Tool Interoperability using iStarML

Carlos Cares[1,2], Xavier Franch[2], Daniel Colomer[2], Lidia López[2]

[1] Universidad de La Frontera, Av. Francisco Salazar 01145, 4811230, Temuco, Chile,
[2] Universitat Politècnica de Catalunya, c/ Jordi Girona 1-3, 08034, Barcelona, Spain,
{ccares, franch, dcolomer}@essi.upc.edu, llopez@lsi.upc.edu
http://www.essi.upc.edu/~gessi/

Abstract. iStarML is an XML-based format for enabling interoperability among *i** tools. Its main design focus was to support data interchange even when involved tools implement different *i** variants. In this paper, we present a summary of the format, we briefly describe the `ccistarml` Java library, and we show an application of it. We finally summarize the requirements for representing new *i** concepts in order to generate a revised version of iStarML.

Keywords: *i** Framework, iStar, iStarML, interoperability.

1 iStarML 1.0: Basics and Structure

As an effect of the past and even current proliferation of different *i** variants, interoperability has become a non-functional requirement hard to accomplish by *i** tools. iStarML [1] is an XML-based proposal that has been conceived to deal with the existence of different *i** variants. It follows a concentric ring structure (see Fig. 1) having a rigid centre and a flexible periphery. Core *i** concepts are in the rigid part of the internal ring whilst flexibility is added going to the periphery up to 4 rings: actor and intentional elements are core concepts; types of core concepts (e.g., softgoal, belief) are in the second ring; particular values for decompositions are in third ring; strong variations of core concepts, e.g. "norm" as intentional element is in the fourth ring.

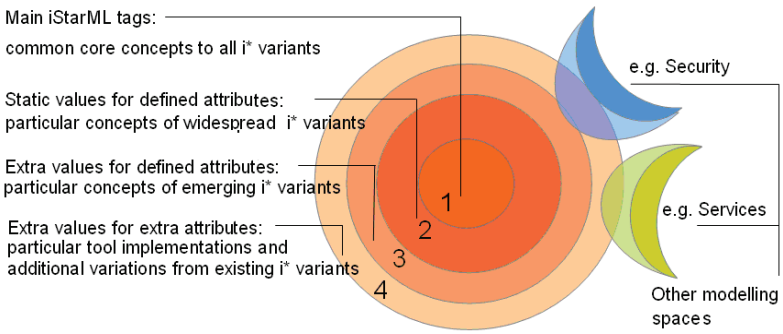

Fig. 1. Concentric ring structure of iStarML.

2 iStarML Computational Support

In [2] we offer a set of basic XML specifications. First, we provide an iStarML specification using an XSD definition. This specification is the most used specification types for XML files. A clear advantage is that XML processors can parse XML files using XSD definitions which means that iStarML parsers can be implemented taking public XML processors and this XSD definition. However, this definition has some limitations derived from the XSD grammar, e.g., a dependency between attributes cannot be represented on XSD specifications. Therefore, although less popular but more accurate, we have provided a Schematron [3] set of rules for iStarML syntax verification.

In addition to these parsing proposals, we have developed a Java package for handling iStarML files. We have called it `ccistarml` package. The several classes inside this package allow creating, reading and modifying iStarML files. Each of these classes is explained and illustrated using simple examples in the `ccistarml` tutorial [4]. The code in Fig. 2 illustrates how an iStarML file can be loaded, parsed, modified by adding a new actor, and saved.

```
ccistarmlFile f = new ccistarmlFile();
f.loadFile("sample01.istarml.xml");
f.istarmlParser();
if (!f.hasErrors()) {
   ccistarmlContent content = f.mainTagStructure();
   content.add_actor("Tutor");
   f.saveFile();
}
```

Fig. 2. Sample code for managing iStarML representations of *i** diagrams.

3 iStarML Tests and Scenarios of Use

Since iStarML was proposed, we have developed different interoperability scenarios and applications: (1) storing diagrams in the H*i*ME hierarchical *i** tool [5]; (2) exporting and importing iStarML in the jUCMNav GRL tool including interoperability proofs [6]; (3) translating files from OME3 tool to iStarML file by using a Java applet [7]. About taking advantage of XML representation of iStarML we have (4) exemplified the use of XPath for metric calculations [1]; (5) used XSLT for transforming iStarML files into Prolog clauses [8]; and (6) used XQuery for quality assurance of *i** models [9]. We also mention the use of iStarML as the basis for formulating a supermetamodel coordinating model interoperability in a semantic-aware scenario [10].

4 Conclusions and Future Work

In this paper we have presented iStarML 1.0 as an interoperability facility for *i** tools. Our proposal considers the existence of *i** variants and hence, a polysemantic scenario. We have shown its structure, some computational support and different

scenarios illustrating both the feasibility of reaching interoperability and the advantages of having a XML representation for *i** models.

It is our position that iStarML is a solid proposal, but its dissemination and usage depends on including most of the current and future representational requirements of the *i** community, which can be reached only by constituting a wide working group motivated to generate iStarML 2.0.

Next steps on iStarML include interoperability tests of the two existing possibilities of graphic representations: iStarML's graphic elements and the nested structures proposed in iStarML for including SVG tags [11]. Here for example we have the challenge of handling the graphical proposal for representing inherited intentional elements from [12]. Finally, we are also planning to improve the current representational capabilities of iStarML extending it to handle concepts such as modules [13] and similar tags enabling Computer Supporting Collaborative Engineering.

Acknowledgments

This work has been supported by the Spanish project TIN2010-19130-c02-01.

References

1. Cares, C., Franch, X., Perini, A., Susi, A.: Towards Interoperability of *i** Models Using iStarML. *Computer Standards & Interfaces*, 33(1), 69-79 (2011)
2. iStarML site, http://www.essi.upc.edu/~gessi/iStarML
3. Jelliffe,R.: The Schematron Assertion Language 1.6, http://xml.ascc.net/resource/schematron/Schematron2000.html (2002)
4. Cares, C.: ccistarml: A Java Package for Handling iStarML files, http://www.essi.upc.edu/~ccares/papers/ccistarml_v0.6.pdf
5. HiME site, http://www.upc.edu/gessi/HIME/
6. Colomer, D., López, L., Cares, C., Franch, X.: Model Interchange and Tool Interoperability in the *i** Framework: A Proof of Concept. In *Proc. of the 14th Workshop on Requirements Engineering* (WER11), 27-29 April, Rio de Janeiro, Brasil, pp.369-381 (2011)
7. OME3 to iStarML Converter, http://www.essi.upc.edu/~gessi/iStarML/ometoistarml-remoto/online_tools.html
8. Cares, C., Franch, X.: 3MSF: A Framework to Select Mobile Office Devices. *Int. Journal of Computer Science and Applications*, 6(5), 121-144 (2009)
9. Cares, C., Franch, X.: Towards a Framework for Improving Goal-Oriented Requirements Models Quality. In *Proc. of the 12th Workshop on Requirements Engineering* (WER09), July 16-17, Valparaiso, Chile. pp. 3-14 (2009)
10. Cares, C. and Franch, X.: A Metamodelling Approach for i* Model Translations. In Proc. of *the 23rd Int. Conference on Advanced Information Systems Engineering* (CAiSE), June 20-24, London, United Kingdom, pp. 337-351 (2011).
11. Scalable Vector Graphics (SVG) site, http://www.w3.org/Graphics/SVG/
12. López, L., Franch, X., Marco, J.: Defining Inheritance in i* at the Level of SR Intentional Elements. In *Proc. of the 3rd International i* Workshop* (iSTAR08), February 11-12, Recife, Brazil, pp. 71-74 (2008).
13. Franch, X.: Incorporating Modules into the i* Framework, LNCS, 6051, pp 439-545 (2007)