An Empirical Evaluation Roadmap for iStar 2.0

Lidia Lopez[1], Fatma Başak Aydemir[2], Fabiano Dalpiaz[2], Jennifer Horkoff[3]

[1] Universitat Politècnica de Catalunya, Barcelona, Spain, llopez@essi.upc.edu,
[2] Utrecht University, Utrecht, Netherlands, {f.b.aydemir, f.dalpiaz}@uu.nl
[3] City University London, UK, horkoff@city.ac.uk

Abstract. The iStar 2.0 modeling language is the result of a two-year long community effort intended at providing a solid, unified basis for teaching and conducting research with *i**. The language was released with important qualities in mind, such as keeping a core set of primitives, providing a clear meaning for those primitives, and flattening the learning curve for new users. In this paper, we propose a list of qualities against which we intend iStar 2.0 to be evaluated. Furthermore, we describe and empirical evaluation plan, which we devise in order to assess the extent to which the language meets identified qualities and to inform the development of further versions of the language. Besides explaining the objectives and steps of our planned empirical studies, we make a call for involving the research community in our endeavor.

Keywords: *i** Framework, iStar 2.0, empirical engineering, evaluation

1 Introduction

Many dialects and extensions of the *i** modelling language have been proposed since its introduction in the 1990s. Although these proposals demonstrate the popularity of the language (at least in the research community) and allowed adaptation of the framework to a variety of domains (e.g., security, law, service-oriented architectures), they have also created difficulties in learning, teaching, and applying *i** consistently.

iStar 2.0 [1] is the result of a collective effort of the *i** community aimed to overcome these difficulties by defining a standard core set of concepts. Given the objectives of iStar 2.0, our aim is to measure how well the language achieves them, also to inform further developments of the standard on the basis of empirical evidence.

More specifically, our research question is the following: *Does iStar 2.0 provide a solid and unified basis for teaching and continuing with research on goal-oriented requirements engineering?* Towards answering this question, we identify several relevant qualities and provide an initial roadmap for the empirical studies to conduct to evaluate iStar 2.0 against those qualities.

The remainder of the paper is structured as follows. Section 2 includes a brief literature review of empirical evaluations in modelling languages and *i**. In Section 3, we define the set of qualities to be empirically evaluated and a tentative roadmap defining the timeline of the implementation of these evaluations. Finally, we draw some conclusions in Section 4.

2 Empirical Evaluation of Modeling Languages

There is a variety of empirical evaluations in the area of modelling languages in general, and in *i** modelling language in particular. This section provides a brief summary of these studies focusing in the qualities evaluated by the studies.

There are several works in the literature on the evaluation of modelling languages. Among them, Lindland et. al [2] propose a framework that defines quality goals and the means for achieving these for conceptual modelling in order to distinguish between what to achieve and how to achieve it. They identify three qualities related to modeling languages: syntactic, semantic, and pragmatic. Semantic qualities refer to the validity and completeness of the language and the models generated using the language, syntactic qualities are related to the syntax of the language, and pragmatic qualities concern the understandability of the language and its application.

Guizzardi et al. [3] suggest domain appropriateness and comprehensibility appropriateness as key qualities of a modelling language. They rely on verifying properties of model instances: lucidity, soundness, laconicity, and completeness. These model properties are then related to corresponding language properties: construct overload, construct excess, construct redundancy, and ontological expressiveness.

Frank [4] proposes a method to evaluate reference models, where the evaluation not only concerns the general qualities of conceptual models but also re-usability of the reference domain. The framework states four different evaluation perspectives: economic, deployment, engineering and epistemological. Each perspective is structured into multiple aspects and for each aspect a success criterion is provided.

Interest in *i** evaluation appears to be on the rise, with studies covering both the language evaluation and the applicability of *i** in the industry. We distinguish between different kinds of studies. Some works evaluate the use of an *i** extension comparing it to the use of *i** [5]. Other approaches compare *i** with other goal-oriented modelling languages such as KAOS [6] or Techne [7]. Finally, other studies evaluate specific characteristics of the language such as visual effectiveness [8].

The majority of the studies providing empirical evidence in the literature are evaluating the applicability of *i** for different purposes in the industrial environment. Elahi et al. [9] studied the use of *i** for gathering and understanding knowledge in an organization, concluding that some constructs are not used by practitioners. Carvallo et al. [10] focus on socio-technical systems and conclude that some models result too difficult to read and modify due to their complexity. A variety of real use cases were presented at the *i** Showcase in 2011[1].

3 iStar 2.0 Evaluation Roadmap

In order to evaluate iStar 2.0, we need to define the set of the language qualities that we want to assess. Based on the review of Section 2, we present a number of quali-

[1] http://www.city.ac.uk/centre-for-human-computer-interaction-design/istar11

ties to evaluate, then discuss suitable empirical methods, and finally devise an initial roadmap for the empirical evaluation.

3.1 Qualities to be evaluated

As iStar 2.0 was not defined as a new language, but a set of core concepts refining the original *i** [11], backwards compatibility is critically important. As a community, we need to collect evidence to determine if iStar 2.0 meets the needs of the users of *i**. The open nature of *i** comes with a drawback that iStar 2.0 is trying to mitigate: the steep learning curve that makes it hard to employ the language in the industry. Therefore, learnability is also a priority quality to be evaluated. Keeping the open nature of *i** was also one of the main objectives during the definition of iStar 2.0. Consequently, we also need to consider the *extensibility* quality, i.e. evaluating whether iStar 2.0 is a suitable baseline for extensions.

Additionally to these qualities, we consider some qualities to evaluate the quality of the language, for example *expressiveness* or *syntactic correctness*. Regarding the expressiveness, we are interested in evaluating if iStar 2.0 has a suitable set of constructs (missing, excess or overload). Syntactic correctness evaluates if using iStar 2.0 the modelers can easily detect, correct and even prevent syntactic errors.

We have also included qualities not directly assessed during the definition of iStar 2.0, such as *scalability*. The detailed set of qualities to be evaluated is included in Table 1. We categorize the qualities based on the classification provided in [2].

Table 1. iStar 2.0 qualities to be evaluated

Category	*Quality*	*Definition*
Syntactic	Syntactic correctness	Does iStar 2.0 facilitate ensure and maintain syntactic correctness?
Semantic	Expressiveness	Does iStar 2.0 allow one to capture a sufficient number of concepts in a socio-technical domain?
Semantic	Unambiguous models	Do iStar 2.0 models have only one interpretation?
Pragmatic	Backwards compatibility	Is iStar 2.0 able to represent the same phenomena as *i**?
Pragmatic	Comprehensibility	Can iStar 2.0 models be understood?
Pragmatic	Cost-Effectiveness	Is the effort required to use iStar 2.0 worth the benefits?
Pragmatic	Extensibility	Is it easy to add new concepts to iStar 2.0?
Pragmatic	Learnability	How does the learning curve of iStar 2.0 look like?
Pragmatic	Modifiability	Does iStar 2.0 facilitate changing and updating models?
Pragmatic	Practical applicability	Can iStar 2.0 be successfully applied to real world cases?
Pragmatic	Scalable	Does iStar 2.0 support the creation and analysis of large problems?

3.2 Empirical Methods: Design Dimensions

In order to evaluate the qualities listed in Table 1, several empirical studies must be designed and conducted. We envision the application of several empirical methods, including experiments, surveys and case studies. We can enumerate a number of dimensions that must be considered when designing such studies.

Choice of subjects participating in the studies is a dimension that must be determined for each study. To classify the subjects, we can use two categories: expertise and background (industry or academy). We need to clearly define a set of *i** experts for inclusion in the backwards compatibility evaluation. For practical applicability, we need to involve practitioners from industry. For other qualities, we can treat the expertise and the background of participants as a variable in the study.

We also need to decide when to evaluate the iStar 2.0 language in isolation and when a comparative analysis comparing iStar 2.0 to *i** is needed. The same reasons that lead us to pay special attention to the backwards compatibility and the learnability lead us to think, that for these specific qualities, we should conduct comparative analysis. Meanwhile, the evaluation of the other qualities can focus only in iStar 2.0.

3.3 Tentative Roadmap

From an empirical software engineering standpoint, we can identify two main phases for the evaluation of iStar 2.0: formative and summative. The formative phase corresponds to the task related to development of the proposal providing some partial empirical validation for the resulting proposal, while the summative phase evaluates if the proposal can be implemented in the real world. We are currently in the formative phase, and precisely in the treatment validation step of Wieringa's design science methodology [12].

We divide the proposed empirical evaluation plan in three phases, divided in a total of five stages. The first two phases correspond to the formative and summative phases in empirical research, while the third one describes side activities:

- In the formative phase, the evaluation will concern the qualities that led the design decisions for iStar 2.0. These qualities include keeping a core set of primitives (stage 1), providing a clear meaning for such primitives (stage 2), and flattening the learning curve for new users (stage 3).
- In the summative phase, the proposal (in our case, iStar 2.0) should be tested for applicability in real cases (stage 4).
- The third phase includes the study of additional properties that do not directly relate to the use of iStar 2.0 as it is, but rather on its capability to be adapted for specific cases or domains (stage 5).

Figure 1 shows the three phases, including the qualities to be evaluated in each stage. *Cost-effectiveness* is a quality that should be evaluated as part of all the stages. The cost can be evaluated in terms of time in all the stages, and in terms of money in stage 4. Note that stages 1 to 3 and 5 could be executed in any order while stage 4 should be executed after stages 1 to 3 have been conducted.

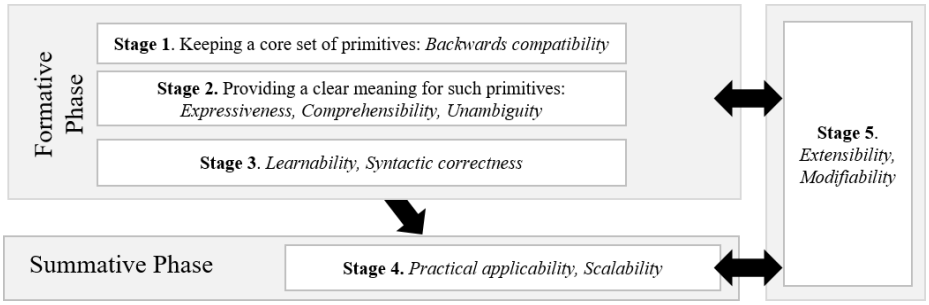

Figure 1: iStar 2.0 Evaluation Roadmap

4 Conclusions

During the last couple of years, the *i** community has been working on the definition of a standard, core version that is called iStar 2.0. The main goal of this effort was to facilitate the learning, teaching, and applying *i** consistently. After the definition of iStar 2.0, the natural next step is evaluating the resulting proposal to provide evidence of whether or not the proposal achieves the expected qualities.

In this paper, we emphasize the necessity of evaluating iStar 2.0 through empirical studies. Our first step is the identification of a set of qualities against which we plan to evaluate iStar 2.0. We also discuss some key dimensions that need to be defined when conducting these empirical studies. Interestingly, many of these qualities we identified are *pragmatic*; we surmise this is linked to the limited adoption of *i** in industry.

We prioritise the evaluation tasks of the qualities grouping them in five stages. Some of these tasks are labelled as formative evaluation, others are part of summative evaluation, and the remaining ones are additional studies on the extensibility and customizability of iStar 2.0. Based on this grouping, we define a tentative roadmap proposing an order of execution for the various evaluation stages.

The next steps consist of conducting empirical studies addressing one or more of the identified qualities for iStar 2.0. Although we plan to design and conduct several studies ourselves, an effective evaluation of the language will require a community-wide effort. We encourage *i** community members to use and evaluate iStar 2.0, keeping in mind the qualities presented here, and reporting the results publicly. Our hope is that, as a community, we build evidence either to support the usefulness of iStar 2.0 as well as to shape the future versions of the language.

Acknowledgments.

This work is supported by EOSSAC project, founded by the Ministry of Economy and Competitiveness of the Spanish government (TIN2013-44641-P), an ERC Marie Skodowska-Curie Intra European Fellowship (PIEF-GA-2013-627489) and a Natural Sciences and Engineering Research Council of Canada Postdoctoral Fellowship (Sept.

2014 - Aug. 2016). The second and third author have received funding from the SESAR Joint Undertaking under grant agreement No. 699306 under European Union's Horizon 2020 research and innovation programme.

References

1. Dalpiaz, F., Franch, X., Horkoff, J.: iStar 2.0 Language Guide. CoRR. abs/1605.0, (2016).
2. Lindland, O.I., Sindre, G., Sølvberg, A.: Understanding Quality in Conceptual Modeling. IEEE Software. 11, 42–49 (1994).
3. Guizzardi, G., Pires, L.F., Van Sinderen, M.: An ontology-based approach for evaluating the domain appropriateness and comprehensibility appropriateness of modeling languages. Proc. of MoDELS. pp. 691–705 (2005).
4. Frank, U.: Evaluation of Reference Models. Reference Modeling for Business Systems Analysis. pp. 118–140 (2006).
5. Teruel, M.A., Navarro, E., Lopez-Jaquero, V., Montero, F., Jaen, J., Gonzalez, P.: Analyzing the understandability of Requirements Engineering languages for CSCW systems: A family of experiments. Information and Software Technology. 54, 1215–1228 (2012).
6. Matulevičius, R., Heymans, P.: Comparing Goal Modelling Languages: An Experiment. Requirements Engineering: Foundation for Software Quality. pp. 18–32 (2007).
7. Horkoff, J., Aydemir, F.B., Li, F.-L., Li, T., Mylopoulos, J.: Evaluating Modeling Languages: An Example from the Requirements Domain. Conceptual Modeling. 260–274 (2014).
8. Moody, D.L., Heymans, P., Matulevicius, R.: Improving the effectiveness of visual representations in requirements engineering: An evaluation of *i** visual syntax. Proceedings of the IEEE International Conference on Requirements Engineering. pp. 171–180 (2009).
9. Elahi, G.., Yu, E.., Annosi, M.C..: Modeling Knowledge transfer in a software maintenance organization - An experience report and critical analysis. Proc. of POEM. LNBIP, 15–29 (2008).
10. Carvallo, J.P. and Franch, X.: On the use of *i** for Architecting Hybrid Systems: A Method and an Evaluation Report. Proc. of POEM. LNBIP, 38-53 (2012).
11. Yu, E.S.-K.: Modelling strategic relationships for process reengineering. Ph.D. Thesis, University of Toronto (1996).
12. Wieringa, R.: Design Science Methodology for Information Systems and Software Engineering. (2014).