Eight Deadly Sins of GRL

Gunter Mussbacher[1], Daniel Amyot[2], and Patrick Heymans[3]

[1]Dept. of Systems and Computer Engineering, Carleton University, Canada
gunter@sce.carleton.ca
[2]School of Electrical Engineering and Computer Science, University of Ottawa, Canada
damyot@eecs.uottawa.ca
[3]PReCISE Research Centre, University of Namur, Belgium
phe@info.fundp.ac.be

Abstract. Goal modeling languages are now an integral part of requirements engineering. They allow for the systematic capture of rationales for stakeholder needs and enable the reasoning about potential system solutions. The goal-oriented Requirement Language (GRL) is both a typical goal modeling language and an international standard, yet it suffers from many shortcomings, akin to deadly sins, that relate to its concrete syntax, semantics, approach to modularity, analysis, and extensibility. Based on 10 years of experience using GRL, we discuss several shortcomings and point to relevant future work areas.

Keywords. Goal Modeling, GRL, Syntax, Semantics, Modularity, Analysis.

1 Introduction

Goal modeling forms an important part of requirements engineering. Over the last decade, through experience developing, standardizing, and using the Goal-oriented Requirement Language (GRL) [9], we have observed several important issues that we consider to be akin to “deadly sins”. We also suspect that many of these sins could be generalized to other goal modeling languages such as i*, KAOS, and Tropos, hence their general interest to the goal modeling community. In the following section, we briefly set out our research objectives, and in Section 3, we discuss the identified issues. Section 4 provides our conclusion and reports in a few words on future work.

2 Research Objectives

We aim to improve the state of the art of goal modeling by first identifying shortcomings in one of the mainstream goal modeling languages (GRL), then analyzing these shortcomings systematically, investigating ideas on how to address them, and finally providing solutions for these shortcomings that can be incorporated into the GRL standard and perhaps into other goal modeling languages. Here, we report on the currently identified eight major shortcomings, analogous to deadly sins (the seven common ones plus Despair) of GRL language designers and tool builders (including us).

3 Scientific Contributions

The list of eight sins of goal modeling languages in this section starts with issues related to the syntax of goal modeling languages (Section 3.1), then continues with semantic issues (Section 3.2), and concludes with issues regarding modularity (Sections 3.3, 3.4, and 3.5), analysis (Sections 3.6 and 3.7), and extensibility (Section 3.8). Whenever applicable, dependencies between issues are stated.

3.1 Pride: Lack of Support for Multiple Concrete Syntaxes

Goal modeling languages are usually very proud of their visual representations. Goal models, however, tend to be complex with many relationships among many intentional elements and actors. In addition, it is difficult for one single concrete syntax to support the many types of goal language users in their various modeling and analysis tasks. For example, while GRL's graphical syntax helps with the understanding and analysis of goal models as an output representation, it is often inefficient as an input representation as it tends to slow down the creation and maintenance of complex models. Being largely based on i*, GRL's notation also likely suffers from many of the deficiencies identified for i* in terms of cognitive effectiveness [12]. The current syntax could be revised to fix these deficiencies, and new graphical syntaxes could even be added for specific types of users. Moreover, a more practical textual concrete syntax, like a programming language with a suitable Integrated Development Environment, could *also* be made available, this time mainly as an alternative input representation. A proposal was made by Liu and Yu in their original GRL proposal [11] but was never followed-up and it never made it to the standard. Similarly, for users unfamiliar with goal modeling, a tabular-based representation à la House of Quality [7] may be used. Fig. 1 shows an example of what Fig. 2 could look like in a tabular form, e.g., intentional elements are shown with their types, links (from rows to columns), and bindings to actors (shown as importance levels between 0 and 100).

Intentional elements	*Links (A: AND, O: OR, D: Dependency, Others: contribution levels)*												*Actors*				*Type*
	Reduce societal effects of dengue	Disease surveillance	Vector control	Vector surveillance	Perform dengue premise surveillance	Test for pesticide efficacy	Field agent-driven assessment	Home owner / tenant self assessment	Provide regular reports to SVCO	Keep public informed accurately a. t.	Do a good surveillance job	Ensure privacy	Dengue S&C Syetem	State Vector Control Official	Field Agent	Home Occupant	
Reduce societal effects of dengue		A	A	A									100				S
Disease surveillance													0				T
Vector control													0				T
Vector surveillance					A	A							0				T
Perform dengue premise surveillance							O	O	+				0				T
Test for pesticide efficacy													0				T
Field agent-driven assessment											D	−	0				T
Home owner / tenant self assessment												+	0				T
Provide regular reports to SVCO													80				S
Keep public informed accurately a. t.									D					0			S
Do a good surveillance job															0		S
Ensure privacy																0	S

Fig. 1. Example of Potential Tabular Representation of a Goal Model

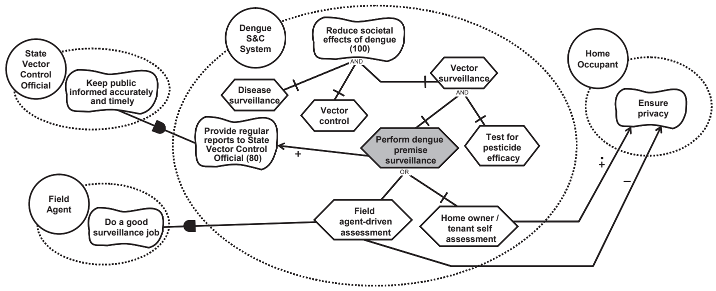

Fig. 2. Example for Modularity Issues

The various XML formats proposed over the years [4,9] are good as interchange formats, but not for human input. This concrete syntax issue is orthogonal to all other issues described in Section 3.

3.2 Envy: Lack of More Formal Semantics

Goal languages are often envious of the maturity of other modeling languages in terms of standardized semantics. GRL possesses a formal abstract syntax in the form of a standard metamodel. However, the meaning of a GRL model is largely defined by the evaluation mechanism chosen for that model (e.g., various categories of mechanisms are presented in [3], with other analysis techniques in [8]). The formal underpinning for evaluation/analysis mechanisms is rather ad hoc. Consequently, it is difficult to understand why one mechanism should be chosen over another. One option to improve formality is the definition and validation of a reference formal semantics for GRL rooted in mathematics (instead of typical operational semantics in the form of algorithms), for instance by following the guidelines proposed in [6]. This could be followed by the development of compliant interpretations in the form of constraint satisfaction problems, for which powerful analysis techniques and tools already exist.

3.3 Lust: Lack of Support for Interface Definitions

While it is somewhat possible to consider an actor as an encapsulated unit, this is quite difficult for a task (or any other intentional element) that is being decomposed into lower level elements. Even with actors, a clear interface definition is absent, leading to lustful situations between modeling elements. Dependencies help to a certain extent to separate goal trees of various actors, but relationships among elements of different actors may occur at any level of abstraction (see how the relationships between the four actors in Fig. 2 manifest themselves at different decomposition levels).

3.4 Sloth: Lack of Support for Relationship Abstractions

In GRL, there is no support (both from a language design point of view and from a tool support point of view) for describing the relationships of an intentional element based on the relationships of its constituent elements. As an example, consider what the relationships of the highlighted task in Fig. 2 should be. Only one contribution is shown for this task, but several others exist when its lower-level elements are taken into account. Of course, not everything needs to be shown in a partial view such as a goal graph, but there should be some way to view *all* relationships of any given actor or intentional element. Tool builders are slothful in their support for dynamically deriving, from the complete goal model, a view for a particular element that takes the relationships of its children into account. Furthermore, it should be possible to explore a goal model with interactive queries such as "show me all intentional elements for this actor" and "expand this intentional element's relationships by two levels" rather than just having static views (queries are a common feature of UML modeling tools).

While dynamic views are mostly a tool support problem for an already finalized goal model, an arguably even worse situation occurs when the goal model is built in the first place. Let us assume that a contribution has been defined for a high-level intentional element to indicate this important relationship at this level of abstraction. When decomposing the element into lower-level elements, one may want to also refine the contribution shown for the element. This, however, is typically not possible without having to remove the higher-level contribution and consequently losing the high-level overview. Note that this issue depends significantly on the Envy, Wrath, and Lust sins; formal semantics with proper support for abstractions would allow interfaces to be defined more easily at various levels of abstraction and vice versa.

3.5 Gluttony: Lack of Support for Separation of Concerns

GRL suffers from gluttony in that all concerns often need to be consumed at once. With aspect-oriented modeling becoming more and more part of mainstream modeling techniques, goal models should support the identification and encapsulation of concerns regardless of whether they are crosscutting or not. This may involve simple support for tagging an element as being part of a concern to full-fledged support for aspect-oriented matching and composition [13]. Model transformations [1] and suggestions made for i* [5] may also help address this modularity issue. Gluttony is largely an orthogonal issue to all other issues in Section 3.

3.6 Wrath: Analysis Anomalies

There is a frustrating and even infuriating interaction between the desire to view relationships at different levels of abstraction and the deployed evaluation mechanisms. At the heart of this issue is that a relationship shown at more than one level of abstraction must be taken into account only once during analysis. While the left and middle GRL graphs in Fig. 3 are equivalent, the right goal graph leads to a different evaluation result because relationships with the softgoal are shown many times, at different levels of abstraction. Furthermore, the contribution value for the higher-level task in the middle goal graph needs to be set to 40 to reflect that the contributions of both sub-tasks in the left goal graph are collapsed into one higher-level contribution.

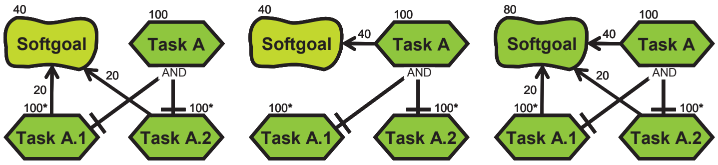

Fig. 3. Example for Analysis Anomaly

While GRL allows factors to be defined for intentional elements to differentiate their importance to a stakeholder, the importance of stakeholders themselves is typically not captured, even though all stakeholders do not have the same influence over the system under construction. Similarly, some intentional element may be non-negotiable from the point of view of one stakeholder but not from the point of view of another stakeholder, but this is also not captured.

3.7 Despair: Meaningless Numbers Problem

Modelers quickly lose hope in exploiting goal models in a useful way when the models do not properly reflect reality. One of the main difficulties when creating goal models is to specify appropriate contribution levels on contribution links. GRL should enable modelers to provide contribution ranges (to enable sensitivity analysis) or alternative levels as part of the goal model. A related issue is that it is often not possible to find a correspondence between the result of an evaluation and real life as it is difficult to translate real life into these seemingly arbitrary quantitative numbers and qualitative symbols, and vice versa. Greater reliance on key performance indicators (KPIs) derived from measurements of the real world may address the latter issue (as suggested in [14]). These issues are connected to Wrath as the ability to define sensible contribution levels is a prerequisite for a useful and sound evaluation of models.

3.8 Greed: Lack of Extensibility

Often, goal modeling languages do not allow language elements to be extended in a systematic way. This leads to different user communities that apply a particular goal modeling language in very greedy ways, and unfortunately, with incompatible dialects. An approach to address this issue is to provide the ability to a) define tags, stereotypes, or metadata for modeling elements, b) establish links between arbitrary modeling elements, c) group modeling elements, and d) define static constraints on the goal model (e.g., with OCL). This is partially supported in GRL [2], but there is much room for improving the packaging and modularization of such extensions. This issue has a dependency with Envy as the ability to define language extensions formally improves the semantic compatibility between variants of the modeling language. Better extensibility could also help tackle the lack of cognitive fitness discussed in Envy, but this may not always be the best way of producing a suitable dialect.

4 Conclusions and Future Work

We have presented a list of shortcomings of GRL with analogies to deadly sins. The most serious issues relate to vague definitions of goal model semantics (which essentially affects most sins), the lack of proper modularization, and the related difficulties when analyzing goal models at different levels of abstraction. The lack of modularized extensibility mechanisms is also a concern. We suspect that other popular goal modeling languages have committed many of these sins as well, to various extents.

In our future work on GRL, we plan to find absolution by a) solidifying the formal semantics with the help of mathematical and constraint-based techniques, b) investigating modules for stakeholders and intentional elements that provide precise interface definitions, c) providing textual and tabular syntaxes for goal models to improve model creation effectiveness, d) providing support for describing relationships of stakeholders and intentional elements based on their lower-level elements, and e) providing dynamic model exploration facilities (e.g., queries) in jUCMNav [10].

References

1. Alencar, F., Lucena, M., Silva, C., Santos, E., Castro, J.: Improving the Modularity of i* Models. 4th International i* Workshop, CEUR vol. 586:3-8 (2010)
2. Amyot, D., Horkoff, J., Gross, D., Mussbacher, G.: A Lightweight GRL Profile for i* Modeling. RIGiM 2009, ER Workshops. Springer, LNCS vol. 5833:254-264 (2009)
3. 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(8):841-877 (August 2010)
4. Cares, C., Franch, X., Perini, A., Susi, A.: Towards interoperability of i* models using iStarML. Comput. Stand. Interfaces, 33(1):69-79 (January 2011)
5. Franch, X.: Incorporating Modules into the i* Framework. Advanced Information Systems Engineering, 22nd Int. Conf., CAiSE 2010. Springer, LNCS vol. 6051:439-454 (2010)
6. Harel, D., Rumpe, B.: Meaningful modeling: what's the semantics of "semantics"? IEEE Computer, 37(10):64-72 (2004)
7. Hauser, J.R., Clausing, D.: The house of quality. Harvard Business Review, May-June, 63-73 (1988)
8. Horkoff, J., Yu, E..: Analyzing goal models: different approaches and how to choose among them. 2011 ACM Symp. on Applied Computing (SAC '11). ACM, 675-682 (2011)
9. 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
10. jUCMNav 4.3.0, http://softwareengineering.ca/jucmnav
11. Liu, L. and Yu, E.: GRL - Goal-oriented Requirement Language (2000); http://www.cs.toronto.edu/km/GRL/
12. Moody, D., Heymans, P., Matulevičius, R.: Visual syntax does matter: improving the cognitive effectiveness of the i* visual notation. Requirements Eng. J. 15(2):141-175 (2010)
13. Mussbacher, G.: Aspect-oriented User Requirements Notation. Ph.D. thesis, SITE, University of Ottawa, Canada (November 2010)
14. 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):269-316 (December 2009)