

# The RE-Tools: A Multi-notational Requirements Modeling Toolkit

Sam Supakkul  
Travel Network Architecture  
Sabre Inc.  
Southlake, Texas, USA  
sam.supakkul@sabre.com

Lawrence Chung  
Department of Computer Science  
The University of Texas at Dallas  
Richardson, Texas, USA  
chung@utdallas.edu

**Abstract**—Requirements engineers need to understand and model different aspects of organizations and systems under construction, and may need to use different modeling notations. However, most modeling tools support only one (or at most a few notations), hindering requirements engineers from using the most appropriate notations for the particular modeling task. The RE-Tools is an open-source toolkit implemented using a UML Profile for StarUML, an open-source UML modeling tool. The toolkit supports many leading requirements modeling notations, including the NFR Framework, the *i\** Framework, KAOS, Problem Frames, and UML. Each of these notations may be used for modeling independent corresponding diagrams or together with non-functional requirements (NFRs). The toolkit also supports the original qualitative reasoning of the NFR Framework and augments with a quantitative one.

## I. INTRODUCTION

Different notations have been used to model different aspects of organizations and systems under construction during requirements engineering. These notations include the NFR Framework [1][2], the *i\** Framework [3], KAOS [4], Problem Frames [5], and UML. Each emphasizes different aspects, hence is good for different things. Additionally, some requirements analysis and reasoning, such as goal achievement and trade-off analysis, are performed based on these notations. However, most modeling tools support only one (or at most a few notations), hindering requirements engineers from being aware of, considering, or using the most appropriate notations for the modeling and analysis task. The problem may be alleviated by using a drawing tool, such as Microsoft Visio. Still, using primitive drawing constructs of nodes and links, without any built-in support for the ontological concepts of the notations, is time-consuming, and the analysis must be done manually.

The RE-Tools [6] is an open-source toolkit that supports the requirements modeling notations. It has been downloaded from about 60 countries, used in teaching (RE courses at Univ. of Trento and Univ. of Texas-Dallas) and in research in Australia, Brazil, Canada, China, France, UK, and the US.

## II. SUPPORTED NOTATIONS

Figure 1 shows sample diagrams of the supported notations, which may be created separately or with non-functional requirements (NFRs) as shown.

## III. INTEGRATED QUALITATIVE AND QUANTITATIVE REASONING

The RE-Tools supports the original qualitative reasoning of the NFR Framework, but extends to support both open-world and closed-world assumptions and augments with a quantitative weight-based trade-off analysis [7], an adaptation of the lightweight quantitative options evaluation [8].

Figure 1(c) shows an example of the integrated reasoning where user-defined weights for prioritized criteria (1.0 for high/!, 0.5 for medium/!, and 0.2 for low) and for contribution links (1.0 for Make/++, 0.5 for Help/+, -1.0 for Break/-, and -0.5 for Hurt/-) are displayed in gray color, and system-calculated scores in blue. The tool recommends the most desirable leaf-level alternative for *Confidentiality* using a depth-first selection algorithm. Specifically, the selection between *Password* and *Biometrics* is not made until the selection between *Fingerprint* and *Retina Scan* is made first, where the latter is selected for its higher score of 0.5, over 0.35 of the *Fingerprint* alternative. *Biometric* is then selected for its higher score of 0.5, over 0.15 of the *Password* alternative. A portion of *Biometric*'s score (0.3/0.5) is inherited from *Retina Scan*'s score that accounts for the positive and negative correlations with *Trustworthiness* and *Cost* (0.5 and -0.2 respectively). The tool labels the selected alternatives (*Retina Scan* and *Biometrics*) as *satisfied* [1] (depicted by check marks) and uses the label evaluation procedure [1] to qualitatively propagate the labels upward in the goal graph to determine the impacts on high-level goals.

## IV. IMPLEMENTATION AND EXTENSIBILITY

The RE-Tools extends StarUML [9], an open-source UML modeling tool, using a UML Profile with stereotypes and tagged values to represent the concepts in the notations. The reasoning and icons are implemented using C#, StarUML's COM-based API and notation extension language.

Through the common model repository, the RE-Tools can be extended to provide additional features. For example, it has been extended to support the capturing, organizing, and reusing of NFRs as patterns [7]. Other possible extensions include model transformations from one notation to another (e.g., from an agent/goal model to a UML Use-Case model) or from one phase to another (e.g., from requirements to an architecture or a design).

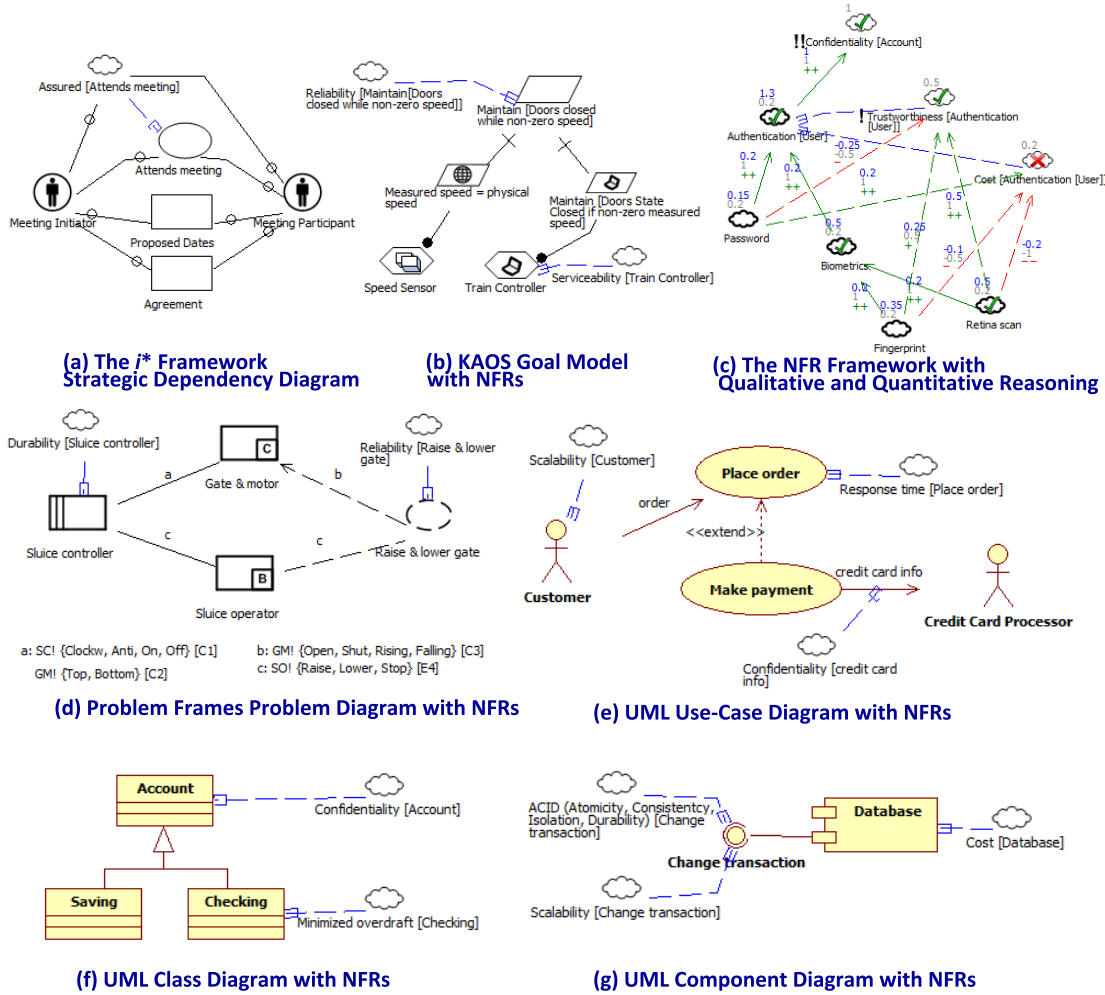


Figure 1. Sample Diagrams of the Supported Requirements Modeling Notations

## V. LIMITATIONS AND FUTURE WORK

StarUML was chosen, during the inception of the RE-Tools, for its better graphical support for custom notations. However, partly due to the lack of built-in Object Constraint Language (OCL) support by StarUML, syntax checking for the notations is currently lacking. Depending on COM-based StarUML also makes the RE-Tools available only on Microsoft Windows. We are exploring options to add syntax checking (e.g., using an external OCL tool), and to port the toolkit to a more portable platform, such as Eclipse.

It is important, but difficult, to maintain a large structure of different models without an ability to cross-reference among them. Currently, contribution links can be used between elements from different notations, although they are not automatically established. In the long run, we plan to investigate (i) automatic recognition of links between diagrams, (ii) semantics-preserving mapping between diagrams – potentially a challenging task, and (iii) support of other notations, e.g., a business process modeling notation.

## ACKNOWLEDGMENTS

We would like to thank Michael Jackson, John Mylopoulos, Axel van Lamsweerde, and Eric Yu for their valuable suggestions and encouragements. Our thanks also go to the anonymous reviewers for their valuable comments, to Michael Lloyd Davies, Rick Ratliff, John Tsang, and Steven Zhou of Sabre Travel Network Architecture for their valuable suggestions and encouragements, and to Sabre Inc. for its support of the RE-Tools.

## REFERENCES

- [1] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*. Kluwer Academic Publishers, 2000.
- [2] L. Chung and J. Leite, "On Non-Functional requirements in software engineering," in *Conceptual Modeling: Foundations and Applications*, A. T. Borgida, V. K. Chaudhri, P. Giorgini, and E. S. Yu, Eds., 2009, pp. 363–379.
- [3] E. Yu, P. Giorgini, N. Maiden, and J. Mylopoulos, *Social Modeling for Requirements Engineering*. The MIT Press, 2011.
- [4] A. van Lamsweerde, *Requirements Engineering: From System Goals to UML Models to Software Specifications*. Wiley, 2009.
- [5] M. Jackson, *Problem frames: analyzing and structuring software development problems*. Addison-Wesley, 2000.
- [6] www.utdallas.edu/~supakkul/tools/RE-Tools.
- [7] S. Supakkul, T. Hill, L. Chung, T. Tun, and J. Leite, "An NFR pattern approach to dealing with NFRs," in *18th IEEE Intl. Req. Eng. Conf.*, 2010, pp. 179–188.
- [8] A. van Lamsweerde, "Reasoning about alternative requirements options," in *Conceptual Modeling: Foundations and Applications*, A. T. Borgida, V. K. Chaudhri, P. Giorgini, and E. S. Yu, Eds., 2009, pp. 380–397.
- [9] http://staruml.sourceforge.net.