A Lightweight GRL Profile for i* Modeling

Daniel Amyot\textsuperscript{1}, Jennifer Horkoff\textsuperscript{2}, Daniel Gross\textsuperscript{3}, and Gunter Mussbacher\textsuperscript{1}

\textsuperscript{1} SITE, University of Ottawa  
\{damyot,gunterm\}@site.uottawa.ca  
\textsuperscript{2} Department of Computer Science, University of Toronto  
jenhork@cs.utoronto.ca  
\textsuperscript{3} Faculty of Information, University of Toronto  
daniel.gross@utoronto.ca

Abstract. The i* framework is a popular conceptual modeling language for capturing and analyzing socio-technical motivation and properties of complex systems in terms of actors, their intentions, and their relationships. In November 2008, the International Telecommunications Union finalized the standardization of the User Requirements Notation (URN). URN is composed of two loosely coupled yet integrated sub-languages: the Goal-oriented Requirement Language (GRL), which is an intentional modeling language based on a subset of i*, and the Use Case Map notation for representing and capturing high-level system scenarios and structures. GRL was specifically defined in a non-restrictive way to foster the development and use of different agent and/or goal modeling approaches and techniques. However, because of its permissiveness, GRL can be used in ways that deviate from conventional i* modeling guidelines. In addition, some i* concepts do not have equivalent first-class concepts in GRL. In this paper, we present a lightweight GRL profile for i* that takes advantage of GRL's extensibility features to capture missing i* concepts. The profile presents formal constraints on the use of GRL and its extensions to restrict it to an i* style. Using GRL constrained by this profile enables GRL modeling and analysis tools to be used for i* models, and ensures that resulting i* models conform to an international standard and that they can be integrated with Use Case Maps. Variants and extensions of the original i* can also be supported in a similar way. This profile is implemented in the jUCMNav modeling tool.

Keywords: Goal-oriented Requirement Language, i*, jUCMNav, OCL, profile, User Requirements Notation.

1 Introduction

The i* modeling framework [12, 13] introduced aspects of intentional and social modeling and reasoning into information system engineering methods, especially at the requirements level. Unlike traditional systems analysis methods which strive to abstract away from the people aspects of systems, i* recognizes the primacy of social actors. Actors are viewed as being intentional, i.e., they have goals, beliefs, abilities, and commitments, which must be discovered, captured and analyzed. The analysis
focuses on how well the goals of various actors are achieved given some configuration of relationships among human and system actors, and what reconfigurations of those relationships can help actors advance their strategic interests. Such analysis supports many software and system requirements engineering activities.

The i* framework has stimulated considerable interest in a socially-motivated approach to systems modeling and design, and has led to a number of extensions and adaptations, many of which are discussed in the i* Wiki [6]. One of these adaptations was recently standardized by the International Telecommunications Union (ITU-T) as part of the User Requirements Notation (URN – Recommendation Z.151) [7]. URN combines the Goal-oriented Requirement Language (GRL) with the Use Case Map (UCM) scenario notation in a single language, with a mature and well-defined metamodel supplemented by a concrete graphical syntax.

GRL supports many of the core concepts of i*, including actors, intentional elements, dependencies, contributions, and decompositions. However, GRL also differs from i* in a number of ways, such as the following:

1) **Missing first-class concepts in GRL:** i* contains concepts that are missing from GRL. For instance, GRL has only one type of actor, whereas i* also defines the notions of roles, agents and positions.

2) **GRL permissiveness:** GRL is voluntarily permissive in how intentional elements can be linked to each other. This is meant to support the wide variety of ways people actually create goal models [5]. However, i* proposes more specific and restrictive usages of relationships. For instance, an i* contribution link cannot have a task as a destination.

3) **Additional concepts in GRL:** GRL contains additional first-class concepts such as strategies (for the analysis of GRL models), metadata, and URN links (which enable the creation of typed links between any GRL/UCM elements).

In this paper, we present a lightweight profile for GRL that enables one to create goal models in a particular i* style according to the i* Guide in [6] and Yu’s work [12, 13]. We take advantage of URN links and metadata to create relationships and stereotypes (respectively) for the missing GRL concepts found in i*. We specify constraints in UML’s Object Constraint Language (OCL) [9] in order to restrict the usage of GRL to commonly used i* guidelines. We say that this profile is lightweight because it uses simple extensibility mechanisms and it does not require the extension of the URN metamodel or the use of heavyweight profiling mechanisms à la UML.

We also provide tool support for this profile with the jUCMNav tool, an Eclipse plug-in for the creation, analysis, and transformation of URN models [8, 10]. jUCMNav supports the notion of metadata together with an OCL engine that can check violations of user-defined constraints [2], enabling low-cost language customization.

A profile enabling the creation of GRL models in an i* style allows i* models to follow the standard defined in Z.151, including its interchange format. In addition, the use of the jUCMNav tool for i* models provides support for the division of models into consistent views (addressing scalability), the application of various pre-defined and automated quantitative and qualitative evaluation algorithms (with easy addition of new ones), the integration with UCMs, and simple modification or addition of constraints (for handling other variants and extensions of i*).