J-PR*i*M: A Java Tool for a Process Reengineering *i** Methodology

Gemma Grau, Xavier Franch, Sebastián Ávila
Universitat Politècnica de Catalunya (UPC)
c/ Jordi Girona 1-3, Barcelona E-08034, Spain.
{ggrau, franch}@lsi.upc.edu, d8861226@est.fib.upc.edu

Abstract

The i* approach is a consolidated modelling technique that has proven to be useful in the requirements engineering phases of software development. Using i* requires the adoption of a methodology for defining the models and tool support for manipulating them. For addressing those aspects, we propose J-PRiM, a tool that allows to define i* models by applying PRiM, our Process Reengineering i* Methodology.

1. Introduction

The *i** framework [1] is becoming a consolidated approach both in academia and industry. As we can check in [2], it is broadly applied in fields such as requirements engineering, organizational analysis and business process reengineering, among others. The *i** framework proposes the use of Strategic Dependency models (SD) and Strategic Rational (SR) models for representing socio-technical systems.

One of the strongest points of *i** is its graphic utility and the freedom it provides for defining and using its elements. However, when working over a specific problem or domain, some difficulties arise: there is an overload of variants of the *i** language, a lack of guidelines for producing the models, and tool support is essential. In order to address those aspects, in [3] we presented a reference meta-model to analyse the different variants of the *i** language and, in [4], we proposed PR*i*M, a methodology for guiding *i*models construction. In this paper we introduce J-PR*i*M, a tool for supporting the PR*i*M methodology, which has been developed in JAVA using the Eclipse framework. J-PR*i*M runs with a MySQL database, which design is compliant with the *i** meta-model presented in [3].

2. J-PR*i*M as tool-support for PR*i*M

The PR*i*M methodology [4] addresses *i** modelling from the process reengineering perspective, where the specification of a new system starts from the observation of the current process and ends with the achievement of the specification of the system-to-be.

The PR*i*M methodology is composed of five phases that are decomposed into steps (see table 1). The tool offers different support for each of the steps of the PR*i*M methodology, according to the necessary level of user information. Therefore, steps that require the user to enter data are supported by forms; steps that complete existent data but still require user expertise are guided; and, steps that can generate new data without user interaction are automatically supported.

In the first phase of PR*i*M, the current process is analysed and the information obtained is summarized into Detailed Interaction Scripts (DIS). DIS are tables that describe the information of each activity of the current process by means of its preconditions, postconditions, triggering events, and a list of the actors, actions and resources involved in the activity. J-PR*i*M allows to define the activities of the process and to fill the DIS templates by providing forms.

In the second phase, the construction of the *i** model begins with the identification of the actors of the

	Phases Steps	TS
1	Analysis of the Current Process	
	1.1. Analysis of the Current Process	F
	1.2. Documenting the Current Process	F
2	Construction of the *i** Model	
	2.1. Actor Identification and Modelling	F
	2.2. Building the Operational *i** Model	A
	2.3. Building the Intentional *i** Model	G
	2.4. Checking the Resulting *i** Model	A
3	Generation of Alternatives	
	3.1. Reengineering the current System	G
	3.2. Adding/Removing System Actors	F
	3.3. Reallocating Responsibilities	G/A
	3.4. Checking Consistency between Alternatives	A
4	Evaluation of Alternatives	
	4.1. Choosing Suitable Properties	F
	4.2. Defining Property Metrics	F
	4.3. Evaluating Alternative Models	A
	4.4. Evaluation Trade-off Analysis	G
5	Specification of the New System	N/S

Table 1. PR*i*M phases and steps, showing how each step is supported by J-PR*i*M (F: Forms, G: Guided, A: Automatic).

system, and accordingly, the tool provides the actors introduced in the previous phase as the starting list. The *i** model is built in two differentiated steps in order to distinguish the functionality performed by the stakeholders from their strategic intentionality. PR*i*M provides prescriptive rules for generating the Operational *i** model of the system from the DIS and, as those rules are very specific, the tool generates this model automatically. Based on the operational *i** model and several guidelines, the tool guides the user for obtaining the Intentional *i** model.

In the third phase, process alternatives are systematically generated by means of adding new actors to the system, removing existing actors, and reallocating the responsibilities between the others. The tool provides: 1) automatic generation of all the possible combinations according to some predefined filters; 2) an assistant that helps the user to distribute responsibilities between actors. In both cases, those different alternatives are evaluated in the fourth phase. J-PR*i*M assists the user in the selection of properties and the definition of their metrics, and automatically evaluates the models according to them. Trade-off analysis is provided for support the selection of the most suitable solution. Finally, in the fifth phase, PR*i*M proposes the generation of the new system specification from the *i** model of the chosen alternative. This aspect is not yet supported by the tool.

To guide the application of the methodology, all the phases and steps are ordered in tabs (see figure 1, where phase tabs are at the top, and step tabs are at the bottom grouped by their phase). As PR*i*M is an iterative process that can be refined by going back on its steps, the tool allows changing the data introduced in a previous step and reapply the method. Traceability over these changes is not yet recorded by the tool.

4. Modelling the *i** Framework in J-PR*i*M

J-PR*i*M also supports *i** SD and SR modelling by introducing the elements individually. This can be done either when using the methodology or when developing models from the scratch. The main difference between J-PR*i*M and other modelling tools is in the way the *i** elements are introduced and visualized. Usually *i** modelling tools represent the models as drawings (see [2], tool questionnaire section). However, J-PR*i*M does not show the *i** elements in a graphical way, but in a tree-form hierarchy (see figure 1, left). Consequently, actors, SD dependencies and SR elements can be viewed in different ways (grouped by SR, SD, actor, attributes or steps where they have been generated in PR*i*M). Changes over the *i** model are executed by clicking on the element and choosing the corresponding
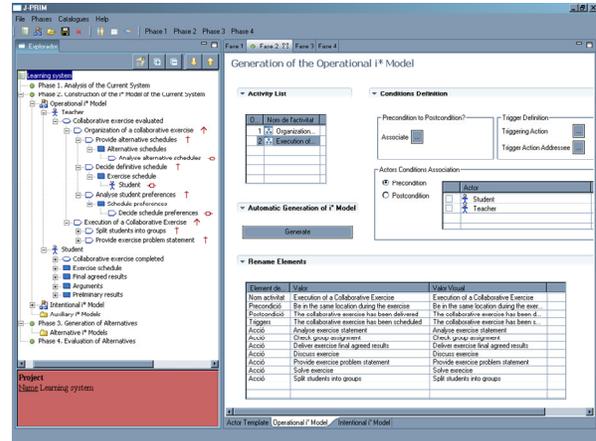

Figure 1. Screenshot of J-PR*i*M.

action on the menu. This approach has been adopted because in our graphical tool REDEPEND-REACT [5] we realized that as *i** models grows, the tool gets slower, models are difficult to manage, individual elements hard to locate, and the model layout has to be rearranged when inserting new elements.

5. Conclusions and Future Work

The intensive use of the *i** framework requires tools and methods to handle the complexity of the resulting models. The PR*i*M methodology facilitates the generation and evaluation of the models by providing concrete guidelines and rules. Using J-PR*i*M, *i** models are easier to define and we can tackle scalability.

As future work we aim at integrating an existing Java *i** tool with J-PR*i*M to allow visualizing the models graphically. We will also add functionality for supporting other methodologies, as well as facilities for transforming the *i** model into a formal specification, as it is proposed in the phase 5 of PR*i*M.

Acknowledgements

6. References

[1] E. Yu. *Modelling Strategic Relationships for Process Reengineering*. PhD. thesis, University of Toronto, 1995.
[2] *i** wiki page: www.istar.rwth-aachen.de. Last visited: May 2006.
[3] C. Ayala *et al.* "A Comparative Analysis of *i**-Based Goal-Oriented Modelling Languages". In *Procs. of* SEKE'05: 43-50.
[4] G. Grau, X. Franch, N.A.M. Maiden. "A Goal Based Round-Trip Method for System Development". In *Procs. of* REFSQ'05: 71-86.
[5] G. Grau, X. Franch, N.A.M. Maiden. "REDEPEND-REACT: an Architecture Analysis Tool". In *Procs. of* RE'05: 455 - 456.