A Metrics Definition Framework for *i**

Xavier Franch

Universitat Politècnica de Catalunya (UPC)
C/Jordi Girona, 1-3, 08034 Barcelona, Spain
franch@lsi.upc.edu

Abstract. Measuring is a central task in the any engineering discipline, and modelling is not an exception. The need for measuring properties over *i** models arises in different contexts. We have analysed this need over *i** models of organizations, projects, systems and architectures. As a result, we have proposed a framework which includes: (a) a metamodel of *i** that defines the elements that build the model; (b) a framework for the definition of metrics; (c) a collection of metrics (currently under construction) of different nature built with the framework. In this work, we present these basic concepts and discuss the applicability of the approach.

1 Introduction

Measuring is a central task in the Information Systems (IS) development process. Some measures are used to evaluate an already built IS, for instance, by establishing its size according to the number of classes or lines of code, or by checking that the resulting system accomplishes its non-functional requirements fit criteria (measuring the response time or failure rate, among others). However, measures can also be taken at the early stages of the IS development process, where they allow predicting some of the quality factors of the system-to-be, and planning corrective actions if needed. In this case, metrics are mainly defined over IS models of different kind. Having good suites of metrics allow not only analysing the intrinsic quality of an individual model, but also comparing different alternative models with respect to some properties in order to select the most appropriate alternative.

Among these models, we find the *i** framework. One of its strengths is versatility: it is currently used in different disciplines such as requirements engineering, business process modelling and reengineering, organizational modelling, or architecture representation, among others. In each of these contexts, an *i** model will be analysed with respect to different properties. The use of metrics may help to conduct these analyses. Therefore, the need for formulating metrics accurately and efficiently arises. In this paper, we report the work of the GESSI group in this subject, presenting a framework for the formulation of such metrics and some possible fields of application.

2 Objectives of the Research

Our work has the following objectives:

- To formulate a metamodel for *i** models (both SD and SR) allowing thus rigorous definition of the metrics on top of the structural form of those models. Our goal has been proposing a metamodel versatile enough to be adaptable and extendible to the different uses and variations of the framework that we may find in the literature. It has to be prepared also to enclose information that may useful for some metrics (e.g., time to perform some task), in addition to the structure itself.
- To define a framework for formulating metrics over *i** models. The framework is intended to provide some general forms of metrics such that defining a new metric basically means instantiating one or more of these forms to obtain the final definition.
- To build a catalogue of metrics using the framework. We basically address two types of metrics: general-purpose, or context-dependant. For context-dependant, we distinguish among domain-specific (e.g., for agent-based systems, databases, etc.) or activity-specific (e.g., for architecture modelling, project management, etc.).
- To propose and show different uses of the framework. We may use the metrics to compare solutions when exploring different alternatives to a problem, to analyse a model of a given system, etc.

3 Scientific Contributions

In [1] we formulated a metamodel for *i** capturing the specificities of the seminal proposal, GRL, and TROPOS variations, and we proposed refactoring as a technique to adapt the metamodel to a particular use. Remarkably, the metamodel includes all the concepts needed to build SD and SR models. Being a UML metamodel, the addition of information as mentioned in the first item of section 2, means just to add some generic attributes to the appropriate classes.

The framework was first outlined in [2] and refined in [3]. Remarkably, we distinguished 2 different axis to formulate metrics. The returned value axis for defining if a metrics is used for checking compliance (logical metrics), for measuring some concept (numerical metrics) or to obtain one or more model elements that fulfil some condition (model-element metrics). The subject of measure axis establishes which kind of model element is measured: we can measure the whole model (global metrics), individual elements (local elements) or even groups of individual elements (group metrics), e.g. pairs of actors). The same property may be measured with different metrics, therefore for strategic importance, we may have a numerical metrics to measure the strategic importance of each actor, a logical metrics to check if actors are over a given threshold, and a model-element metrics to obtain the most strategic actor. In [3], we also proposed the OCL as the language to formulate metrics over the metamodel.

We have formulated an initial (but yet incomplete) catalogue of metrics. In [3] we provide a comprehensive example of general-purpose, non-trivial metrics, predictability. In other works, we have addressed specific domains. In [4], we explored reengineering of software architectures over a documented case study (Home Service Robot) and for this purpose we defined over the *i** framework two classical metrics, coupling and cohesion. In [5], we targeted reengineering of software processes and we focused mainly on defining the functional size of a software system in the COSMIC-FFP framework (using then cfsu, COSMIC functional size unit, as metrics); we also included some results about process agility and ease of communication in the considered organizational alternatives. Other metrics are not currently available in the form of publications.

Concerning uses of the framework, our first interest was in comparing different requirements-oriented alternatives of COTS-based architectures [6]. Afterwards, as mentioned above, we have used *i**-based metrics as an important conceptual tool when comparing alternatives in reengineering systems. We have formulated a customizable reengineering framework [7] that has been customized both for software process reengineering [8] and architecture reengineering [4].

4 Conclusions

The use of metrics is very common in different type of models. For instance, there are some suites of metrics in the field of object-oriented modeling, which refer to structural properties like cohesion and coupling. Properties referring to the system itself, such as security, efficiency or cost, which mainly fall into the category of non-functional or organizational requirements, appear when considering models of the system architecture. For this reason, having metrics defined over *i** models is not surprising. In this paper, we have assumed this fact and then proposed a framework for formulating metrics over *i** models. The most significant contributions of the proposal are:

- Accuracy. We have provided a UML metamodel that is used as a baseline upon which the framework is built. Metrics are expressed with the OCL. We also have defined a method, R*i*SD, to build models in a systematic way [9], which also helps to get more accurate results.
- Expressiveness. The use of the OCL allows expressing metrics both in a comfortable and expressive way (although OCL is sometimes a bit messy).
- Sensitivity. Metrics can be defined more or less accurately depending on: 1) the expert judgement available; 2) the state of refinement of the model; 3) the effort we want to invest in model analysis.
- Easy tool support. The form that our framework takes allows implementation of tool support to drive metrics definition, model edition, generation of alternatives and evaluation of models. Our J-PR*i*M tool [10] is a first running prototype.
- Reusability. The metrics obtained may be reused in different projects of the same kind.

5 Ongoing and future work

As ongoing work, we mention: (a) Completing the catalogue with new, validated metrics constructed with them; (b) Incorporating the catalogue into J-PR*i*M going further than the current prototype.

Acknowledgements

This work has been supported by the Spanish research project TIN2007-64753. A significant part of the metrics framework and its application to reengineering activities has been developed jointly with Gemma Grau as part of her PhD thesis. The work related to the the metamodel formulation has been made jointly with other members of the GESSI group at the UPC.

References

1. Ayala C., Cares C., Carvallo J.P., Grau G., Haya M., Salazar G., Franch X., Mayol E. and Quer C.: A Comparative Analysis of *i**-Based Agent-Oriented Modeling Languages. *Proceedings of the 17th Conference on Software Engineering and Knowledge Engineering* (SEKE'05) (2005) 43-50
2. Franch X., Grau G., Quer C.: A Framework for the Definition of Metrics for Actor-Dependency Models. *Proceedings of the 12th IEEE International Requirements Engineering Conference* (RE'04) (2004)
3. Franch X.: On the Quantitative Analysis of Agent-Oriented Models. *Proceedings of the 18th International Conference on Advanced Information Systems Engineering* (CAiSE'06), LNCS 4001 (2006)
4. Grau G., Franch X.: A Goal-Oriented Approach for the Generation and Evaluation of Alternative Architectures. *Proceedings of the 1st European Conference on Software Architecture* (ECSA'07), LNCS 4758 (2007)
5. Grau G., Franch X.: Using the PR*i*M method to Evaluate Requirements Models with COSMIC-FFP. *Proceedings of the IWSM-Mensura 2007 Conference*
6. Franch X., Maiden N.: Modelling Component Dependencies to Inform Their Selection. *Proceedings of the 2nd International Conference on COTS-Based Software Systems* (ICCBSS'03), LNCS 2580, 2003.
7. Grau G., Franch X.: ReeF: Defining a Customizable Reengineering Framework. *Proceedings of the 17th International Conference on Advanced Information Systems Engineering* (CAiSE'07), LNCS 4495 (2007)
8. Grau G., Franch X., Maiden N.: PRiM: an *i**-based process reengineering method for information systems specification. *Information and Software Technology* (IST), 50(2), Jan. 2008.
9. Franch X., Grau G., Mayol E., Quer C., Ayala C., Cares C., Navarrete F., Haya M., Botella P.: Systematic Construction of *i** Strategic Dependency Models for Socio-technical Systems. *International Journal of Software Engineering and Knowledge Engineering* (IJSEKE), Vol. 17, N. 1, January 2007.
10. Grau G., Franch X., Ávila A.: J-PRiM: A Java Tool for a Process Reengineering i* Methodology. *Proceedings of the 14th IEEE International Requirements Engineering Conference* (RE'06) (2006)