UML Profile for Goal- oriented Modelling

Muhammad Rizwan Abid
mabid006@uottawa.ca

Supervising Professors:
Daniel Amyot
Stéphane Sotèg Somé

Université d’Ottawa | University of Ottawa

www.uOttawa.ca
Problem

• Can UML be profiled to support goal-oriented modelling with a semantics rooted in a standard metamodel such as that of the Goal-oriented Requirement Language (GRL)?

• Can such profile be implemented with Telelogic Tau G2?
Contributions

• The creation of a UML profile for GRL, where UML metaclasses are mapped in detail to GRL’s metaclasses. Standard guidelines have been followed while defining this profile.
• A proof of concept implementation, which demonstrates the feasibility of supporting such profile in Telelogic Tau G2 4.0
• Illustration of typical usage of this profile with examples where GRL is used standalone in a model, and then where GRL diagrams are combined with selected UML diagrams in a model.
Presentation Overview

- Motivation and background work
- GRL and current tool support (jUCMNav)
- UML profiles
- UML profile for GRL
- GRL profile in Tau: Stereotype Mechanism
- GRL profile in Tau: Metamodel Extension Mechanism
- Example
- Evaluation and Conclusions
Motivation

• UML (Unified Modeling Language) does not address explicitly the modelling of goals and non-functional requirements.

• Some work in the area of UML profiles for goal modelling exists, but the solutions proposed suffer from many deficiencies, including non-compliance to minimal requirements such as:
  – **R1**: Integration with UML
  – **R2**: Diagram pollution avoidance
  – **R3**: Metamodel stability
  – **R4**: Implementability of the profiling mechanism
Related Work

- Dallons et al.: A Template-based Analysis of GRL
  - Does not satisfy any of our requirements
- Grangel et al.: UML Profile for Enterprise Goal Modelling
  - Satisfies R1 and partially satisfies R4
- Supakkul and Chung: UML Profile for Softgoal by Use Case Driven Approach
  - Satisfies R2, partially satisfies R3, and R4 is not applicable
- Cysneiros et al.: Using UML to Reflect Non-Functional Requirements
  - Does not satisfies R1 and the remaining requirements are not applicable
Goal-oriented Requirement Language

- Part of ITU-T’s User Requirements Notation (URN - Z.151)
- Subset of $i^*$ syntax + NFR Framework evaluations
- Goal-oriented Requirement Language used to model
  - Goals and stakeholders
  - Requirements
  - Alternatives
  - Rationales
- There are 3 main categories of concepts:
  - Actors
  - Intentional elements
  - Links
GRL Notation Elements

(a) GRL Elements
- Softgoal
- Belief
- Goal
- Task
- Actor
- Resource
- Actor Boundary

(b) GRL Satisfaction Levels
- Satisfied
- Weakly Satisfied
- Undecided
- Weakly Denied
- Denied
- Conflict

(c) Link Composition
- AND
- XOR
- IOR

(d) GRL Links
- Contribution
- Correlation
- Dependency
- Decomposition

(e) GRL Contributions Types
- Break
- Hurt
- Some-
- Unknown
- Make
- Help
- Some+
- Equal

UML Profile for Goal-oriented Modelling
jUCMNav

- An Eclipse-based GRL editor:
  - Supports the User Requirements Notation (GRL + UCM)
  - Allows users to create and maintain GRL models
  - Supports the analysis of GRL models
  - Supports the creation of links and annotations
  - Exports to various formats
  - Open source (EPL)
UML Profile

- Generic extension mechanism for tailoring UML to a particular domain.
- Profiles can be used to adapt/extend the UML metamodel, but they do not allow the existing UML metamodel to be directly changed.
A UML Profile Can…

- Identify a subset of the UML metamodel.
- Specify new “standard elements”.
- Specify “well-formedness rules” beyond those specified by the identified subset of the UML metamodel.
- Specifies semantics, expressed in natural language, beyond what is specified by the identified subset of the UML metamodel.
Profiling Mechanisms

Profiles can be created in one of two ways:

1) Stereotype Mechanism (SM)

2) Metamodel Extension Mechanism (MEM)
Stereotype Mechanism

• Very straightforward way of customizing UML.
• Extension of basic UML elements.
• Contains simple customizations (custom name, custom attribute, custom appearance).
  – For instance, GRL task element can be represented as a stereotype of UML class.
• Limitations
  – GRL elements that are just stereotypes of existing UML elements can only be used in regular UML diagrams.
  – Non-GRL elements can be included in GRL diagrams.
  – Lead to unfriendly editors and confusing models.
Metamodel Extension Mechanism

- More robust extension mechanism.
- Has all functionalities of stereotype mechanism.
- Also has the ability to customize non-basic UML elements such as *diagrams*.
  - E.g., GRL diagram can be represented as a metaclass extension of UML class diagram and then restrictions on the GRL diagram can be added to allow only GRL elements.
- More flexible but more complex way of profiling.
Stereotype Summary for GRL

- Mapping of each GRL metaclass from a UML metaclass
- Narrows the scope of semantics
- Addition of new attributes
- Addition of new constraints without violating inherited constraints
<table>
<thead>
<tr>
<th>Stereotype</th>
<th>Stereotyped metaclass</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRLspec</td>
<td>Model</td>
</tr>
<tr>
<td>GRLmodelElement</td>
<td>NamedElement</td>
</tr>
<tr>
<td>GRLLinkableElement</td>
<td>Class</td>
</tr>
<tr>
<td>Actor</td>
<td>Class</td>
</tr>
<tr>
<td>IntentionalElement</td>
<td>Class</td>
</tr>
<tr>
<td>IntentionalElementType</td>
<td>Enumeration</td>
</tr>
<tr>
<td>ImportanceType</td>
<td>Enumeration</td>
</tr>
<tr>
<td>ElementLink</td>
<td>Relationship</td>
</tr>
<tr>
<td>Contribution</td>
<td>Association</td>
</tr>
<tr>
<td>ContributionType</td>
<td>Enumeration</td>
</tr>
<tr>
<td>Dependency</td>
<td>Association</td>
</tr>
<tr>
<td>Decomposition</td>
<td>Association</td>
</tr>
<tr>
<td>DecompositionType</td>
<td>Enumeration</td>
</tr>
</tbody>
</table>
Overview of Profile

GRL Model Element

```
<<metaclass>>
NamedElement

<<stereotype>>
ElementLink

extends

<<stereotype>>
GRLModelElement

<<stereotype>>
GRLLinkableElement
```
Element Link
GRL Spec

UML Profile for Goal-oriented Modelling
Enumerations

- **DecompositionType**
  - AND
  - XOR
  - IOR

- **ContributionType**
  - Make
  - Help
  - SomePositive
  - Unknown
  - SomeNegative
  - Hurt
  - Break

- **IntentionalElementType**
  - Softgoal
  - Goal
  - Task
  - Resource
  - Belief

- **ImportanceType**
  - High
  - Medium
  - Low
  - None
Tool Support

Telelogic Tau G2 4.0

- Supports Model Driven Development (MDD) in a UML-based environment
- Supports both Stereotype Mechanism and Metamodel Extension Mechanism for UML profile creation
- Available for all well-known operating systems including Microsoft Windows, Sun Solaris, Redhat Enterprise Linux and Citrix XPe
- Allows to extend the UML metamodel
GRL Profile in TAU

Stereotype Mechanism
GRL Profile in TAU

Metamodel Extension Mechanism
Four Sub-Packages

- **GRL Model**
  - Contains all metaclasses used for GRL model creation.

- **GRL Editor**
  - Contains all of the information necessary to create a GRL editor, to specify which information can be kept by the editor, as well as to whom this information can be passed to.

- **GRL Concrete Elements**
  - Shows the metaclasses created for GRL elements. These metaclasses describe the actual GRL constructs.

- **GRL Abstract Elements**
  - Contains all of the stereotypes that represent the GRL profile elements.
Example of Profile Usage

• Merchant and Customer Dependencies
  – Covers almost all the possible usage scenarios of GRL constructs
  – Includes GRL goal, softgoal, task, belief, resource, dependency, contribution, correlation, decomposition and actor
  – Includes four actors: Customer, Clerk, Bank and Merchant
Example in jUCMNav

UML Profile for Goal-oriented Modelling
Example in Tau G2 – GRL Profile
Tau Support for Profiling (1/2)

- Provides predefined stereotypes to obtain advanced profile functionalities.

- Most work is wizard base and user friendly.

- Many options are in property view
  » makes the environment easier to use.

- Most of the UML metamodel classes are supported in our scope.

- Supports the association of customized icon with Enumeration literals which enable us to assign customized icons.
Tau Support for Profiling (2/2)

• Allows to create a specific GRL editor with a customized tool palette.

• Provides start link and end link feature that allows to navigate from one diagram construct to the other.

• Technical support is very quick and beneficial.

• Provides the re-usability of the constructs. This feature allows a UML diagram to re-use a reference to a GRL construct from another diagram. This means that any change in the re-used construct (e.g., the name or some other attribute) appears automatically on all its other occurrences in all the diagrams of the model.
Tools Limitations (1/3)

- No construct or mechanism by which an Actor boundary can be created.

- Tool does not support all of the UML metamodel classes eg Enumeration metaclass and the NamedElement metaclass.

- Lacking support for the association of customized multiple icons with Enumeration literals.

- The stereotypes that are associated with metaclasses other than the Class metaclass are neither applied nor selected automatically by the tool at runtime.
Tools Limitations (2/3)

- The tool has a number of views to show the different aspects of the model with their limitations:
  
  - **Diagram View:** When a UML diagram and GRL diagram are created together, the diagram view reveals all of the diagrams but not the customized tool palette for the GRL diagram. Only the Class diagram tool palette is shown, as the GRL diagram is based on the Class diagram.
  
  - **Standard View:** Conversely, the standard view only shows the GRL diagram and its tool palette, but no other UML diagrams.
Tools Limitations (3/3)

• Decompositions differ visually in the jUCMNav example from the Tau GRL profile. This is because there is no customized appearance for links in Tau GRL profile.

• Tau does not support OCL implementation for profiling. There is a notion of informal constraints in Tau that are limited to text.
Profile Analysis

- Integration with UML  **Satisfied**
  - UML diagram and GRL Diagram connected with each other
  - Reusability of constructs
UML Profile for Goal-oriented Modelling
• Diagram Pollution Avoidance **Satisfied**
  - Separate GRL editor
  - Separate customized tool bar
• Metamodel Stability \textit{Satisfied}
  – Considered metamodel is undergoing standardization at ITU-T

• Implementability of the Profiling Mechanism \textit{Satisfied}
  – Implementation in both Stereotype Mechanism and Metamodel Extension Mechanism
## Comparison with Previous Work

<table>
<thead>
<tr>
<th></th>
<th>Integration With UML</th>
<th>Diagram Pollution Avoidance</th>
<th>Metamodel Stability</th>
<th>Tool Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Template based analysis of GRL</td>
<td>Not Satisfied</td>
<td>Not Satisfied</td>
<td>Not Satisfied</td>
<td>Not Satisfied</td>
</tr>
<tr>
<td>UML profile for Enterprise Goal Modelling</td>
<td>Satisfied</td>
<td>Not Satisfied</td>
<td>Not Satisfied</td>
<td>Partially Satisfied</td>
</tr>
<tr>
<td>UML profile for Softgoal by use case driven approach</td>
<td>Satisfied</td>
<td>Not Satisfied</td>
<td>Partially Satisfied</td>
<td>N/A</td>
</tr>
<tr>
<td>Using UML to reflect non-functional requirements</td>
<td>Not Satisfied</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>UML Profile for Goal-oriented Modelling</td>
<td>Satisfied</td>
<td>Satisfied</td>
<td>Satisfied</td>
<td>Satisfied</td>
</tr>
</tbody>
</table>

**UML Profile for Goal-oriented Modelling**
Conclusion

• Tool-supported, UML profile for goal-oriented modelling will make GRL more usable and understandable for UML users.
  – Editor has some limitations in terms of visualization
• Thesis contains more examples and analysis results
• Thesis also contains step-by-step instruction on how to create profiles
Thank you!

• Muhammad Abid
  – Ph.D. student
  – University of Ottawa
  – mabid006@uottawa.ca
  – Master of Computer Science thesis available upon request

• jUCMNav Tool
  – http://jUCMNav.softwareengineering.ca/jucmnav/