Reinventing Goal-Based Requirements Modeling

Vikas Shukla and Guillaume Auriol

CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
Univ. de Toulouse, INSA, LAAS, F-31400, Toulouse, France
vshukla@laas.fr gauriol@laas.fr

Abstract. Understanding user needs, requirements, architecture specifications, and design specifications for a system holds up-most importance in a systems engineering project. The early phase requirements engineering deals with elicitation of goals, objectives and environment of the system under development and determine the needs and requirements of the various stakeholders. The needs, requirements should traced to the various rationales of stakeholder with their preferences. In this paper, we provide a goal-based requirements modeling methodology and language to understand the needs, requirements,... and represent them in a much clear manner to improve the quality of requirements written for the project and their early phase traceability. We also integrate the preference modeling of the various stakeholders for the various goals and hence provide a traceability for the requirements and their preferences in early phase of requirements engineering.

Keywords: Requirements Modeling, Goal-Based Requirements Engineering, Stakeholder preference, Traceability, Features.

1 Introduction

The primary goal of the *systems engineering* (SE) is the creation of a set of high quality products and services that enable the accomplishment of desired tasks and needs of the clients or user groups. Typical systems engineering project can be divided in to three phases: definition, development, and deployment [1]. A systems engineering project involves multiple stakeholders, in their various forms of roles and actors, during and after the previously mentioned SE project phases. These stakeholders and their various roles lead to various needs and requirements. *Requirements engineering* (RE) activities en-globe activities like stakeholder identification, requirements elicitation, modeling, analysis, documentation, verification and validation, negotiation, etc. All these activities require certain understanding of the requirements itself.

Poor RE is the reason for majority of all the challenged and failed SE projects. Many empirical studies previously carried out have indicated that poor RE leads to poor requirements, faulty design, poor requirement traceability, rework and cost/budget overflows [2–4]. Requirements modeling is carried out during early phase of the RE. There are a few of approaches like: Goal-Oriented RE (GORE), Scenario-Based RE (SBRE), Problem-Based RE (PBRE), and Aspect-Oriented RE (AORE). The two most popular and referenced modeling methodology are goal-based and scenario-based RE, owing to the benefits and ease it provides during the requirement modeling phases. In this paper, the two referenced and used methodologies are goal-based and scenario-based RE. There are tools built upon popularly known GORE methodologies i* and KAOS, and have reported lots of success in industrial application because of ease of understanding it offers to both technical and non-technical stakeholders. But still there is some scope of improvement and certain difficulties and problems to be improved as discussed later in Section 2. RE research has focused on goals as a way of providing the rationales (why) for a system under development [5].

In this paper, we provide ways to model the requirements of the core and optional features of the system from early phases of RE and equally the stakeholders' preferences associated to them. We provide ways to visual modeling, which are more scalable and comprehensible to the engineers and the various stakeholders. We show how the viewpoints based modeling can be carried out from the very beginning and be carried out separately and integrated later. We call our proposed approach Comprehensive Requirement Modeling Language (CReML), which is based on Goal-Oriented Requirements Engineering (GORE) methods. CReML can be used complementarily with UML and hence help to provide better design specifications and insights to the system under study. Previously, it has been argued and shown that the GORE and SBRE complement each other during the requirement modeling phase of RE [6].

The major contributions of this paper include, a GORE based requirements modeling language with preference modeling and provisions for core or optional goal feature modeling together with much needed traceability and notations for domain assumptions. Demonstration of our CReML tool with an example demonstrating various aspects of the CReML developed.

This paper is organized as follows: Section 2 presents the early phase RE problems and goal-based RE. Section 3 presents the relevant state of art of goal-based RE. Section 4 presents our proposed approach. Section 5 presents an example of our approach. Section 6 concludes and presents future perspectives.

2 Early phase requirements engineering problems

Early phase requirements engineering start once the requirement elicitation process starts following to the interviews, questionnaires, ethnography and other elicitation techniques mentioned in literature. Through all the elicitation techniques the information gathered is converted to textual documents, often known as user-stories. These user-stories form the foundation of the requirements modeling (RM) processes. We have identified a few of the problems which seek attention and proper resolution during RM. ***Ease of Scalability:*** recent empirical studies using i* GORE methodology have shown that scalability can turn out to be big problem when modeling requirements for a sufficiently large projects either with different viewpoints or integrated modeling [7]. GORE based approach needs to be organized in a manner that their comprehensibility is independent of number of participants and number of requirements and views used during the modeling. ***Traceability of goals:*** often, during RM goals are elicited through stakeholders and as the project evolves, the complexity of the models may increase to an extent where it becomes tedious task to answer why a particular goal exists in the model and which particular stakeholder needs it. Also, it can be equally cumbersome to link a goal to a particular user story previously elicited, owing to syntactic differences [7]. There is need of dedicated mechanism to link goals to the user-stories previously documented during elicitation. ***Preference of multiple stakeholders over goals:*** in a systems engineering project, it is of great importance that most of stakeholders are satisfied with the various decisions taken during the product development and with the final resulting end product. A higher satisfaction among the stakeholders can be guaranteed if the various stakeholders preferences are taken into account in a transparent and holistic manner during the goal modeling. ***Multiple view-point requirement modeling:*** during the requirement modeling multi-view point modeling is often instrumental in understanding the system under study. Multi-view requirements modeling allows separation of concerns and can help to elaborate particular aspects of the system under study. But it becomes tedious task to combine these multiple views and present one single coherent and comprehensive models. Often, the resultant combined model is inconsistent and hard to understand, which demands significant amount of resources. Need of multi-view modeling has been often demanded in many previous works for particular actors, traceability and events [8, 9]. ***Modeling of core and optional features:*** modeling of core features and optional features from the very beginning of project can allow the engineers to have better understanding of the systems under study and lots of effort and resources can be saved if they can be modeled in early phase of RM. There are some dedicated feature modeling languages in the literature but this often leads to redundant efforts. ***Representation of domain assumptions:*** domain assumptions or beliefs are often implicit during the requirement modeling but often lead to goals and requirements. They are usually held by the stakeholders and sometimes designers. Their implicit nature during the RM may cause potential traceability errors and may cause worries for the quality control of the product. Domain assumptions usually become the basis of many decisions during the RE activities, they too need to be given requisite attention.

3 State of Art

Literature of GORE based methodologies is vast if we take in account all the RMLs mentioned. There are a few notable GORE frameworks such as i*, Tropos, KAOS, GBRAM, NFR, etc., but still there are various aspects to be improved upon as mentioned in Section 4. As previously mentioned, our approach refers to i* and KAOS owing to the proximity of our approach. i* and KAOS frameworks are apparently the two most popular GORE methodologies. The seminal work of Erik Yu [10] introduced the i* framework. It is hard to provide a fair comparison between them, as both of them have certain benefits and disadvantages when compared to each other [11]. Underlying principles of GORE were reinforced by Regev *et. al.* [12] by bringing together the various concepts from various methodologies. Recent works involving goals, preferences, and inconsistency have led to development of an abstract requirement modeling language called Techne [13]. Recent work have tried to address the optionality and preference of the requirements during the modeling [14]. The preference of the goals are marked on the the goal notation thus allowing to evaluate the importance of one goal with respect to another. Goal argumentation methods (GAM) were introduced to make it explicit the reasons of selecting a goal [15].

Tropos [16] framework was founded on social and intentional aspects of information system, used requirement modeling based on their operational environment. Tropos introduced textual syntax to allow the later phase formal analysis of the early requirements models done using i* to represent their social milieu. The major advantage given by i* based frameworks is they allow to represent the strategic relationships existing between the various stakeholders of the project. But many empirical studies [7, 11] have shown that the readability of the models designed using i* are greatly marred when the number of participants is sufficiently large. This problem comes due to layout used by the i* models, on the contrary it can be argued that the tree based layout of KAOS models are much better in this aspect of the goal modeling. One of the disadvantages of the KAOS models is its deficiency in modeling the strategic relationships of the stakeholders. Previously mentioned techniques for integrating preference in goal models do not help user in any readability aspect [14, 13]. In can be argued that surcharge of information increases the pressure on the designer or stakeholders. The way the data is represented and made available to the design engineer can be rendered more readable.

Currently, there are a variety of tools available for the GORE modeling such as Objectiver based on KAOS [17], recently introduced RE-TOOLS [18]. There are a few of light weight RMLs introduced recently like VLML [19]. Still there are numerous issues to be solved by a RML and lots of lessons learned during all there years of RE needs to be brought together to a standard GORE language. A standardized GORE notation based on KAOS seems to be most appropriate for this unifications of lessons learned and hence we propose in this paper few modifications into the KAOS modeling notations to the benefit of RE community.

4 Proposed Goal-Based RE Language

Our approach aims to use and devise techniques previously used and matured in domain of software engineering for the benefit of systems engineering community. Many advances in the RE community come from software industries. These advances provide new opportunities to systems engineer to make their process more lean and mean. Our approach is designed for early phase requirements engineering. It is not here to replace use-cases or requirements block used in UML/SysML, it is their precursor and complementary technique to them in RE activities.

4.1 Comprehensive Requirements Modeling Language(CReML)

Our approach identifies nine types of early phase RE artifact: *Stakeholders, Goals, Rationale, Viewpoint, Objectives, Constraints, Domain Property, Assumptions and Requirements*, we identify three types early phase relationships : *Contribute, Derive, Conflict*. Requirement artifact diagrams and their semantic meanings is shown in Table 1. They are used to address the problems previously raised in Sections 2 and 3. UML components such as class diagrams and use cases can also used complementarily to enrich CReML models. There are three types of models introduced in CReML: *Goal models*, *Responsibility model* and *Strategy model*. *Goal model* and *Strategy models* are developed simultaneously. Strategy models can be started once the goal modeling begins. *Strategy*

Table 1. Requirement Artifacts definition in CReML

Graphical representation	Semantic meaning
 Goal	*Goals* represents the fundamental state, that the stakeholder would like to achieve by using the system under study. A system can have one primary goal and many other secondary goals. Goals may not be strictly measurable or tangible but stakeholders agree are upon certain conditions for determining the acceptability of goal by system under study.
 Rationale	Rationales are the fundamental reasons, why the goals needs to be achieved by the system under study. Rationales are extracted from stakeholders, and often a single stakeholder may have multiple rationales. These rationales are actually linked to various responsibilities and roles played by a given stakeholder.
 Viewpoint	System viewpoints specify, from the developer's perspective, what characteristics system has to possess and with what magnitude in order to satisfy goals.
 Objective	*Objectives* are the measurable set of tasks and conditions, which the system needs to meet in order to satisfy customer. Goals projected with a specific viewpoint lead to an objective in a direction of particular viewpoint to achieve the goal.
 Constraint	*Constraints* are the limitations imposed on the system by the non-development stakeholders or they may represent some challenges to overcome by the system.
 Domain-Property	A domain-property can defined as a knowledge or information about the domain of the system under study which is uniformly acceptable by all stakeholders: technical or non-technical and which can be verified and validated using scientific methods.
 Assumption	Assumption are hypothesis or non-verifiable information or condition which is considered valid for the system under study. They are close to domain-property but domain-properties can be verified and easily validated.
 Requirement	*Requirements* specify, from the stakeholders' viewpoint, what characteristics it is to possess and with what magnitude in order to achieve the stakeholders' objectives. Requirements are derived from a particular or a set of objectives, constraints, assumption and domain properties.
 Conflict	*Conflict* relationship is used to represent the conflict occurring between the two objectives, or two requirements and hence between the source stakeholders. Conflict means that the implementation of the two requirement artifact cannot be achieved by the system under study at the same time.
 Derive	*Derive* relationships represent the parent-child relationships existing between the various requirements artifacts. A derive relationship exists between the goal and rationales, rationales and viewpoint, viewpoint and objective, objective and requirements, constraints and requirements, domain-property and requirements, assumption and requirements, and between requirements and requirements.
 Contribute	*Contribute* relationship is used to represent the direct contribution of information about requirement artifact from an stakeholder for system under study.
 Stakeholder	*Stakeholders* of the project are the entities which have genuine interest in the project. They are of two types: stakeholders from the client side or end-users and stakeholders responsible for the development of the project. In our terminology, we use Agents also as an stakeholder, as they interact with the system.

model and *goal model* provide inputs to each other over different iterations to better understand the system goals and environment. *Responsibility model* is development can be carried out once the *goal-model* and the *strategy model* are available.

In **first stage** Goal models and strategy models are developed simultaneously. For *Goal modeling* the stakeholders are identified and their goals are extracted out of their corresponding available user stories. Since the stakeholders contribute the goals the relationship between the stakeholders and goal is called *contribute*. These stakeholders are then analyzed and their responsibilities are analyzed taken in account both in presence and absence of the system under study. This analysis leads to various rationales of the corresponding stakeholders role. These rationales of the stakeholders provide the basis of strategy modeling. Rationales provide the inputs regarding how the different stakeholders can be satisfied. Strategy models provide the statements (in form of rationales), which bind together the stakeholders and the statements for their potential conflict. At this stage stakeholders' preference about the various goals are gathered and core and optional goals are identified. This preference gathered over the goals from different stakeholders provides the inputs for the strategy formulations for system development. Stakeholders' preference about traceability of various rationales are also gathered, they are later used to formulate the traceability policies according to the needs of the stakeholders.

In the **second stage**, the various viewpoints are which are of concern to each stakeholders are gathered; a viewpoint is a particular aspect of the system under study which is of interest to stakeholder: client or developer. The analysis of viewpoints lead to the formulation of objectives corresponding to a particular stakeholder. This allows us to understand very clearly the capabilities stakeholder wants to acquire with respect to each viewpoint. At this very stage, potential conflicts among the objectives can be observed, a conflict relationship is marked for further resolution and negotiation for the magnitude of accomplishment of particular objectives.

In the **third stage**, the *responsibility model* is created by separately mapping the stakeholders, viewpoint, objectives, constraints and assumptions. Mapping of objectives with the actors (roles of stakeholders) allows to model the responsibility and point of interactions between the agents (roles of stakeholders) and system. This mapping allows to determine the constraints and assumptions held by the stakeholders and their interrelationship.

In the **Final stage**, the *Goal model* is enriched with the assumptions, and constraints previously extracted from stakeholders. Developers held domain properties are included at this stage to transform them together with objectives into achievable requirements. Each objective may lead to one or more requirement.

Tool Implementation of CReML We implemented a software platform which supports CReML, the platform is called **SysEngLab**: it is composed of three major components: requirements engineering module, decision-engineering module [20], and reliability engineering module. In this paper we are concerned with requirements engineering module and partially with decision-engineering module. The capabilities of requirements engineering module is discussed in this section:

Tagging User Stories with Goals: During the stakeholders requirement elicitation process various techniques are used to elicit the stakeholder requirements. Often, empirical studies have shown that during the first encounter with the clients the needs, desires, and wants are first hand written down in natural languages. The implemented tool allows to create tags for the various user stories based on the goals determined allowing to keep trace of exact origin of a goal in a user story. Often, these requirements represented by various stakeholders also represents the various roles attached to them, which are sometime hidden in the first iteration of the project. The stakeholder identification process first should be carried out to determine all the potential stakeholders with their all potential roles. With each of their roles there are some potential rationales attached. Requirements are actually projection of these rationales in stakeholders' statements often known as user needs or stakeholder requirements. This information which provides links to the stakeholder requirements and various roles is critical for providing the traceability in the later stages.

Integrated Traceability: The problem of traceability lies actually the way it is done. Usually, requirement traceability is carried out when it is demanded by the quality control departments, i.e., when it is solicited. Our proposed tool tracks the links from the very beginning of the goal modeling, every requirement artifact is linked to its parent and child artifact and the root is linked to the user stories. RE module in our tool allows to model the traceability preference of the system, from the very early stage the system tracks which stakeholder has more affinity to which goal and in which viewpoint. As the goal models can be bridged to some of UML/SysML diagrams (Use-case and Block definition diagram) the traceability is continued to the next stage of system design.

Modeling preference In a systems engineering project, it is of great importance that most of stakeholders are satisfied with the various decisions taken during the product development and with the final resulting end product. A higher satisfaction among the stakeholders can be guaranteed if the various stakeholders criteria weights are taken into account in a transparent and holistic manner. Preference modeling if the goals and prioritization of requirements is essential activity, our tools allows to elicit and model both of them over the goal and requirements notations. Unlike goal preference modeling in [14], our approach takes in account the preferences of group of stakeholders and calculates the net preference of each goal using the integrated decision module, and shows it above the goal notation. Similarly, the core and optional features of the system under study can also be marked to keep track of requirements of a product line.

Including Boiler Plates: The requirements diagram used in goal-modeling are equipped with boiler plates mentioned in [3, 21] to help user to write the requirements. The boiler templates aide user to put their capability, capacity, constraint, performance requirements and other type of requirements.

Generation of reports: Our tool supports automatic generation of reports supporting various types of formats. The requirements specification document can be generated in pdf format, user stories can be exported using excel, goal and responsibility models can be generated in image forms, traceability information can be generated in form of matrix with demanded parameters.

5 Application Example using *CReML* and *SysEngLab*

To show the ease of usage of *CReML*, we present here an example of goal modeling using preference and other features previously mentioned implemented in our developed tool SysEngLab, Figure 2 shows an screen-shot with user stories. The current example is based on IBIS project currently under implementation in our research group. The project aims at developing a platform which is capable of demonstrating to show the life-cycle of simulation based systems engineering for an aircraft. We take simplified parts of the original diagram developed for the project to show the various capabilities of *CReML*.

Fig. 1. Client-Stakeholder analysis

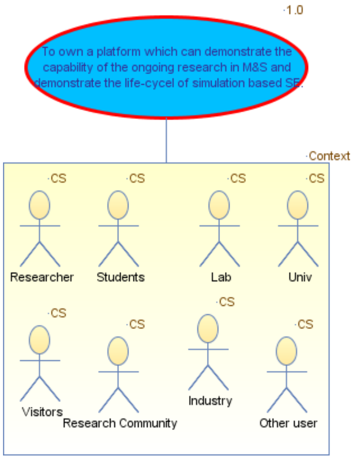


Fig. 2. SysEngLab Screen shot

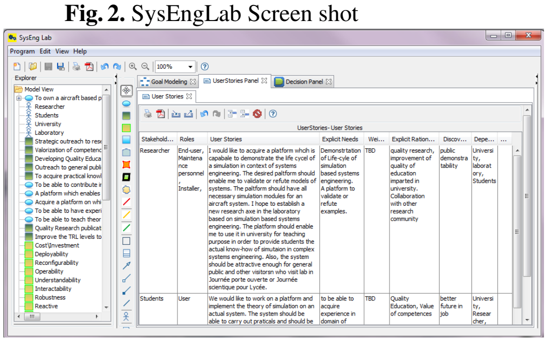


Figure 1, shows the potential client stakeholders of the project. The identified client stakeholders are: Researcher, Students, Laboratory, University, Research Community, Industrial partners, General public and Other users. Notation ‘CS’, DS, and ‘A’ above stakeholders represent the type of stakeholders client-stakeholders, developer stakeholder and actor/agent respectively.

Figure 3, shows the four high priority stakeholders in the goal map, primary goal is derived from the Researcher stakeholder from his user story and the secondary goals are extracted from the other three high priority stakeholder from their user stories. The rationales are the reasons for which they need their goal to be accomplished. The notation above the secondary goals marks their weight in the project, which is decided by the decision makers, *SysEngLab* allows to weight the subgoals using Utility Rank Order Weighting technique (UROW) [20], which is integrated with its decision support module. Figure 4, shows the various rationales and derived viewpoints which are of interest to them. As it is clear that the diagram gets more and more complex. *SysEngLab* allows to export a particular rationale and associated viewpoint in a separate files while keeping trace with the original file. This allows to concentrate on a particular rationale and the related viewpoints. In Figure 5, we take only two viewpoints from the previous diagram

Fig. 3. Goals, subgoals, Rationales analysis

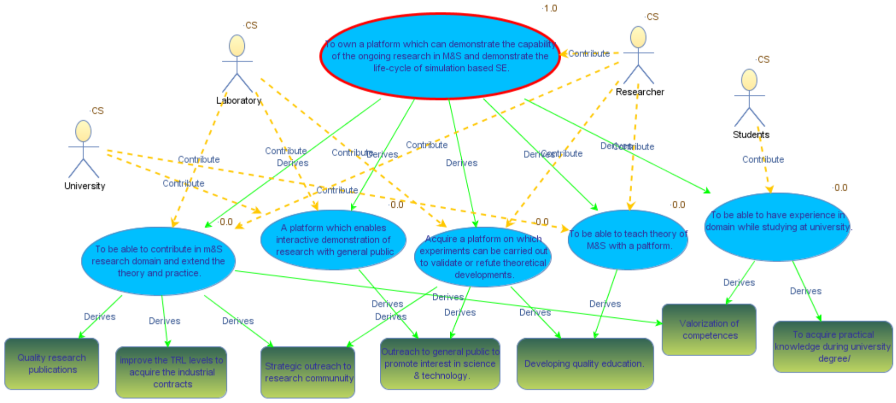
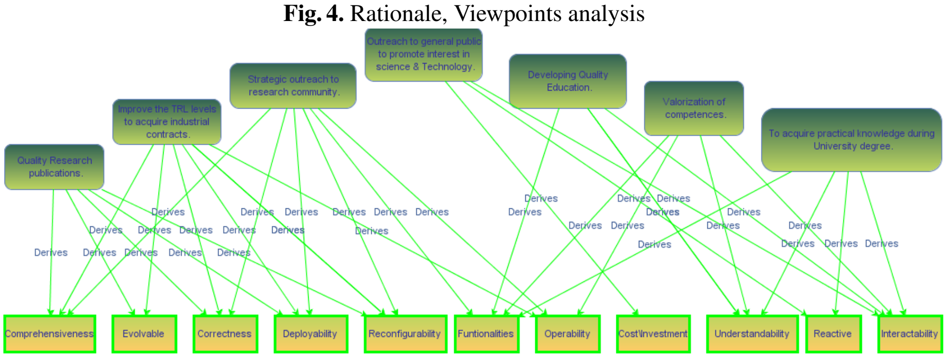
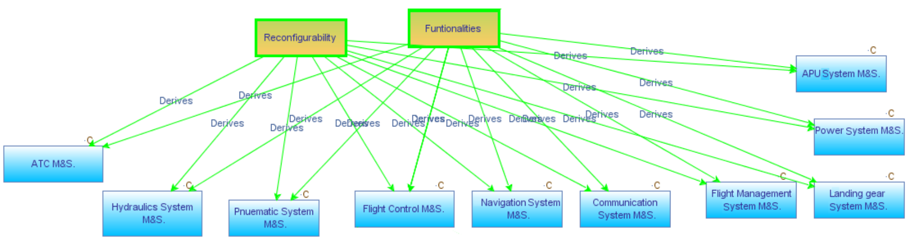


Fig. 4. Rationale, Viewpoints analysis



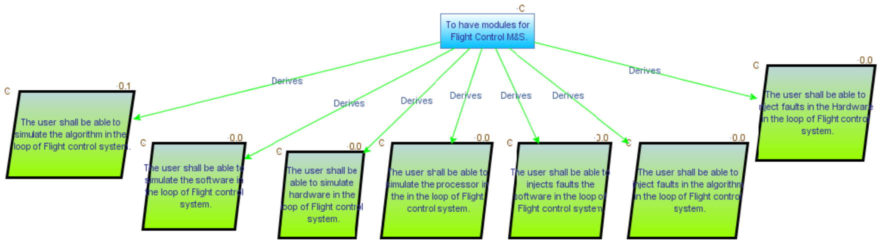
and show which are the objectives derived from the two viewpoints: Reconfigurability viewpoint and Functionality viewpoint. The objectives derived are listed in diagram into various functional and reconfigurable systems. Figure 6, shows the derived requirements

Fig. 5. Viewpoints and objectives analysis



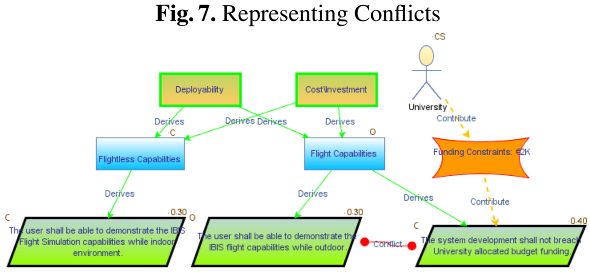
from the objective representing the need for module of flight control system Modeling & Simulation. The notation 'C' above the objective shows that it is core objective and the numbers and notations 'C'or 'O' above the requirements diagram show core requirement or optional requirements and digits show their weight in the objective to be realized. Figure 7, shows how the conflicts can be represented easily between the two requirements.

Fig. 6. System requirements derived from objectives



Conflict relations mark the impossibility of co-existence of two requirements without negotiation.

Fig. 7. Representing Conflicts



6 Conclusion and Future perspectives

We proposed a graphical modeling language which is capable of functionalities typical to popular GORE techniques like i* and KAOS and other functionalities which are of concern to systems engineers and other stakeholders. Proposed language and supporting tool allows requirement engineer to represent the preferences of the various stakeholders on the various goals and objectives. It allows to model both the core and optional features of the system under study. The goals can be traced back to the user stories which are linked to the goal modeling diagram. The responsibility and interaction among the agents is separately modeled and can be integrated if the developer wishes. The other interesting capability our tool provides is to model the rationales using view-points. The stakeholder rationales are projected and divided into various viewpoints from the very early stage, which allows to better understand the user requirements. The end-product of goal modeling leads to system requirements which can be allocated to the UML/SysML diagrams. Our tool supports a few of the diagrams of the UML/SysML notably Use-case diagrams and Block definition diagrams. This is to provide direct traceability throughout the V-cycle. In future we look forward to implement and integrate all the structural and behavior diagrams of UML/SysML in our tool, to make it more comprehensive and useful.

References

1. A.P. Sage and J.E. Armstrong. *Introduction to systems engineering*. Wiley series in systems engineering. Wiley, 2000.
2. J.L. Eveleens and C. Verhoef. The rise and fall of the chaos report figures. *Software, IEEE*, 27(1):30 –36, 2010.
3. E. Hull, K. Jackson, and J. Dick. *Requirements engineering*. Springer London, 2011.
4. Edmond Tonnellier and Olivier Terrien. Rework: models and metrics. In *Complex Systems Design & Management*, pages 119–131. Springer, 2012.
5. A. Van Lamsweerde. Goal-oriented requirements engineering: a guided tour. In *Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on*, pages 249–262, 2001.
6. S. Misra, V. Kumar, and U. Kumar. Goal-oriented or scenario-based requirements engineering technique - what should a practitioner select? In *Electrical and Computer Engineering, 2005. Canadian Conference on*, pages 2288–2292, 2005.
7. Steve Easterbrook, Eric Yu, Jorge Aranda, Yuntian Fan, Jennifer Horkoff, Marcel Leica, and Rifat Abdul Qadir. Do viewpoints lead to better conceptual models? an exploratory case study. In *Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on*, pages 199–208. IEEE, 2005.
8. A. Gross and J. Doerr. What you need is what you get!: The vision of view-based requirements specifications. In *Requirements Engineering Conference (RE), 2012 20th IEEE International*, pages 171–180, 2012.
9. O. Gotel, J. Cleland-Huang, J.H. Hayes, A. Zisman, A. Egyed, P. Grünbacher, A. Dekhtyar, G. Antoniol, and J. Maletic. The grand challenge of traceability (v1. 0). *Software and Systems Traceability*, pages 343–409, 2012.
10. E.S.K. Yu. Towards modelling and reasoning support for early-phase requirements engineering. In *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on*, pages 226–235, 1997.
11. Vera Maria Bejamim Werneck, Antonio de Padua Albuquerque Oliveira, and JCSdP Leite. Comparing gore frameworks: i-star and kaos. In *Workshop em Engenharia de Requisitos (WER 2009), Val Paraiso, Chile*, 2009.
12. G. Regev and A. Wegmann. Where do goals come from: the underlying principles of goal-oriented requirements engineering. In *Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on*, pages 353–362, 2005.
13. I.J. Jureta, A. Borgida, N.A. Ernst, and J. Mylopoulos. Techne: Towards a new generation of requirements modeling languages with goals, preferences, and inconsistency handling. In *Requirements Engineering Conference (RE), 2010 18th IEEE International*, pages 115–124, 2010.

14. S. Liaskos, S.A. McIlraith, S. Sohrabi, and J. Mylopoulos. Integrating preferences into goal models for requirements engineering. In *Requirements Engineering Conference (RE), 2010 18th IEEE International*, pages 135–144, 2010.
15. I.J. Jureta, S. Faulkner, and P. Schobbens. Justifying goal models. In *Requirements Engineering, 14th IEEE International Conference*, pages 119–128, 2006.
16. Jaelson Castro, Manuel Kolp, and John Mylopoulos. Towards requirements-driven information systems engineering: the tropos project. *Information Systems*, 27(6):365 – 389, 2002.
17. Objetiver. http://www.objectiver.com/, June 2013.
18. S. Supakkul and L. Chung. The re-tools: A multi-notational requirements modeling toolkit. In *Requirements Engineering Conference (RE), 2012 20th IEEE International*, pages 333–334, 2012.
19. Martin Glinz. Very lightweight requirements modeling. In *18th IEEE International Requirements Engineering Conference*, pages 385–386, 2010.
20. V. Shukla and G. Auriol. Methodology for determining stakeholders' criteria weights in systems engineering. Poster in CSDM 2013, Paris.
21. Requirements Working Group (RWP). *Guide for Writing Requirements*. INCOSE, 2012.