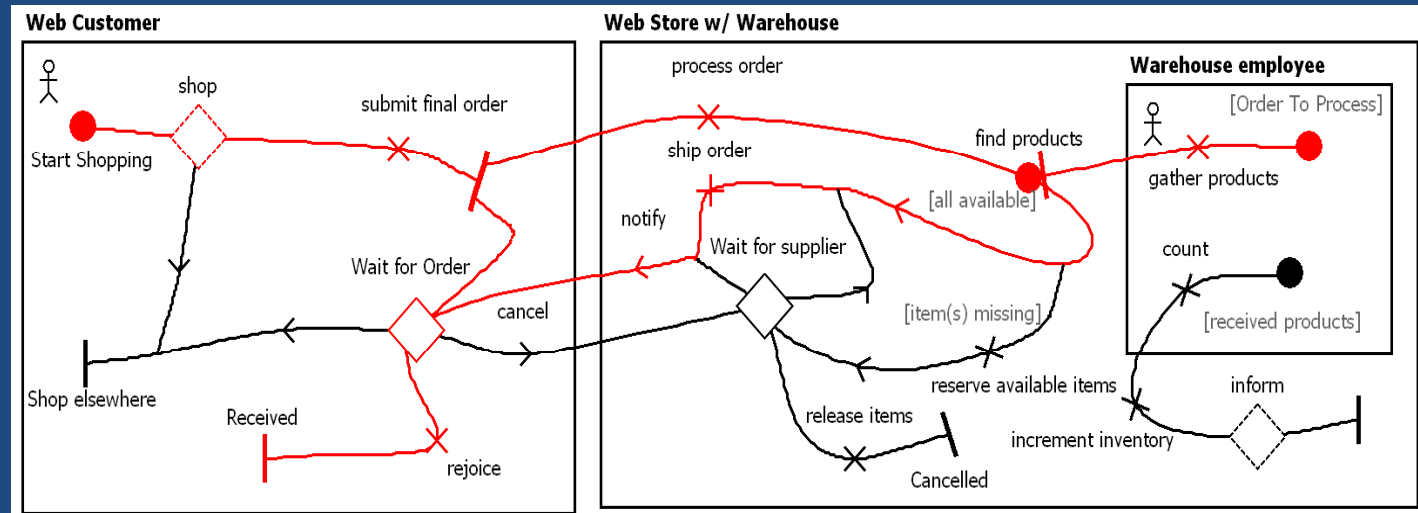
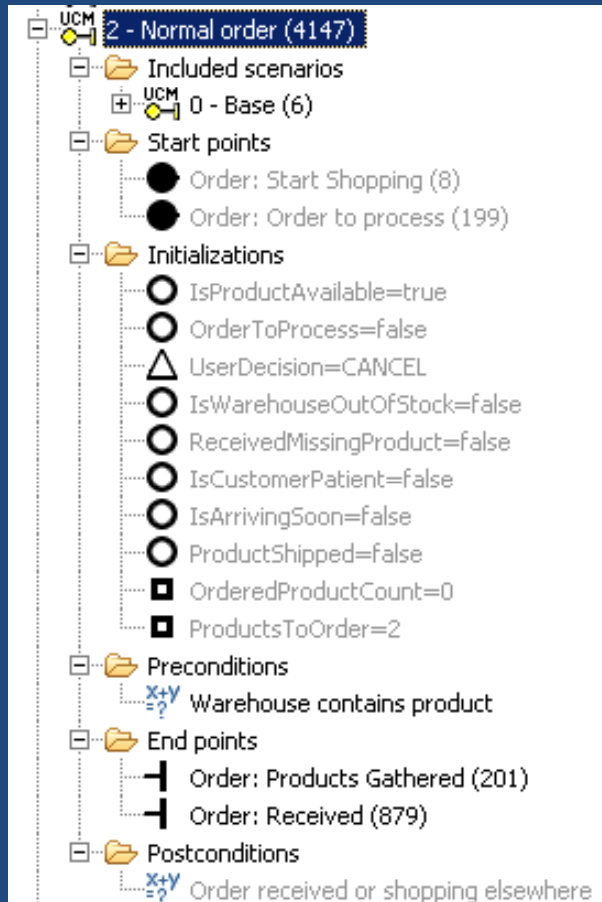
OrderToProcess=true;
ProductShipped=false;



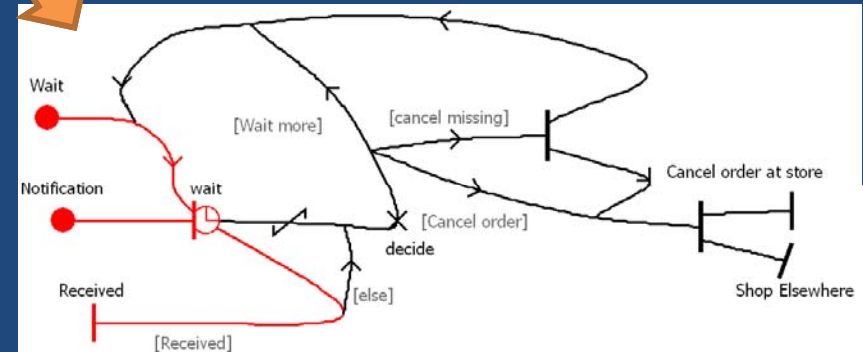
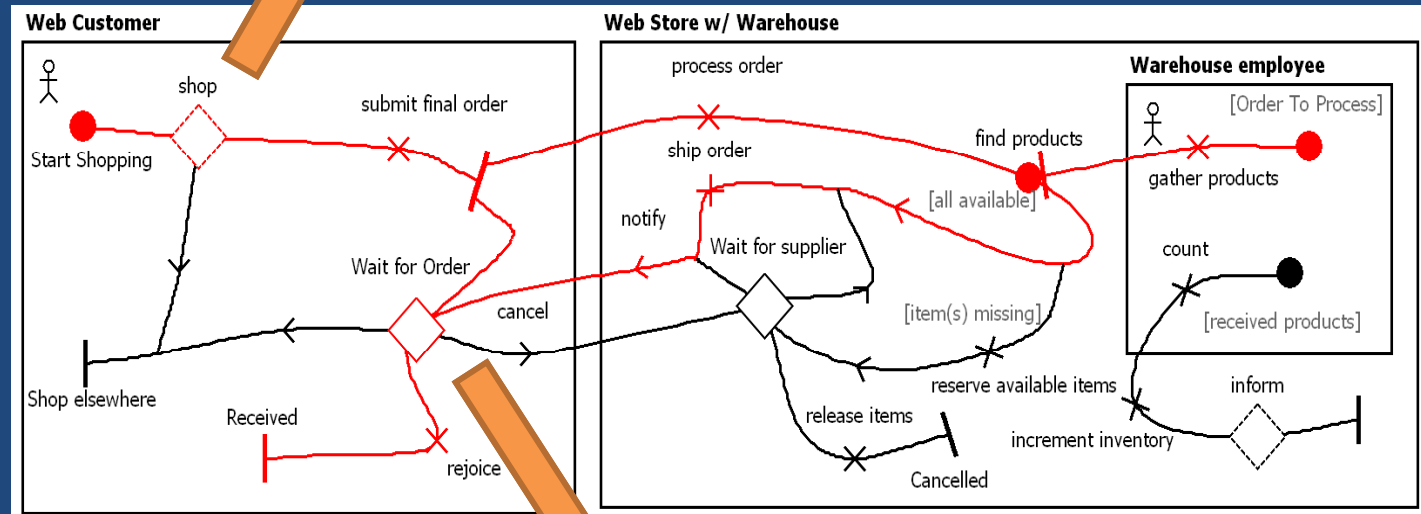
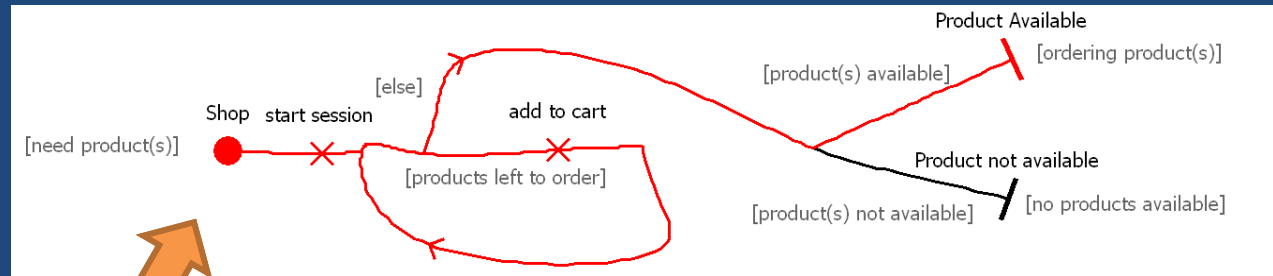
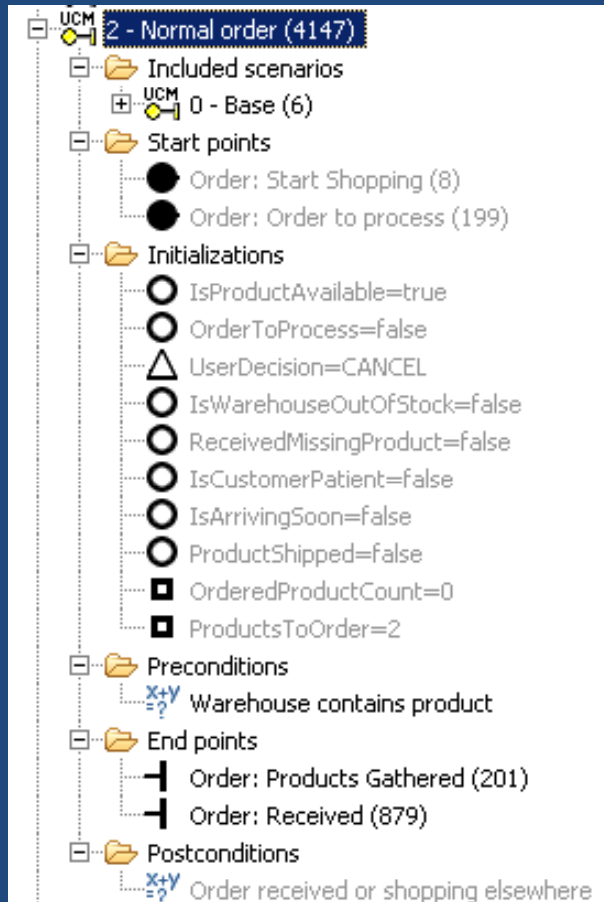
IsProductAvailable

!IsProductAvailable

Example UCM Model with Scenarios



Example UCM Model with Scenarios

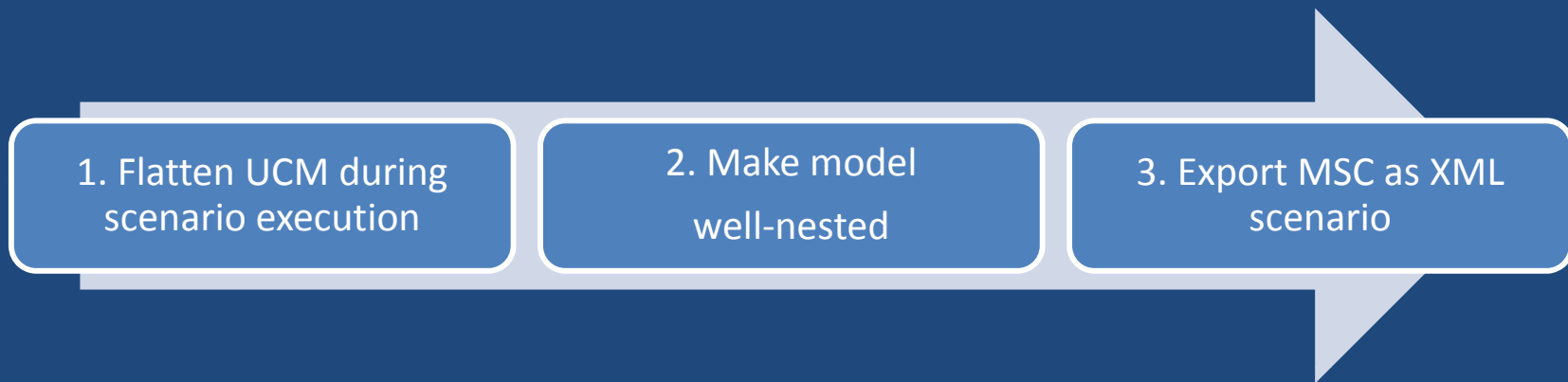


Enhanced Use Case Map Traversal Semantics Presentation Overview

- Background
 - Use Case Maps and jUCMNav
- New Scenario Traversal Semantics
 - Improved tool support for model analysis
- **New Scenario Export Mechanism**
 - **Improved tool support for model transformation**
- Semantic Variations, Workflow Limitations
 - Lessons learned
- Conclusions and Future Work

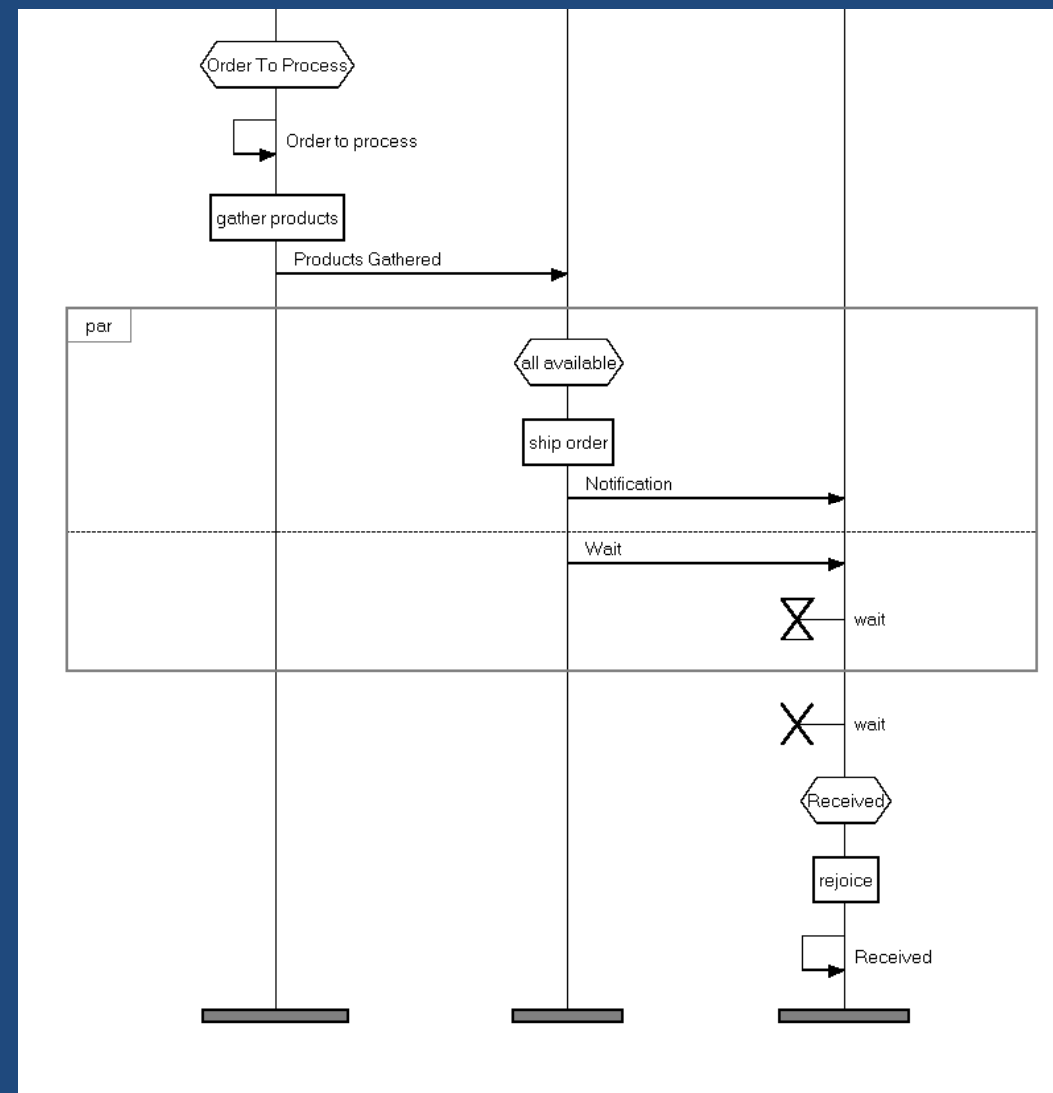
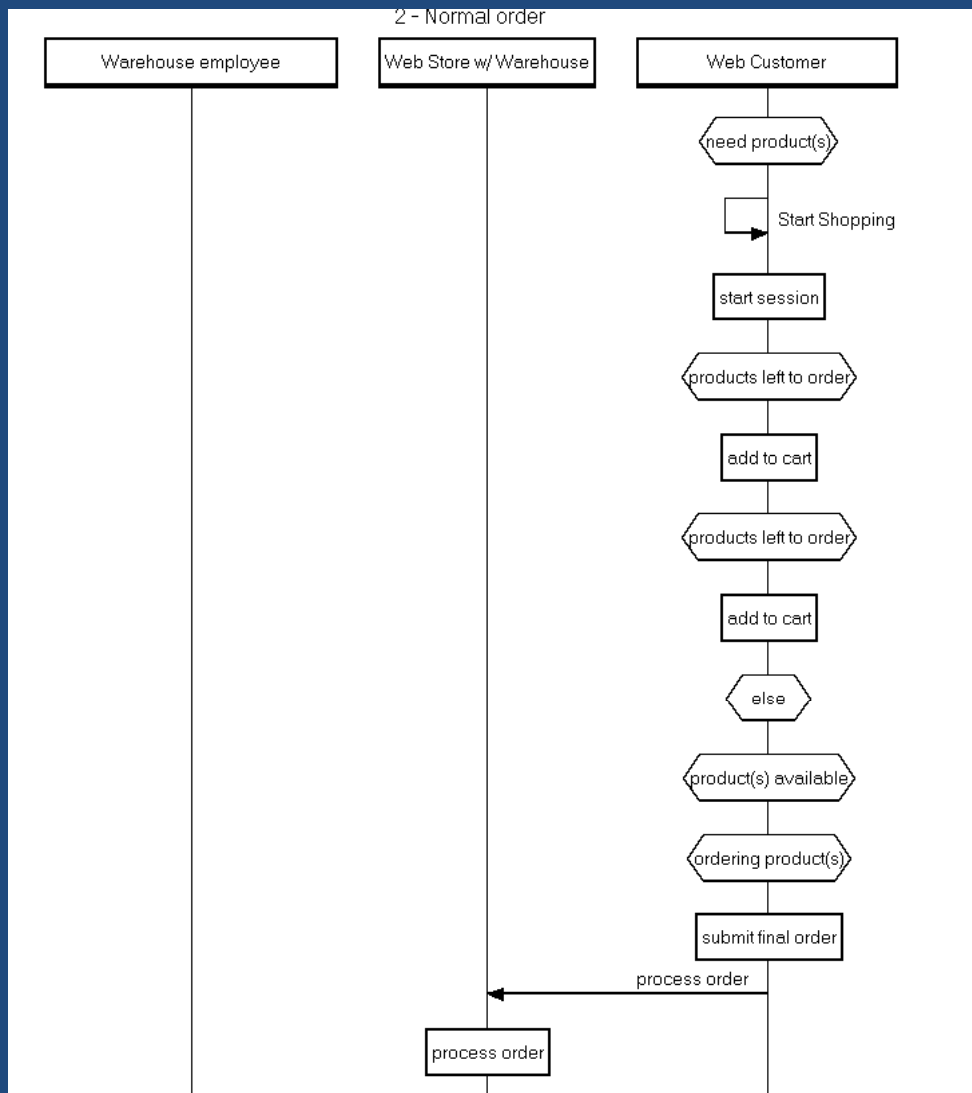
Architecture and Algorithm Highlights

Goal: Generate a MSC representing a UCM scenario execution trace.



Note: The UCM flattening algorithm is implemented as a listener to the scenario traversal algorithm, to avoid repeating UCMNav's maintainability issues.

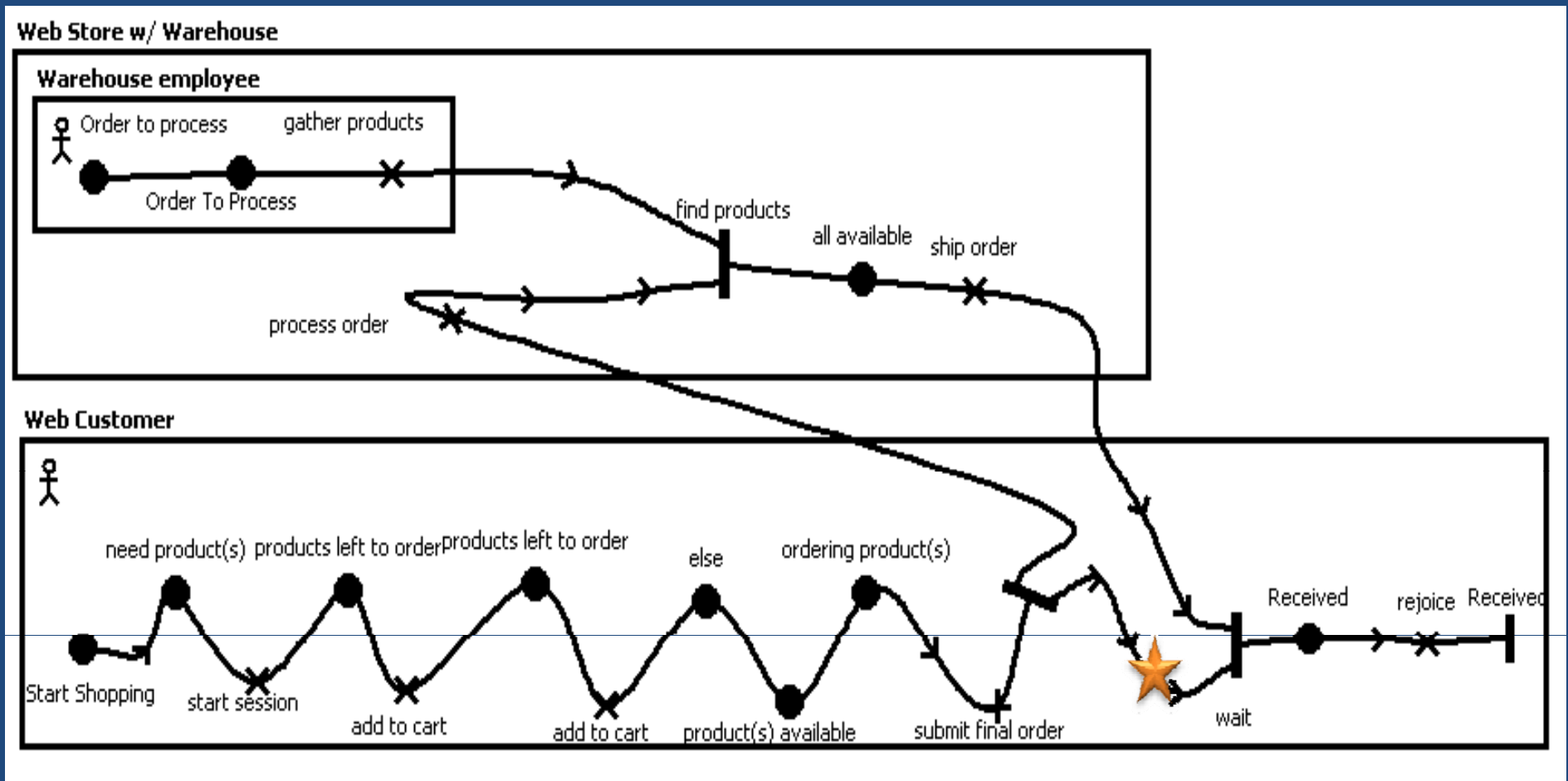
Message Sequence Charts (MSCs)



(MSC generated from the previous example viewed in MSC Viewer)

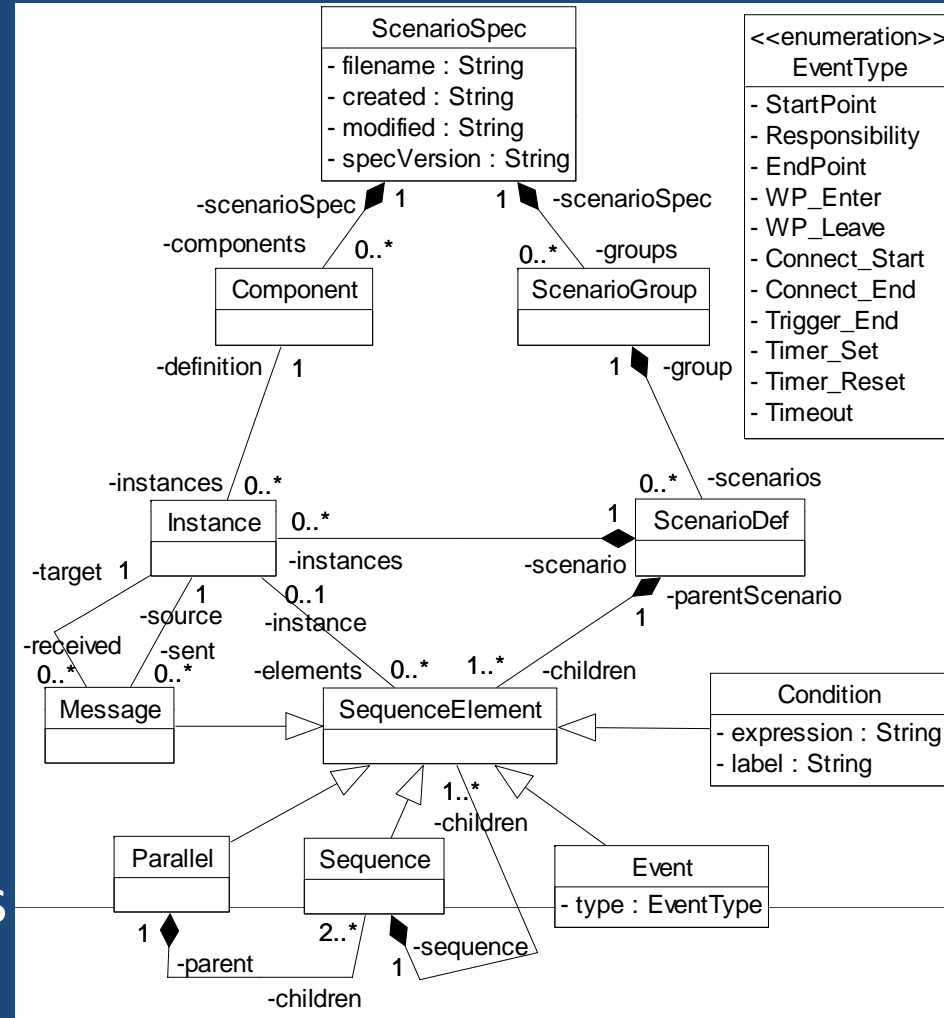
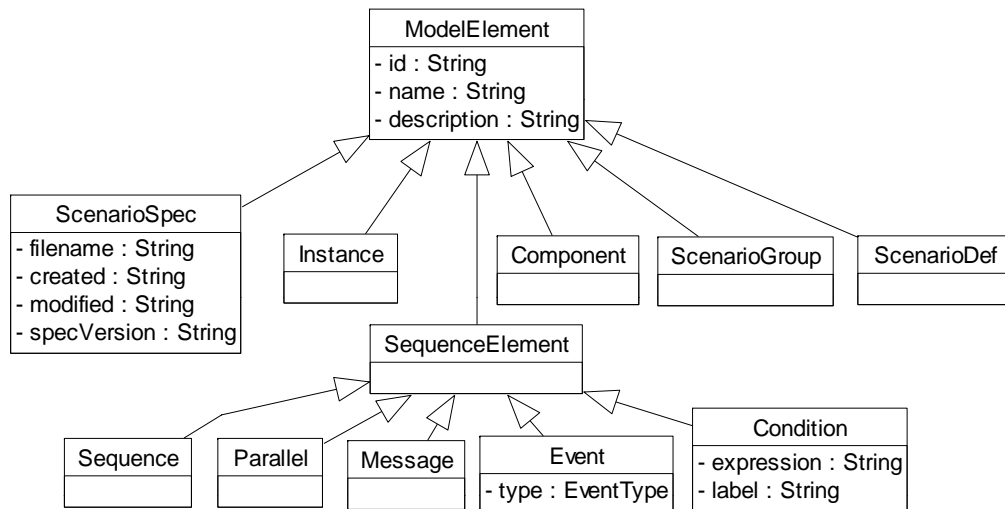
1. UCM Flattening

Listen to events and build an equivalent, **flat UCM**.



Even used alone, this feature is a great debugging/analysis tool.

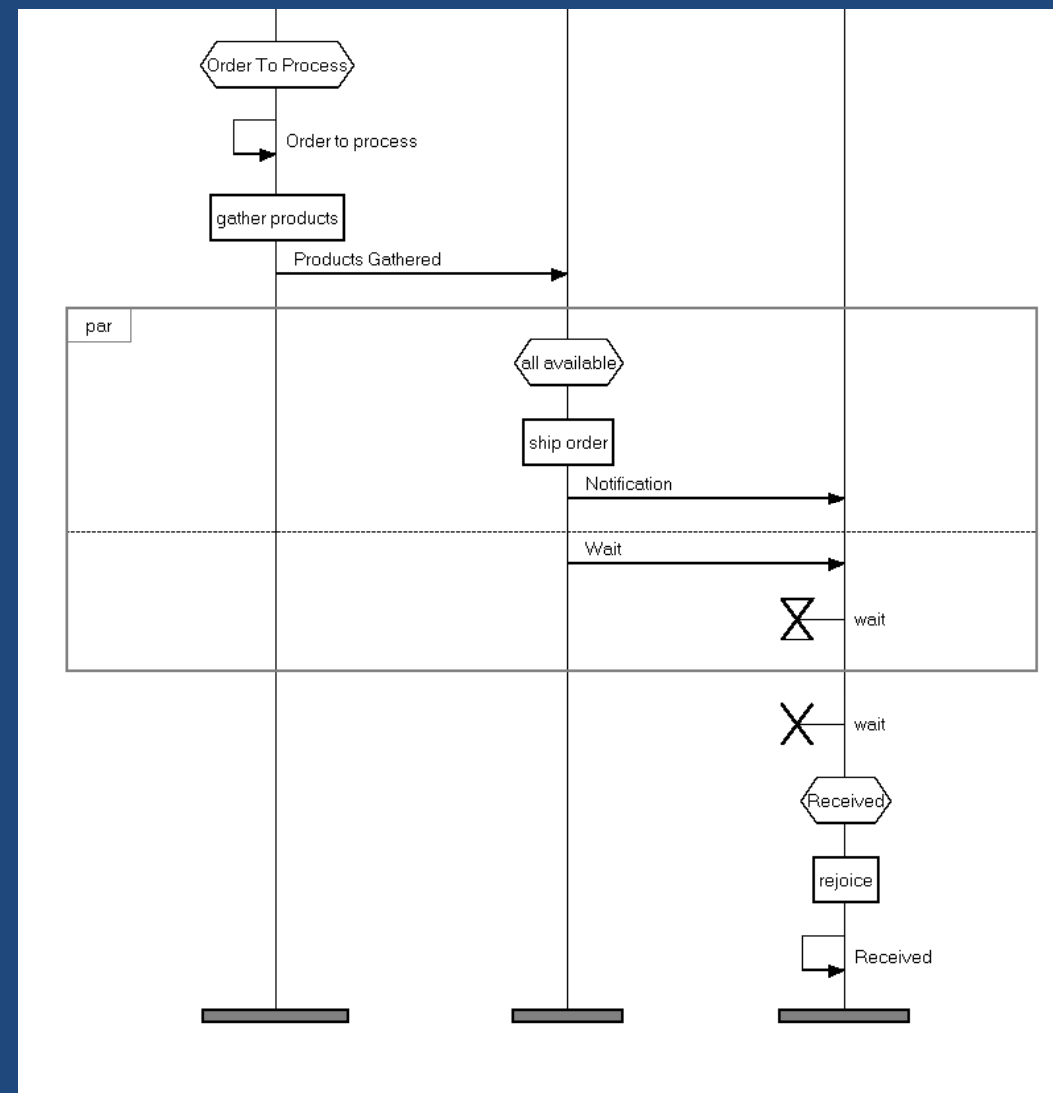
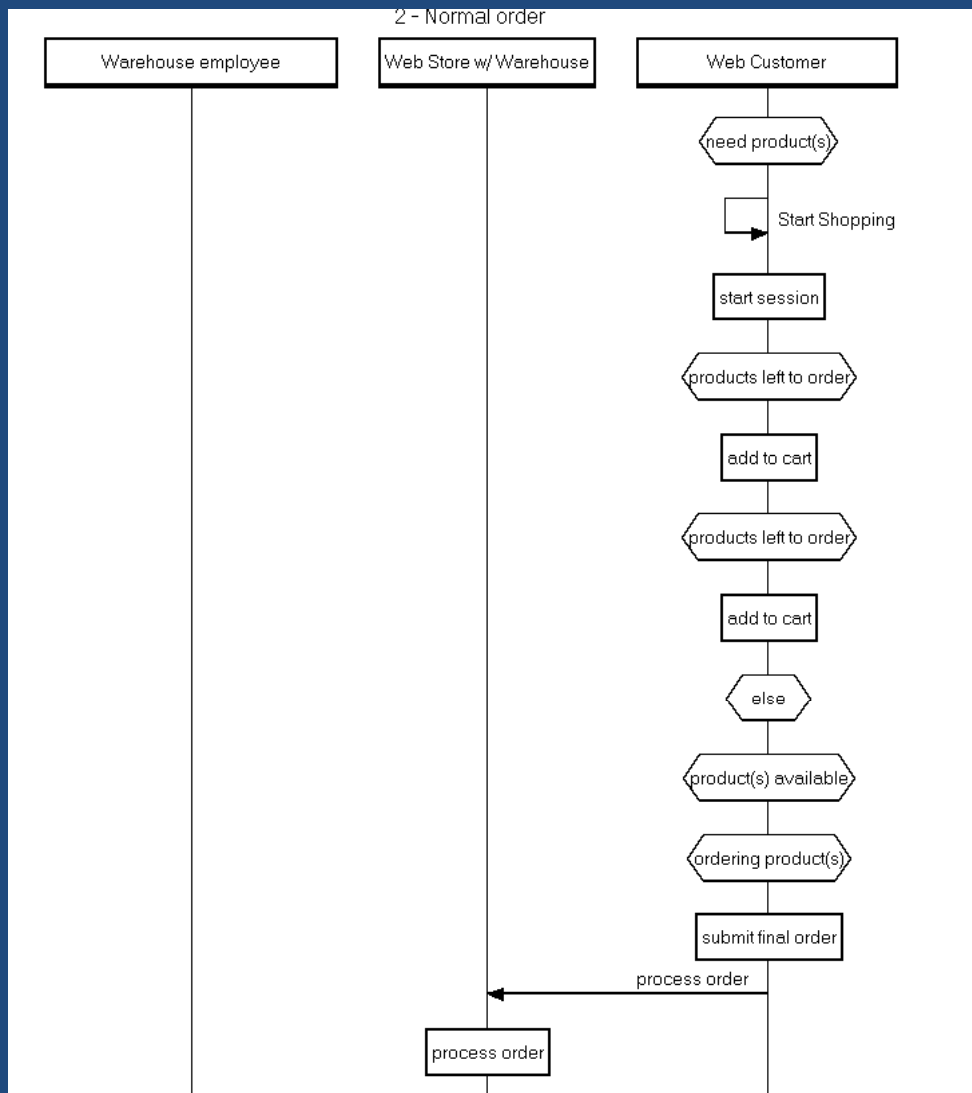
3. Message Synthesis and XML Export



- Simple representation of MSCs
 - **New EMF-based input layer**
- Simple message synthesis algorithms

- The MSC Viewer can read files generated by UCMNav as well as textual **MSCs described using the Z.120 format**

Message Sequence Charts (MSCs)



(MSC generated from the previous example viewed in MSC Viewer)

Enhanced Use Case Map Traversal Semantics Presentation Overview

- Background
 - Use Case Maps and jUCMNav
- New Scenario Traversal Semantics
 - Improved tool support for model analysis
- New Scenario Export Mechanism
 - Improved tool support for model transformation
- Semantic Variations, Workflow Limitations
 - Lessons learned
- Conclusions and Future Work

Semantic Variations

Table 1 Identified semantic variations

#	Element	Semantic Variation	Main Alternatives
1	Start Point	Precondition is false	Abort, Pause , Continue
2	Or-Fork Dynamic Stub	Multiple branches evaluate to true	Abort, Pick one (deterministic or not) , Follow all
3	Or-Fork Dynamic Stub	No branches evaluate to true	Abort, Pause , follow random
4	And-Join	Not all in-branches arrived	Abort , Continue
5	And-Join	Do they have memory?	Yes, No, Hybrid
6	Multiple	Can block simultaneous instances of a node?	Yes, No
7	Timer	Both continuation and timeout path are enabled	Pick continuation , Pick timeout, Or-Fork Behaviour
8	Timer / Wait	Do they have memory?	Yes, No
9	Timer / Wait	Connected path arrives at unblocked node	Error , Pause, Continue
10	Stub	No plug-in exists	Error, Continue
11	End Point	Multiple different out bindings to fire	All in parallel , All in sequence, Pick one
12	End Point	Same out binding should be fired multiple times	All in parallel, All in sequence, Once
13	End Point	Post-condition is false	Abort , Pause, Continue
14	Scenario	How are start points launched?	In parallel, In sequence , Mixed

Workflow Pattern Limitations

Table 3 Relationships between workflow patterns and semantic variations

Workflow Pattern	Solution	Related to
Multiple choice	Or-Fork with concurrent branches, And-Fork with conditions on branches Dynamic Stub with concurrent plug-ins	SV: 2
Synchronizing merge	Synchronizing Dynamic Stub with concurrent plug-ins	SV: 6, 12
N-out-of-M join	Synchronizing (n/m) Dynamic Stub with concurrent plug-ins	SV: 4, 6
Discriminator	Special case (1/m) of above	SV: 4, 6
Multiple Instances without synchronization	Use of replication factor on component or plug-in binding	SV: 2, 6, 11, 12
Multiple Instances with a priori design time knowledge	Synchronizing static stub with replication factor as fixed number	SV: 2, 6, 11, 12
Multiple Instances with a priori runtime knowledge	Synchronizing static stub with replication factor as variable expression	SV: 2, 6, 11, 12
Interleaved parallel routing	Component bindings and specialized traversal	SV: 14
Deferred choice	Specialized traversal	SV: 2, 3
Milestone	Specialized traversal	SV: 2, 3

Enhanced Use Case Map Traversal Semantics Presentation Overview

- Background
 - Use Case Maps and jUCMNav
- New Scenario Traversal Semantics
 - Improved tool support for model analysis
- New Scenario Export Mechanism
 - Improved tool support for model transformation
- Semantic Variations, Workflow Limitations
- **Conclusions and Future Work**

Goals Attained, Problems Solved

- Improved tool support for UCM analysis and transformation
- The **original UCMNav implementation** of this mechanism suffers from **many limitations**.
 - Non-standard syntax
 - The only data type allowed is Boolean
 - **Scenario definitions are not reusable**
 - The same start points cannot be triggered multiple times
 - Rigid traversal, no semantic variations, intolerant to errors
 - Only one scenario can be run at a time
 - **Dependency on external tools** for MSC generation (e.g., UCMExporter) and visualization (e.g., Telelogic Tau), hence hindering usability
 - **Difficult to debug and maintain**
 - strong coupling with MSC Export and trace linearization

Contributions

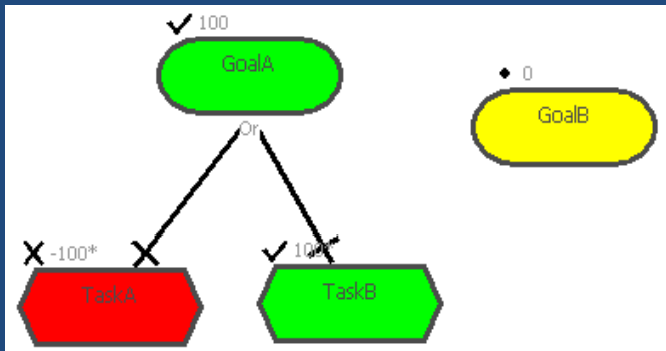
- **Greatly improved UCM analysis tooling**
 - New functionality, greater expressiveness
 - More reliable, maintainable, and usable.
- **De-coupled MSC export functionality**
 - Integrated with MSC Viewer
 - More reliable, maintainable, and usable.
- **Identified semantic variations and relationships to workflow limitations**
- **Additional related work presented in thesis**
 - <http://www.jasonkealey.com/thesis>

Future Work

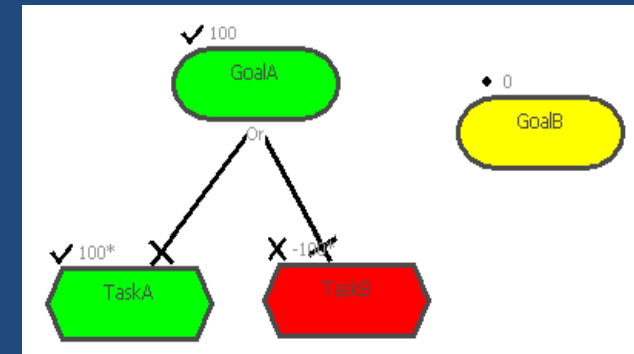
- Tool support for variations / parameterization
- Syntactical and semantic enhancements related to the workflow notation
- Aspect-oriented URN
- Reinforce URN's integrated support of goals (via GRL) and scenarios (via UCM) in one model

GRL Strategy & UCM Scenario Integration

GRL Strategy 1



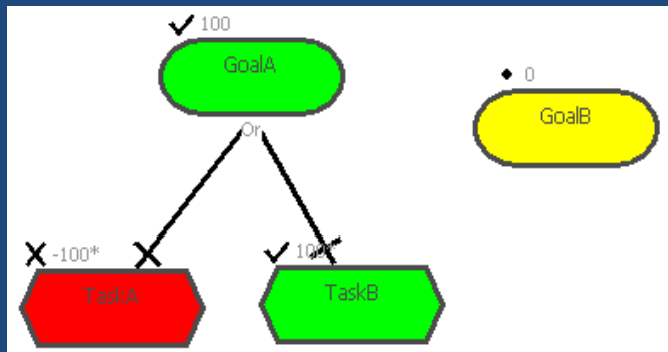
GRL Strategy 2



- The Goal-Oriented Requirement Language has **GRL Strategies** which are akin to UCM scenarios
 - **Initial goal satisfaction** is defined and **propagated** to the other goals in the graph.
 - Visual indications using color and satisfaction levels

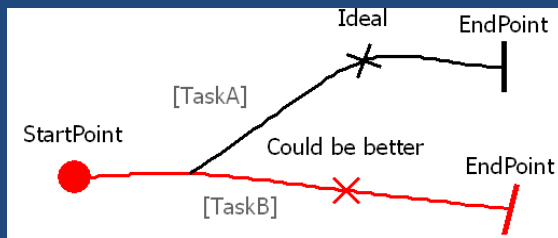
GRL Strategy & UCM Scenario Integration

GRL Strategy 1



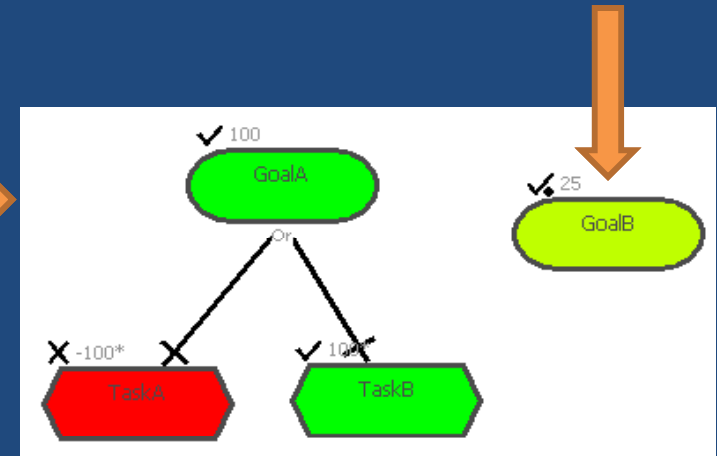
Satisfaction = 0

1) GRL Strategy influences UCM scenario traversal



Satisfaction = 25

2) UCM scenario traversal influences GRL goal satisfaction (see Goal B)



Acknowledgements

- This research was supported by the **Natural Sciences and Engineering Research Council of Canada**, through its programs of Discovery Grants and Postgraduate Scholarships.
- We are grateful to **J.-F. Roy, E. Tremblay, J.-P. Daigle, J. McManus, and G. Mussbacher** for various contributions to the jUCMNav tool, and to **Alex Boyko et al.** for their MSC Viewer plug-in.



Enhanced Use Case Map Traversal Semantics

<http://jucmnav.softwareengineering.ca>

<http://www.jasonkealey.com/thesis>

Jason Kealey (jkealey@shade.ca)

Daniel Amyot (damyot@site.uottawa.ca)

University of Ottawa, Ottawa, Ontario, Canada