Towards Advanced Goal Model Analysis with jUCMNav

Daniel Amyot and Azalia Shamsaei
School of EECS, University of Ottawa, Canada
{damyot, asham092}@uottawa.ca

Jason Kealey and Etienne Tremblay
JUCM Software Inc.
{jkealey, etremblay)@jucm.ca

Andrew Miga, Gunter Mussbacher, and Mohammad Alhaj
Department of SCE, Carleton University, Canada
andrew_miga@sympatico.ca, gunter@sce.carleton.ca, malhaj@connect.carleton.ca

Rasha Tawhid, Edna Braun, and Nick Cartwright
Aviation Security Directorate, Transport Canada, Canada
{rasha.tawhid, edna.braun, nick.cartwright}@tc.gc.ca

Abstract. Goal modeling is an important part of various types of activities such as requirements engineering, business management, and compliance assessment. The Goal-oriented Requirement Language is a standard and mature goal modeling language supported by the jUCMNav tool. However, recent applications of GRL to a regulatory context highlighted several analysis issues and limitations whose resolutions are urgent, and also likely applicable to other languages and tools. This paper investigates issues related to the computation of strategy and model differences, the management of complexity and uncertainty, sensitivity analysis, and various domain-specific considerations. For each, a solution is proposed, implemented in jUCMNav, and illustrated through simple examples. These solutions greatly increase the analysis capabilities of GRL and jUCMNav in order to handle real problems.

Keywords: Analysis, Goal-oriented Requirement Language, jUCMNav, strategies, tool support, User Requirements Notation, visualization

1 Introduction

Goal modeling is an important part of requirements engineering activities. Goal models capture stakeholder and business objectives, alternative means of meeting them, and their positive/negative impacts on various quality aspects. The analysis of such models guides the decision-making process as well as the refinement of imprecise user requirements into precise system requirements.

The Goal-oriented Requirement Language (GRL), part of the User Requirements Notation (URN) [2,5], is a standard notation for goal modeling. GRL enables requirements engineers and business analysts to describe stakeholders (actors) and intentions (e.g., goals, softgoals, and tasks), together with their decomposition structure, dependencies, and contribution levels. Given initial satisfaction levels associated with some of the elements of a goal model (i.e., a strategy), tool-supported analysis techniques can determine the satisfaction levels of the other elements [1]. In particular, jUCMNav [3, 6] is a free Eclipse plug-in that enables the creation and management of complex GRL models. It also provides features to support various analysis algorithms that exploit strategies, to help visualize analysis results, and to generate reports.

Yet, the realities of complex application domains, such as regulatory compliance [8], have pushed the limits of the language and of current tool support. Through our experience modeling and analyzing real regulations with GRL, we have observed important issues related to the comparison of strategies and evolving models, the management of complexity of sets of strategies, the management of uncertainty related to contribution links, the sensitivity of analysis results when localized changes are explored, the usability of the standard GRL evaluation scale, the practicality of unilingual models in a multilingual environment, and facilities for handling strategies separately from their model.

This paper explains each of these issues and proposes solutions that we have implemented in the latest version of the jUCMNav tool, with simple but illustrative examples. We believe these solutions will help address similar issues beyond the regulatory compliance context. They may also inspire language designers to evolve other goal-oriented languages and their tools.

2 Strategy and Model Differences

In GRL (see metamodel extract in Fig. 1), a model can include *evaluation strategies*, which are sets of initial *evaluations* (quantitative value or qualitative labels) associated with *intentional elements* [5]. Strategies are also *grouped* for classification and convenience. Various qualitative, quantitative and hybrid propagation algorithms take these values and propagate them to the other intentional elements (through contribution, decomposition and dependency links), and to actors that contain intentional elements with non-null importance [1]. In GRL, the importance level of an intentional element to its actor is shown between parentheses (e.g., see Fig. 2). Intuitively, using a quantitative scale (as used in our examples), the satisfaction level of an intentional element is: the maximum of the children's evaluation values for an OR decomposition, the minimum for an AND decomposition, and the bounded weighted sum for contributions. jUCMNav also uses color feedback to highlight satisfaction levels (the greener the better, the redder the worse) as well as dashed lines for the border of intentional elements that are part of strategies (see Fig. 2).

Usually, many strategies are defined for a model to explore different global alternatives or tradeoffs in a decision support context, to represent as-is and to-be contexts, or to capture historical contexts (e.g., the situation or compliance level of the organization at different times). There is a need to compare strategies and to visualize this comparison in terms that the model user can understand. jUCMNav already supports the generation of reports (in PDF, RTF, and HTML formats) that contain a tabular representation of all strategies and their results. This is useful for sharing models and strategy evaluations with people who do not have access to the modeling tool, but this is not really amenable to the real-time analysis of differences between strategy results. The issue here is: *Can we highlight differences within the graphical model itself in order to provide more immediate feedback and support discussions between stakeholders around the model, its strategies, and the supporting tool?*

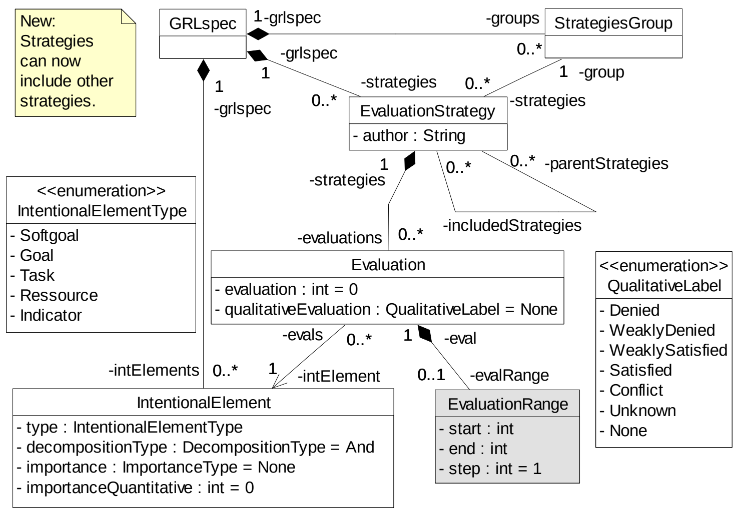
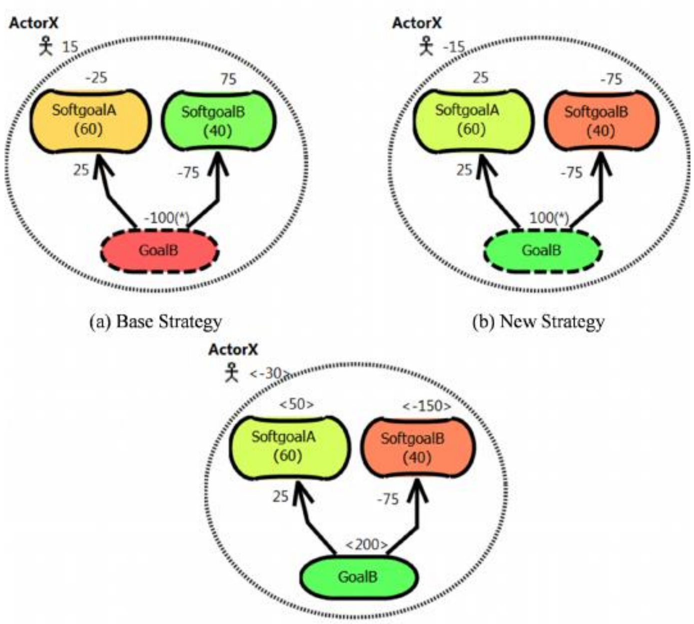


Fig. 1 Extract of URN metamodel – Strategies

To answer this question, we propose a new jUCMNav feature that highlights *strategy differences* visually in terms of evaluations of intentional elements and actors. The difference is computed between a base strategy (e.g., Fig. 2a) and a current strategy (e.g., Fig. 2b) on a per element basis (including actors). The standard GRL scale for quantitative evaluations goes from –100 (fully denied, shown in red) to 0 (neutral, in yellow) to +100 (fully satisfied). Consequently, the difference scale is [–200..200]. Differences are displayed between angle brackets (to differentiate them from normal satisfaction values), again with color feedback (<–200> in red, <0> in yellow, and <+200> in green), so the tradeoffs can be understood at a glance. Fig. 2c shows the difference results of our simple example; with the new strategy, ActorX becomes less satisfied by a difference of 30. jUCMNav allows one to select a base strategy and then switch between many alternative strategies to visualize (instantly) their differences.

In a context where the GRL models themselves and their strategy definitions evolve (e.g., as we gain more insights about the domain being modeled), another question is: *How can we highlight, understand, and control model evolution?* Ideally, *model differences* would need to be done at the level of GRL graphical model elements. However, this poses technical challenges, especially for the presentation of deletions and modifications of model elements and their properties.



(a) Base Strategy (b) New Strategy

(c) Difference: New Strategy – Base Strategy

Fig. 2 Strategy difference example

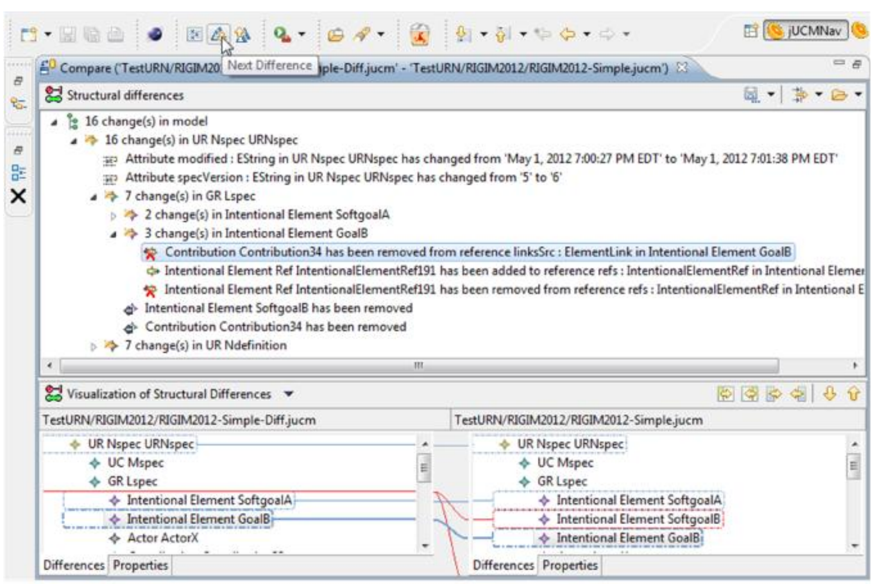


Fig. 3 URN model difference in jUCMNav based on EMF Compare

The approach we have prototyped in jUCMNav reuses the facilities of the *EMF Compare* plugin [4], a generic difference engine for modeling tools based on the Eclipse Modeling Framework (EMF). EMF Compare represents a simple and yet efficient solution to the comparison of URN/GRL models. For example, Fig. 3 displays the results of comparing the simple model used in Fig. 2 with one where we have removed SoftgoalB (including its incoming contribution), and changed some attributes. EMF Compare also allows one to copy changes (or merge) from one version to the other. Finally, EMF Compare offers means to filter out comparison results of little value (e.g., a change in the size or position of an element) in order to focus on the most important changes. However, filtering is left for future work in our context.

3 Complexity/Uncertainty Management and Sensitivity Analysis

Complexity in goal models can take many forms. One is related to the size of the models and the number of strategies to handle. jUCMNav already offers several features to handle models that include many diagrams (e.g., navigation, search, different views, and the sorting of diagrams). However, one issue remains: *How should we manage large collections of strategies?*

Our solution is to have a parent-child *inclusion relationship* between strategies (see the corresponding new association in Fig. 1). In essence, a parent strategy can now include another strategy, which means that the initial evaluations of the latter will be included automatically (i.e., reused) in those of the former. These included evaluations can then be overridden by parent evaluations (if they target the same intentional element), or complemented by additional evaluations. Strategy inclusion can be done recursively (across many levels). jUCMNav ensures that inclusion loops are avoided. This solution hence improves consistency and reduces the number of updates required when new strategies or model elements are added. It can also be combined with the strategy difference feature described in the previous section.

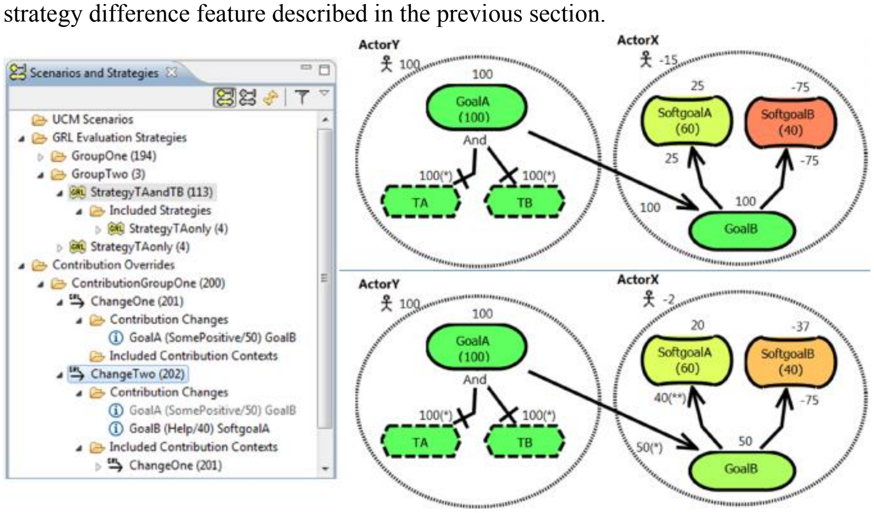


Fig. 4 Examples of strategy inclusions and of contribution contexts

As an example, the model in Fig. 4 (top) was evaluated against StrategyTAandTB (selected in the left view), which includes StrategyTAonly (initializing the evaluation of task TA with 100) and adds as a second evaluation, this time for task TB (100).

Another level of complexity lies in the uncertainty surrounding weights (or levels) of contribution links in goal models. It is often difficult for modelers to determine appropriate contribution levels for such links (see the Despair sin in [7]), and the real impact of using different levels is difficult to assess. The issue here becomes: *Can we investigate alternative combinations of contribution levels during analysis without having to produce and maintain different variants of a goal model?*

Our proposed solution is to support the concept of *contribution contexts*, which are to contribution levels what strategies are to evaluation values. As formalized in Fig. 5 (the dark gray metaclasses are new), a contribution context contains a set of *contribution changes*, which override the quantitative and/or qualitative contribution levels of contribution links in a GRL model. Like for strategies, contribution contexts are *grouped*, they can include other contexts, and they can be used in strategy differences.

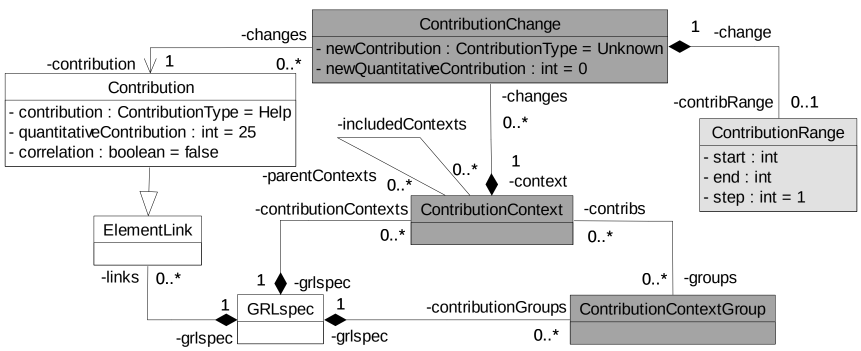


Fig. 5 Extract of URN metamodel – Contribution contexts

In the left view of Fig. 4, ChangeOne changes the contribution from GoalA to GoalB to 50, whereas ChangeTwo includes ChangeOne *and* overrides the contribution from GoalB to SoftgoalA with 40. The result of evaluating StrategyTAandTB against the model modified by ChangeTwo is shown at the bottom of Fig. 4. Note that (**) on a contribution indicates a direct change while (*) indicates an included change.

A third issue that touches both complexity and uncertainty is whether localized changes to a satisfaction level or to a contribution level actually impact significantly or not the satisfaction of high-level objectives in a goal model. This is akin to sensitivity analysis, which is the study of how the variation (uncertainty) in the output of a model can be attributed to different variations in the inputs of the model. The problem is as follows: *Can we support simple sensitivity analysis without having to declare strategies and contribution contexts for all single values of interest?*

Our proposed solution is to allow for *ranges* of values to be used for strategy evaluations (Evaluation Range in Fig. 1) and for contribution changes (ContributionRange in Fig. 5) instead of just single values. The *step* of a range is the increment by which we iterate from the *start* to the *end*. By associating a range to an initial evaluation, all other intentional elements impacted directly or indirectly will also have a range, but this time for computed values. Fig. 6 (top) shows an example where TB has an initial range of [75..100] with 5 as a step value. TA is not impacted, but all of the other intentional elements are. Their resulting ranges are also displayed. In addition, all intermediate values (for each iteration) are accessible as metadata, and hence visible as a tooltip by hovering over the desired element (SoftgoalA in Fig. 6). This simple sensitivity analysis enables the modeler or analyst to assess the impact of localized changes and to determine whether a change to an initial satisfaction value really matters or not.

A similar usage is possible for contributions. Fig. 6 (bottom) shows an example where the contribution from GoalA to GoalB is overridden by a [40..60] range with a step of 4. The results are shown for StrategyTAandTB, which does not include any evaluation range. Again, the impact on intentional elements can easily be assessed.

Sensitivity analysis in jUCMNav is currently limited to one dimension only, i.e., to a range for one evaluation or for one contribution. Allowing for more than one dimension to be explored at once would lead to visualization challenges (e.g., tables or cubes instead of linear arrays of values) that would negatively impact understanding. Other visualization schemes are required in that context. The support for ranges on the actors and possibly on importance values is also left to (near) future work.

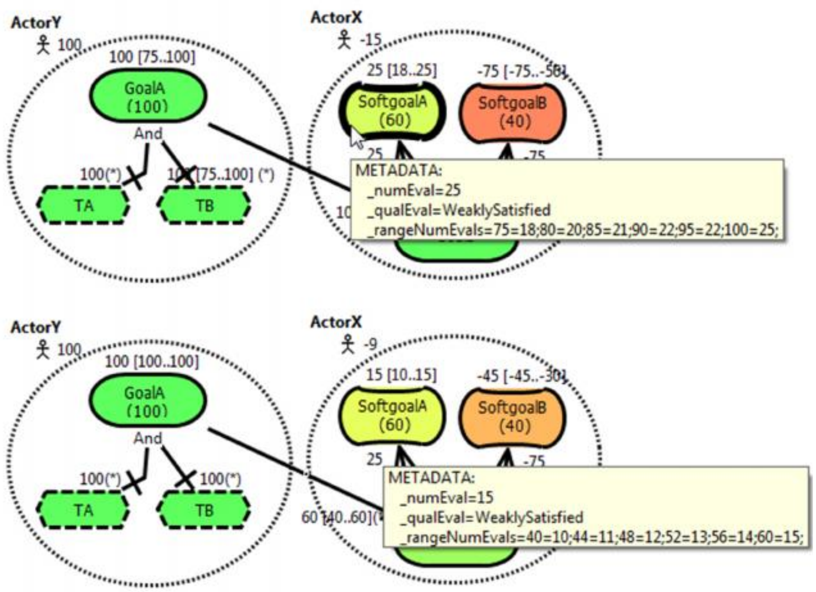


Fig. 6 Use of ranges for sensitivity analysis in strategy evaluations (top) and in contributions levels (bottom)

4 Domain Considerations During Analysis

While interacting with policy makers and other stakeholders, we realized that the standard GRL satisfaction range ([–100..100]) was really counter-intuitive to many people, even more so when a goal with a negative evaluation that has a negative contribution to another intentional element leads to a positive evaluation value for that element (see Fig. 2a). This issue was also raised by many undergraduate and graduate students to whom GRL was taught over the past 8 years. This problem is therefore stated as: *Can we support an alternative range of satisfaction values for domains where the standard one is counter-intuitive?*

We have implemented an alternative [0..100] evaluation scale (where 0 now means fully denied) and adapted the user interface (e.g., pop-up menus with predefined values) and the propagation algorithms accordingly. The color feedback in jUCMNav now also depends on the scale being used (with the new scale, 0 is red as there is no longer any negative satisfaction values, and 50 is yellow). Fig. 7 (left) shows the same model and strategy as in Fig. 2b, but evaluated with the new scale. Note that a satisfaction level of 25 is orange now, indicating partial dissatisfaction, rather than light green. Contributions are still allowed to be negative, but they cannot lead to a negative satisfaction values; this is why the evaluation value of SoftgoalB is 0, i.e., the lowest value allowed by this new scale. The modeler can choose between one scale or the other when creating a model. After a few weeks of usage and the training of nearly 50 people in the government on GRL for regulations, there is much ad hoc evidence that this indeed leads to a more intuitive interpretation (especially by non-experts in GRL) of goal models used for compliance analysis.

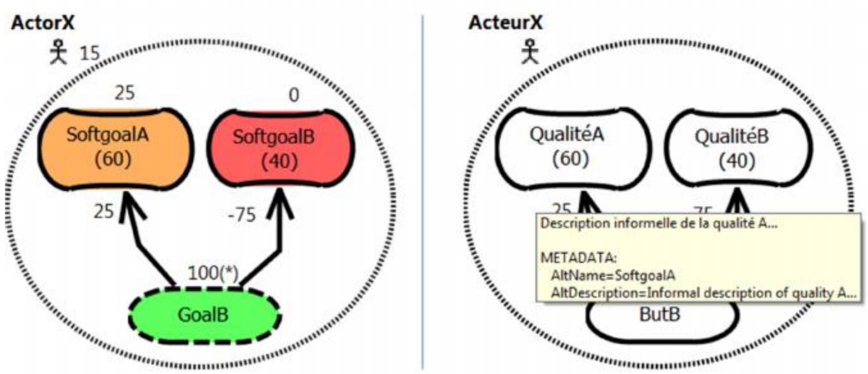


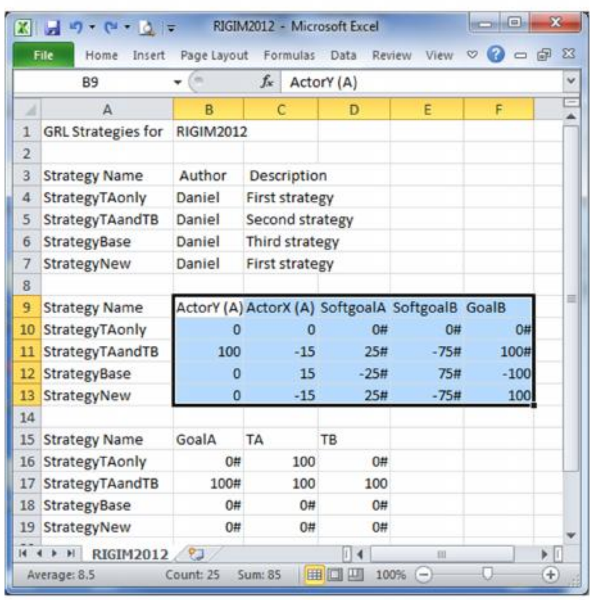
Fig. 7 Strategy evaluation in a [0..100] scale (left) and multilingual model (right)

Another interesting domain consideration is that in Canada, regulations are written in two languages (English and French). Obviously, creating French and English versions of a same model in not desirable. The issue here is: *Can we support models in multiple languages without having different models, to avoid maintenance issues?*

jUCMNav's user interface is already multilingual (and supports French and English), but this is sufficient as there is no way of attaching multiple names and descriptions to model elements. We implemented a feature that allows the modeler to switch between model languages and to provide alternative names and descriptions for model elements, including actors, goals, strategies, and diagrams. When switching languages, the name and description of each element are swapped with alternative values attached to the element as metadata. This is limited to two languages at the moment, but this could be extended to more than two in the future. Fig. 7 (right) presents the French version of the names and descriptions used in Fig. 7 (left). Both are stored in the same model and hence can be easily maintained as the model evolves (minimizing the risk of inconsistencies). There is no automatic translation at the moment as this was seen as potentially dangerous in a regulatory context, but this could likely be added in the future. The same feature is also being explored to support many levels of language in the same context (e.g., for regulation experts, and for non-experts).

One last interesting domain issue that we considered as part of our recent work relates to the fact that, sometimes, strategies need to be stored independently from models. In a compliance context, the people creating a GRL model may not have sufficient privileges to access strategies used to evaluate the model. For example, analyzing the impact of airport incidents might require access to highly confidential data used to populate initial values in the strategies. Moreover, strategies might be generated automatically from data sources (e.g., airport inspection reports) and their results consumed by other analysis and reporting tools (e.g., for Business Intelligence). Hence: *Can we handle strategies and their results separately from their GRL model?*

Our solution involves the *import/export of strategies*, with results, as simple comma-separated value (CSV) files. This enables one to split strategy definitions and results from the model, and hence they can be stored in different places and be restricted to particular users. This format is also easy to process as output (e.g., from a database, or from Microsoft Excel as seen in Fig. 8) or as input (e.g., to a business intelligence tool, or to Excel). Rows represent named strategies while columns represent mainly the actors (results only) and intentional elements. One particularity is that we separate, for intentional elements, results (suffixed with the # symbol, which can be removed easily for post-processing when needed) from definitions (no # symbol).



	A	B	C	D	E	F
1	GRL Strategies for	RIGIM2012				
2						
3	Strategy Name	Author	Description			
4	StrategyTAonly	Daniel	First strategy			
5	StrategyTAandTB	Daniel	Second strategy			
6	StrategyBase	Daniel	Third strategy			
7	StrategyNew	Daniel	First strategy			
8						
9	Strategy Name	ActorY (A)	ActorX (A)	SoftgoalA	SoftgoalB	GoalB
10	StrategyTAonly	0	0	0#	0#	0#
11	StrategyTAandTB	100	-15	25#	-75#	100#
12	StrategyBase	0	15	-25#	75#	-100
13	StrategyNew	0	-15	25#	-75#	100
14						
15	Strategy Name	GoalA	TA	TB		
16	StrategyTAonly	0#	100	0#		
17	StrategyTAandTB	100#	100	100		
18	StrategyBase	0#	0#	0#		
19	StrategyNew	0#	0#	0#		

Fig. 8 Strategies (definitions and results) as imported/exported CSV files

For example, GoalB is initialized with –100 in StrategyBase, but SoftgoalB is not initialized. Another feature is that, when there are many intentional elements, the format allows for a user-defined number of columns to be used, which is convenient for inputs from tools such as Excel (as less horizontal scrolling is required). Strategies then span multiple rows.

During an import, jUCMNav currently creates a new strategy group where the imported strategy definitions are stored (results with a # and actor evaluations are simply ignored). This allows for multiple versions of the strategies (e.g., compliance results evaluated at different times) to be used and then compared. Future work items for this mechanism include the support for strategy groups, included strategies, and ranges.

5 Conclusions

This paper presented many concrete issues with the applicability of goal modeling, and particularly of GRL and jUCMNav, for supporting analysis in a real context. We proposed and implemented a collection of advanced analysis and management features to handle these issues. Although these features represent major advancements over past jUCMNav versions [3], many remaining items for future work have been identified. The real usefulness and validity of these new features also requires further experiment. Regulatory compliance was used here as a context but we suspect that the identified issues and proposed solutions will also be valid for other domains, and probably even for other languages and tools. We finally plan to propose some of our language extensions to become part of a future release of the URN standard [5].

References

1. Amyot, D., Ghanavati, S., Horkoff, J., Mussbacher, G., Peyton, L., and Yu, E.: "Evaluating Goal Models within the Goal-oriented Requirement Language". *International Journal of Intelligent Systems*, John Wiley & Sons, Vol. 25, Issue 8, pp. 841–877 (2010)
2. Amyot, D. and Mussbacher, G.: "User Requirements Notation: The First Ten Years, The Next Ten Years". *Journal of Software (JSW)*, Vol. 6, No. 5, pp. 747–768 (May 2011)
3. Amyot, D., Mussbacher, G., Ghanavati, S., and Kealey, J. "GRL Modeling and Analysis with jUCMNav". *5th Int. i* Workshop (iStar 2011)*, Trento, Italy. CEUR-WS, Vol-766, pp. 160-162 (Aug. 2011)
4. Eclipse Foundation: *EMF Compare*. 2012. http://www.eclipse.org/emf/compare/
5. International Telecommunication Union: *Recommendation Z.151 (11/08), User Requirements Notation (URN) – Language definition*. 2008. http://www.itu.int/rec/T-REC-Z.151/en
6. jUCMNav, Version 5.1.0, University of Ottawa, http://softwareengineering.ca/jucmnav
7. Mussbacher, G., Amyot, D., and Heymans, P.: "Eight Deadly Sins of GRL". *5th Int. i* Workshop (iStar 2011)*, Trento, Italy. CEUR-WS, Vol-766, pp. 2–7 (Aug. 2011)
8. Tawhid, R., Alhaj, M., Mussbacher, G., Braun, E., Cartwright, N., Shamsaei, A., Amyot, D., Behnam, S.A., and Richards, G., "Towards Outcome-Based Regulatory Compliance in Aviation Security". *Requirements Engineering (RE'12)*, USA. IEEE CS, 2012 *(to appear)*