

# REDEPEND-REACT: an Architecture Analysis Tool

Gemma Grau<sup>1</sup>, Xavier Franch<sup>1</sup>, Neil A.M. Maiden<sup>2</sup>

<sup>1</sup>Universitat Politècnica de Catalunya (UPC) c/ Jordi Girona 1-3, Barcelona E-08034, Spain.

<sup>2</sup>Centre for HCI Design. City University. Northampton Square, London EC1V 0HB, UK.

{ggrau, franch}@lsi.upc.edu, n.a.m.maiden@city.ac.uk

## Abstract

*The  $i^*$  approach includes the construction of Strategic Dependency models (SD), which has been developed to model information systems and are currently applied in a wide variety of contexts. We are interested in the formulation of metrics over  $i^*$  models to be used in components system architectures analysis. These metrics are defined to analyse the models with respect to some properties that are interesting for the system being modelled, such as security, efficiency or accuracy. Metrics are defined in terms of the actors and dependencies of the models and the results of their evaluation are used to inform multiple component selection. As the definition and evaluation of the property metrics are cumbersome activities, the need of tool support arises. Thus, we have enlarged REDEPEND, a Microsoft Visio plug-in that graphically models system goals using the  $i^*$  formalism, to support the definition and evaluation of architectural properties obtaining the REDEPEND-REACT prototype.*

## 1. Introduction to $i^*$ and REDEPEND

The  $i^*$  approach [1] was originally developed to model information systems and, in particular, the dependencies between human and technological actors in software-oriented system architecture by means of Strategic Dependency models (SD). Nowadays, it is currently used in early phases of requirements engineering, organizational analysis and business process reengineering, among others.

Some tools have arisen for constructing  $i^*$  models, among which is REDEPEND, a graphical modelling tool implemented as a Microsoft Visio plug-in that enables to model complex agent-based systems and their dependencies using the  $i^*$  formalism [2]. Complex  $i^*$  SD models are produced by using a Visio stencil that allows to drag and drop the different  $i^*$  shapes (actors, goals, softgoals, resources, tasks and dependencies) into the drawing page, supporting the development and

validation of large  $i^*$  SD and Strategic Rationale models (SR).

## 2. A framework for the definition of metrics for strategic dependency models

The software components market is growing quickly and the functionalities of available software components vary: some of them partially fit the functionality of a unique software domain, while others cover the functionality of more than one. Also, having two or more components with similar functionality in the same system can improve properties such as availability. By taking those aspects into account, several combinations of components are feasible when designing a software system, each of them yielding to an alternative architecture.

In [3] we propose to apply  $i^*$  to model software architectures in terms of actors that represent software domains and actors (human, organisation, hardware or other software) that interact with these domains. This entire interaction is modelled in terms of dependencies among the involved actors. The system modelled with the  $i^*$  SD model is analysed in order to inform non-functional and organizational goals by means of system properties (security, efficiency, accuracy...) that can be evaluated with metrics. Metrics are defined in terms of the actors (actor-based metrics) or the dependencies (dependency-based metrics) of the model and the results of this analysis are used to inform multiple component selection. In [4] we have defined a framework that includes the general form of this kind of metrics over  $i^*$  SD models.

## 3. REDEPEND-REACT

REDEPEND-REACT includes a prototype for the application of the framework over  $i^*$  models defined with REDEPEND. The generation and evaluation of architectures is done by means of several activities, which involve building the  $i^*$  SD model, defining the properties and evaluating the architectures. More details about REDEPEND-REACT can be found at [5].

### 3.1. Properties, actors and components

Properties addressing non-functional aspects such as security, efficiency and so on are likely to appear over and over in system analysis. Thus, it becomes useful to have a complete and versatile catalogue of metrics for such properties in order to allow reusability and return of investment. REDEPEND-REACT supports the definition of the two types of metrics, actor- and dependency-based metrics (see figure 1).

Following the same principle, an actors catalogue and a components catalogue are provided. The catalogue of actors contains different kinds of actors (software, human, organization or hardware), that can be used for defining models by dragging & dropping them to the central page. In the catalogue some generic dependencies for each actor can be defined, making the definition of the model easier. The aim of the components catalogue is to record the available components and state which software domains are covered by each component.

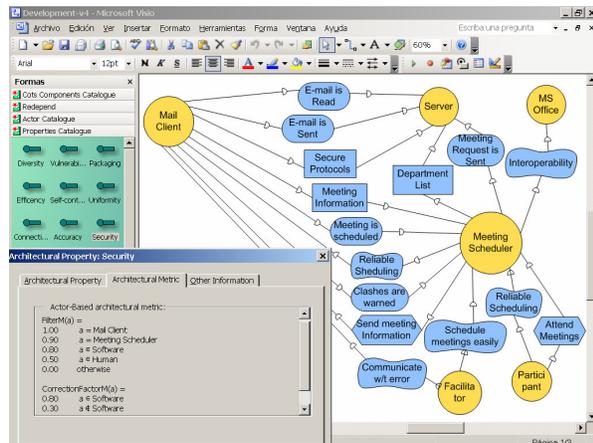


Figure 1. Part of the definition of an Actor-Dependency metric for the Security Property in REDEPEND-REACT.

### 3.2. Evaluation of architectures

REDEPEND-REACT guides the process of generating and evaluating different architectures for the system. Some heuristics guide the process of generating architectures. For instance, an architecture with an actor covered for more than two components is recommended if there are many dependencies between those two actors. Also an architecture with an actor covered for more than two components is not recommended because it is not a usual situation. Only architectures covered with a set of existing components are recommended by the tool. REDEPEND-REACT offers a feature to state a limitation on the coverage of a particular actor, and allows the concept of anchoring: a software actor is anchored if it is covered by a software subsystem whose selection is not negotiable.

Concerning the evaluation, it is necessary first to choose which properties are taken into account and next all the generated architectures are evaluated with respect to those properties. Once the evaluation results are available, the different architectures can be analysed with respect to the values for the properties that are interesting for the system.

### 4. Conclusions

REDEPEND is a tool that enables a user to construct and analyse  $i^*$  SD and SR models that has been used extensively to produce  $i^*$  models in several projects. REDEPEND-REACT is an extension of REDEPEND that assists the component selection process by allowing the definition of several catalogues (actors, software components and properties where basic dependency-based and actor-based metrics can be defined). Candidate architectures are generated by following a certain criteria and providing a strong rationale. Architectures can be evaluated with regard to some properties considered important for the system being modelled. Finally, this analysis is used to assist a multiple software component selection process, always as a complement for other selection techniques.

As a future work the current version will be enlarged with some facilities for tuning the weights of the metrics of the properties and more heuristics for generating architectures. Also some facilities for analyse the evaluation results will be provided. Finally, REDEPEND-REACT will be integrated with some other existing tool that allow the evaluation of components to help, not only in the selection of the most appropriate architecture, but also the selection of the most appropriate software components [6].

### Acknowledgements

This work has been partially supported by the CICYT programme, project TIN2004-07461-C02-01. G. Grau work is supported by an UPC research scholarship.

### 5. References

- [1] E. Yu. *Modelling Strategic Relationships for Process Reengineering*. PhD. thesis, University of Toronto, 1995.
- [2] P. Pavan, N.A.M. Maiden, X. Zhu. "Towards a Systems Engineering Pattern Language: Applying  $i^*$  to Model Requirements-Architecture Patterns". In *Proc. of STRAW'03*.
- [3] X. Franch, N.A.M. Maiden. "Modeling Component Dependencies to Inform their Selection", In *Proc. of ICCBSS'03*, LNCS 2580, Springer-Verlag, 2003.
- [4] X. Franch, G. Grau, C. Quer. "A Framework for the Definition of Metrics for Actor-Dependency Models", In *Proc. of RE'04*.
- [5] <http://www.lsi.upc.edu/~GESSI/REDEPEND-REACT/>
- [6] J.P. Carvallo, X. Franch, G. Grau, C. Quer. "QM: A Tool for Building Software Quality Models". In *Proc. of RE'04*.