

GRL Modeling and Analysis with jUCMNav

Daniel Amyot¹, Gunter Mussbacher², Sepideh Ghanavati¹, and Jason Kealey³

¹School of Electrical Engineering and Computer Science, University of Ottawa, Canada
damyot@eecs.uottawa.ca, sghanava@eecs.uottawa.ca

²Dept. of Systems and Computer Engineering, Carleton University, Ottawa, Canada
gunter@sce.carleton.ca

³LavaBlast Software Inc., Gatineau, Canada
jkealey@lavablast.com

Abstract. The Goal-oriented Requirement Language (GRL), part of the User Requirements Notation (URN) standard, is used to model and analyze stakeholder goals and requirements. jUCMNav is a free Eclipse-based modeling, analysis, and transformation tool for URN that was first released in 2005. This paper reviews the status of this tool (version 4.4, the 20th official release), highlights its main features, and briefly discusses future development plans.

Keywords. Analysis, Eclipse, Goal Modeling, Goal-oriented Requirement Language, jUCMNav, User Requirements Notation.

1 Introduction

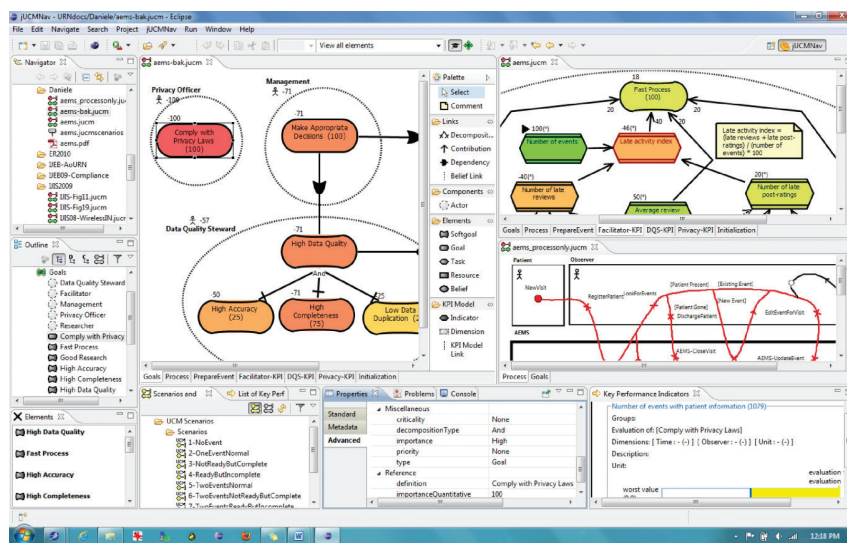
The User Requirements Notation (URN) is a graphical requirements modeling language that combines two complementary views: the Goal-oriented Requirement Language (GRL) for stakeholder objectives and decision rationales, and the Use Case Map (UCM) notation for scenarios and business processes combined with architectural components [4]. In November 2008, URN was officially standardized by the International Telecommunications Union as Recommendation Z.151. The design of URN itself started more than a decade ago, and this notation is now used for requirements engineering activities in a multitude of different contexts all around the world [3].

In 2005, students at the University of Ottawa started developing *jUCMNav*, an Eclipse-based tool for UCM modeling. Roy extended it to support GRL modeling and analysis [9], and Kealey later added UCM analysis [6]. This free, open source project has since benefited from many contributions. Now at its 20th official release, jUCMNav (version 4.4) [5] has become the most comprehensive URN tool available.

2 jUCMNav Modeling and Analysis Features for GRL

jUCMNav supports the standard GRL notation, with several extensions. The main editor view provides the list of GRL/UCM diagrams in the model, together with palettes and context-sensitive pop-up menus. This editor has the particularity of prevent-

ing the construction of syntactically incorrect GRL models. Also, different diagrams in a URN model can reference the same element definitions (actor, intentional element, link, etc.), hence improving consistency and allowing for complex models to be split into multiple diagrams. jUCMNav includes standard views such as a navigator for projects and model files, various outlines, element properties (with tabs), and problems. Typical Eclipse features such as zooming are supported, in addition to unlimited undo/redo and copy/pasting of model elements in the same model, across models, or to the clipboard (e.g., to paste a diagram into a word processor document).



Interesting extensions to URN include: i) *Key Performance Indicators* (KPIs) in GRL, to monitor external sources of data and convert them to the conventional GRL [-100, 100] evaluation scale [8], and ii) aspect-oriented extensions with an aspect composition mechanism, currently only supported for UCMs [7].

For analysis, jUCMNav supports GRL *strategies* (i.e., initial satisfaction values for several intentional elements), and six evaluation algorithms [2]: quantitative, qualitative, two hybrid ones, quantitative with KPI functions/aggregation, and constraint-oriented. While the first five support bottom-up propagation only, the last one uses a constraint solver and is hence more generic than bottom-up and top-down propagation algorithms, at the cost of a lower performance. These algorithms are independent, but qualitative ones are often used when little knowledge about the goals is available, whereas quantitative ones become useful as deeper knowledge is available. The Scenarios and Strategy view enables modelers to create, manage, and execute strategies. Color feedback is provided in real-time during evaluations (from *denied* in red to *satisfied* in green). Moreover, UCM scenarios can be defined and simulated. In jUCMNav, GRL satisfaction values can be used as variables in conditions attached to UCM elements (e.g., in forks or timers) and be updated in UCM responsibilities. The UCM/GRL views of a URN model can hence influence each other during analysis.

URN supports model element grouping (*concerns*), traceability (*URN links*), and annotating/stereotyping (*metadata*). In addition, OCL rules applied to the URN metamodel can be defined/selected by users to assess additional static semantic or stylistic constraints (violations are reported to the Problems view). When combined, these mechanisms can support lightweight profiles, both by extending and restricting the usage of URN elements while remaining compliant with the Z.151 standard. The tool comes with predefined, user-selectable constraints, including rules supporting a GRL profile for i* [1]. OCL is also used to compute user-defined metrics on models.

jUCMNav supports many model transformations through importing (Z.151 XML format, GRL catalogues) and exporting (Z.151 XML format, GRL catalogues, Message Sequence Charts and Core Scenario Model from UCM, and .CSV for GRL strategy evaluations) mechanisms. Reports can also be generated in PDF, RTF, and HTML, and diagrams can be exported in various bitmap formats. An integration with IBM/Telelogic DOORS for full requirements management is also available.

3 Conclusions and Future Plans

Since 2006, jUCMNav has been used successfully in software/requirements courses in more than a dozen universities, and it is used in dozens of labs and companies for research and industrial projects. The tool and its user/developer documentation (help, metamodel, examples, tutorials, etc.) are freely available online at [5]. Future plans include usability improvements, a textual syntax for GRL and UCM, improved URN link management, interoperability with i*, and the support of aspect-oriented GRL.

References

1. Amyot, D., Horkoff, J., Gross, D., Mussbacher, G.: A Lightweight GRL Profile for i* Modeling. 3rd Int. Workshop on Requirements, Intentions and Goals in Conceptual Modeling (RIGiM 2009), ER Workshops. Springer, LNCS vol. 5833:254-264 (2009)
2. Amyot, D., Ghanavati, S., Horkoff, J., Mussbacher, G., Peyton, L., Yu, E.: Evaluating Goal Models within the Goal-oriented Requirement Language. International Journal of Intelligent Systems (IJIS), Vol. 25, Issue 8, 841-877 (August 2010)
3. Amyot, D., Mussbacher, G.: User Requirements Notation: The First Ten Years, The Next Ten Years. Invited paper, Journal of Software (JSW), Vol. 6, No. 5, 747-768 (May 2011)
4. ITU-T: User Requirements Notation (URN) – Language definition, ITU-T Recommendation Z.151 (11/08). Geneva, Switzerland (2008); <http://www.itu.int/rec/T-REC-Z.151/en>
5. jUCMNav 4.4.0 (2011); <http://softwareengineering.ca/jucmnav>
6. Kealey, J., Amyot, D.: Enhanced Use Case Map Traversal Semantics. SDL 2007: Design for Dependable Systems. Springer, LNCS vol. 4745:133-149 (2007)
7. Mussbacher, G.: Aspect-oriented User Requirements Notation. PhD thesis, School of Information Technology and Engineering, University of Ottawa, Canada (2010)
8. Pourshahid, A., Chen, P., Amyot, D., Forster, A.J., Ghanavati, S., Peyton, L., Weiss, M.: Business Process Management with the User Requirements Notation. Electronic Commerce Research, 9(4), Springer, 269-316 (December 2009)
9. Roy, J.-F., Kealey, J., Amyot, D.: Towards Integrated Tool Support for the User Requirements Notation. SAM 2006: Language Profiles. Springer, LNCS 4320:183-197 (2006)