

REDEPEND: Tool support for i^* modelling in large-scale industrial projects

James Lockerbie & Neil Maiden

Centre for HCI Design, City University London, UK
J.Lockerbie@soi.city.ac.uk, N.A.M.Maiden@city.ac.uk

Abstract. This paper describes our REDEPEND tool for i^* modelling and analysis, and its features designed to make it more usable and useful in industrial projects. We present usability features, pattern-based techniques that generate text requirements statements from i^* models, an extension to i^* means-end links using satisfaction arguments, and an approach to integrate i^* with an existing in-house requirements process.

1 Introduction

Whilst the i^* approach [1] has been developed and applied to case studies for some time, it has not been applied widely in industrial requirements projects. This is despite the undoubted strengths of i^* , which include a simple but formal and stable semantics, a graphical modelling notation that is simple to use, models that are amenable to computational analysis, and applicability in both agent-oriented and goal-oriented requirements methods. We believe this lack of industrial uptake is due to: (i) the lack of robust, useful and usable tools for developing and analysing i^* models; (ii) inadequate semantics to express i^* means-end links; and (iii) poor integration with in-house requirements processes. These points form the basis of one of main research objectives – to deliver a usable and useful i^* modelling tool along with new techniques to integrate the i^* approach successfully in large-scale industrial requirements projects.

The remainder of this paper presents our i^* modelling tool, REDEPEND, and a summary of our research results over the past 6 years.

2 The REDEPEND i^* Modelling Tool

The REDEPEND tool provides systems engineers with i^* modelling and analysis functions, coupled with additional functionality and reliability of Microsoft Visio. It provides a graphical palette from which systems engineers can drag-and-drop i^* concepts to develop Strategic Dependency (SD) and Rationale (SR) models. REDEPEND also provides systems engineers with simple functions to assist modelling, such as providing a verification alert if a model change violates i^* modelling constraints.

We have successfully applied i^* and REDEPEND to model requirements for major air traffic management systems [2, 3, 4]. These projects have provided valuable recommendations from analysts to enable us to improve the usability of REDEPEND, and to extend its features to better integrate i^* with in-house requirements processes. We believe that these new features and approaches are essential to the successful uptake of i^* in industrial projects. The following sections describe these features.

3 Usability Features of REDEPEND

REDEPEND has been developed to contain features that make it more usable when handling larger i^* models. For example, the user can link corresponding actors in related models to facilitate simple navigation and rapid access to different parts of a modelled system according to viewpoints – commonly between SD and SR models. This link feature also provides the user with change synchronization options to enable model consistency, e.g. changing an actor name in one model can be propagated through to another. REDEPEND also provides colour-coding and check features to highlight and shade-out model elements using layers, to partition and mark up models during analysis and review tasks. Some of these features can be seen in Figure 1(a).

4 Productivity Features of REDEPEND

To make REDEPEND more useful to requirements analysts we designed simple patterns – recurring syntactic and semantic structures in i^* – that are applied automatically to any SD model expressed in REDEPEND to generate textual requirement statements. Each pattern defines one or more desired properties (requirements) on the future system that must be satisfied for the SD model dependency to hold for the future system. As such, the SD model, which has been signed off as complete and correct, informs further discovery and specification of requirements statements. The concepts and patterns underlying this approach are described at length in [4].

Figure 1 consists of three panels labeled (a), (b), and (c). Panel (a) shows a screenshot of the REDEPEND software interface with a pull-down menu open. Panel (b) shows a screenshot of an Excel spreadsheet titled 'REDEPEND Candidate Requirements Statements' with columns for 'Type', 'Requirement Statement', and 'Dependency'. Panel (c) shows a screenshot of a Word document titled 'REDEPEND generated VOLERE requirements shells' containing a table with requirement details.

REDEPEND generated VOLERE requirements shells	
Requirement	FR1
Type	Functional
Description	The DHAN shall enable ATC Tower Supervisor to provide Departure strategy.
Rationale	Derived from SD model, which specifies the basic rationale for the modelled dependency.
Source	Dependencies: [D1: DHAN depends on ATC Tower Supervisor for [Departure strategy].
Requirement	IFE1
Type	Lock and Feed
Description	The DHAN shall have an interface to enable ATC Tower Supervisor to provide Departure strategy.

Figure 1 The three stages of requirements generation in REDEPEND

Figure 1 demonstrates how REDEPEND generates requirements from an analyst's perspective: (a) the analyst accesses the requirements generation function from the REDEPEND pull-down menu. (b) REDEPEND delivers the candidate requirements into tailored MS Excel sheets. The analyst can tick and un-tick selected requirement statements prior to generating structured VOLERE [5] shells in MS Word, as depicted in (c). Each requirement in the document is structured using and expressed with a partially complete VOLERE shell specifying: a unique id; the requirement type; the requirement description; a rationale of canned text describing how the requirement was generated; and the source dependency in the SD model.

5 Requirements Process Integration Using REDEPEND

We extended REDEPEND to support the specification of satisfaction arguments from REVEAL [6] to provide additional information to *i** means-end links by associating them with important properties of the problem domain. REDEPEND automatically generates a new satisfaction argument sheet for the selected goal or soft goal as shown in Figure 2a. The selected goal or soft goal is the default end-element, and each element that is a means to the goal or soft goal is a means-element. Means-elements from within the same actor boundary as the end-element are documented in the internal tab, with the external tab displaying means-elements from other actors. Changes made by the analyst to the satisfaction argument sheet and model are propagated automatically to both, thus keeping each model and its arguments consistent. The analyst manually completes each satisfaction argument by selecting an existing property from the database or adding a new one to it. The domain properties are stored in a global database (see Figure 2b) associated with the SR model to ensure effective reuse of properties that, we believe, can improve the specification of satisfaction arguments.

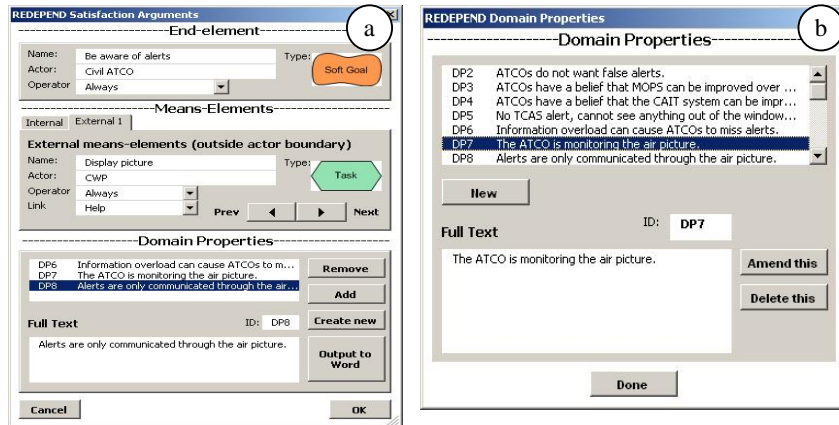


Figure 2 Satisfaction arguments (a) and domain properties management (b) in REDEPEND

We also implemented a procedure to analyse the impact of software requirements on system-wide goals and soft goals using an embedded functional requirement-SR matrix, as shown in Figure 3. An analyst copies functional requirements into the left column, then REDEPEND automatically generates the other columns with tasks and resources from the selected software actor in the SR model. The analyst then completes the matrix by adding a simple + or - to indicate whether the task or resource is enhanced or detracted by the functional requirement. The analyst can then use the tool to propagate this mapping through to goals and soft goals in the model. To aid this task REDEPEND supports 2-way navigation between elements in the SR model and the matrix. We consider such model navigation is essential to support the analysis of large systems. Full details of this work can be found in [3]

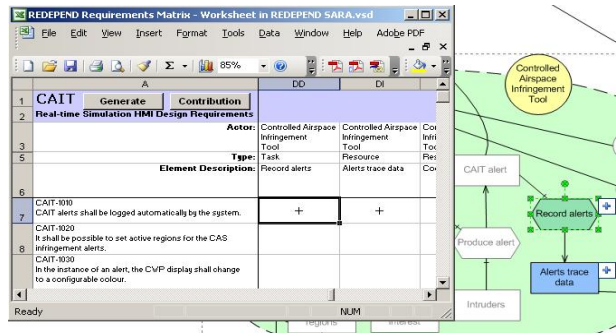


Figure 3 Mapping software requirements from the matrix to i^* model elements in REDEPEND

6 Future Research

We continue our research to develop and refine the use, and hence effectiveness, of REDEPEND through its application in industrial projects. Work is already under way in the following areas:

- Incorporating the SR model in the pattern-based requirements generation facility.
- Automatically generating a first cut SR model from a complete SD model.
- Exploration into the new functionality of Visio 2007, such as improved external data linking and presentation capabilities.
- Collapsing actor boundaries for improved model scalability.

7 References

- 1 Yu E. & Mylopoulos J.M., 1994, 'Understanding "Why" in Software Process Modelling, Analysis and Design', Proceedings, 16th International Conference on Software Engineering, IEEE Computer Society Press, 159-168.
- 2 Maiden N.A.M., Jones S.V., Manning S., Greenwood J. & Renou L., 2004, 'Model-Driven Requirements Engineering: Synchronising Models in an Air Traffic Management Case Study', Proceedings CaiSE'2004, Springer-Verlag LNCS 3084, 368-383
- 3 Maiden N.A.M., Lockerbie J., Randall D., Jones S. & Bush D., 2007, 'Using Satisfaction Arguments to Enhance i^* Modelling of an Air Traffic Management System', Proceedings 15th IEEE International Conference on Requirements Engineering, IEEE Computer Society Press, 49-52.
- 4 Ncube C., Lockerbie J. & Maiden N.A.M., 2007, 'Automatically Generating Requirements from i^* Models: A Case Study with a Complex Airport Operations System', Proceedings 13th International Working Conference, REFSQ'2007, Trondheim Norway, Springer-Verlag Lecture Notes on Computer Science LNCS 4542, 33-47.
- 5 Volere Requirements Resources, <http://www.volere.co.uk/template.htm>
- 6 Hammond J., Rawlings R. & Hall A., 2001, 'Will It Work?', Proceedings 5th IEEE International Symposium on Requirements Engineering, IEEE Computer Society, 102-109.