Towards iStarML 2.0: Closing Gaps from Evolved Requirements

Carlos Cares[1], Lidia López[2]

1 Universidad de La Frontera, Temuco, Chile, carlos.cares@ceisufro.cl
2 Universitat Politècnica de Catalunya, Barcelona, Spain, llopez@essi.upc.edu

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 analyse required changes to the format from two main sources (i) the evolution of *i** into a consistent and clear set of core concepts expressed in the new iStar 2.0 specification and (ii) recurrent necessities due to a wide use of *i** modelling. In order to address these requirements, we propose new XML elements to be considered in a new version of iStarML: iStarML2.0

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

1 Introduction

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 following Lotman's semiosphere theory of human communication [2]. 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 are in the second ring, for example goal, softgoal, task; particular values for decompositions are in third ring; strong variations of core concepts can be represented by customs' attributes, which is represented in the fourth ring.

iStarML 1.0 was defined according to an extensible *i** metamodel [1]. The main goal was make the language as much extensible as possible. The iStarML metamodel (see **Fig. 2**) contains six different areas corresponding to the six types of core concepts: actors, intentional elements, dependencies, actor's boundary, intentional element links and actor association. Each area is considered a category that drove the structure of the XML tags and properties according to the ring structure.

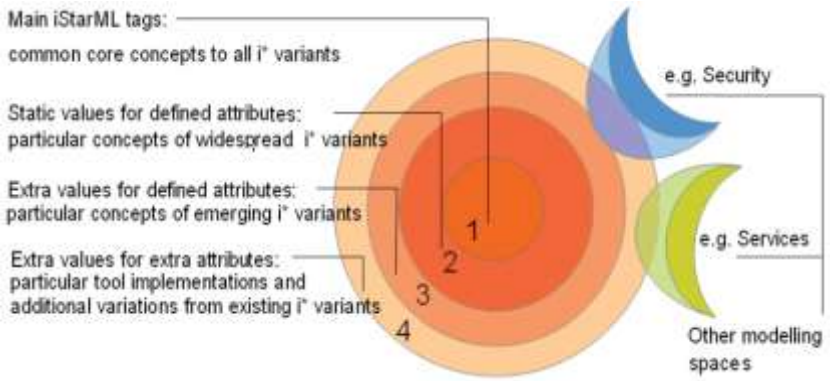

Fig. 1. Concentric ring structure of iStarML.

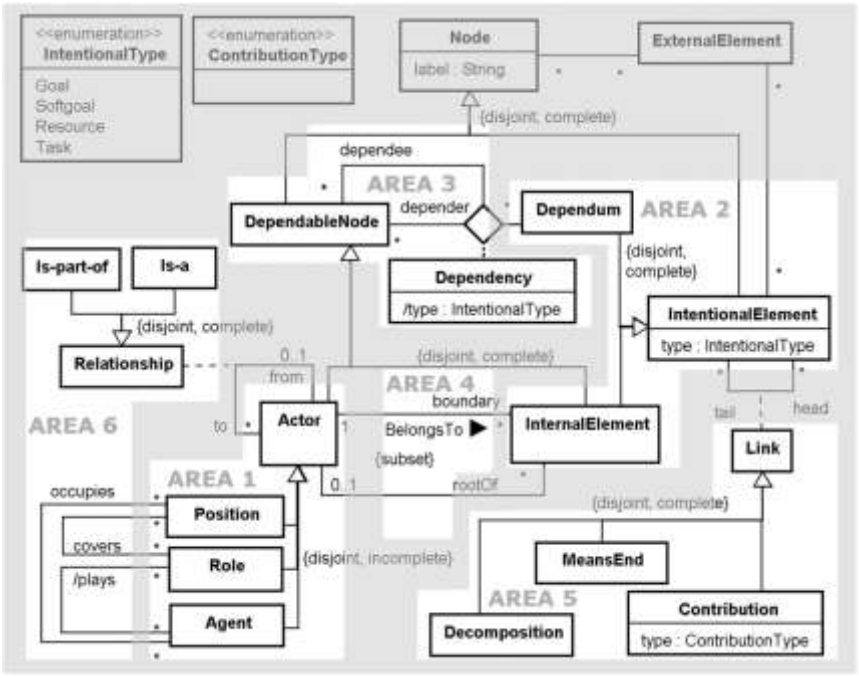

Fig. 2. The core concepts in the context of the *i** metamodel [1]

2 iStar 2.0: Basics and Structure

The iStar 2.0 main goal is evolving the basic concepts of *i** into a consistent and clear set

of core concepts, not losing the open the ability of *i** to tailor the framework. In this sense, iStar 2.0 is aligned to the iStarML goal of making the language as much extensible as possible. Referent to the concentric ring structure of iStarML (Fig 1), the iStar 2.0 is defining the concepts included in the two inner rings.

The iStar 2.0, presented in [3], includes its metamodel (Fig. 3), describing the language constructs of iStar 2.0 and some restriction on their use. This new version mainly includes the same kind of concepts (first ring): actors, actor association links, intentional elements, intentional element links and dependencies. The main difference with its predecessor is the kinds of each concept (second ring) and the rules behind the constructs.

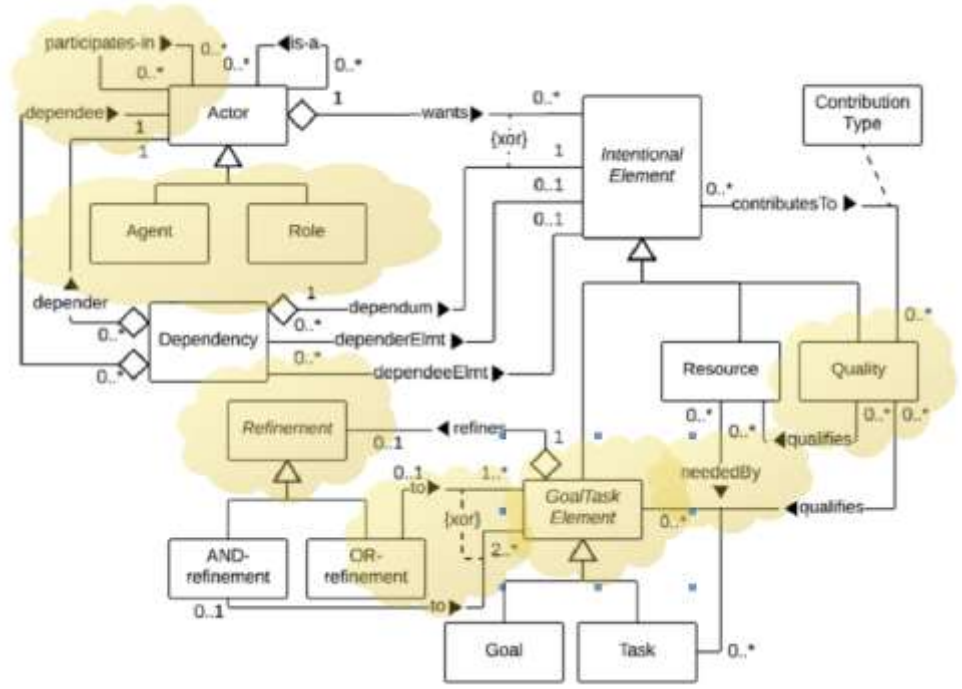

Fig. 3. Areas of evolution of *i** to iStar 2.0 metamodel [3]

3 Towards iStarML 2.0

As any other modelling language, its wide use has resulted in a set of new requirements which has been discussed by *i** research community. First, scalability as the problem of dealing with large *i** models [4] and modularity as its solution. Second, to make analysis on *i** models in order to propose alternative designs or to assess elements of them [5][6]. Third, traceability and representation of incomplete models [7].

Moreover, we aim to keep expressiveness of first iStarML proposal and to add backward compatibility that is a common criterion for format evolution [8]. This last requirement imposes a cross-cutting issue for all modification which implies that valid iStarML 1.0 files must be well-formed iStarML 2.0 files.

In Table 1 we have summarized a set of addition or change proposals to iStarML1.0 in order to consider above requirements. Additionally to the requirements coming from the iStar 2.0, we also include an initial solution addressing traceability, scalability and analysis. Although traceability is a relevant requirement, it seems necessary to analyse if existing version control tools (like Mercurial, Apache subversion and so on) may give enough support. Including the notion of view defined by iStarML 2.0, the boundary can be open or closed, and extending this property to the other elements in the model, we are addressing in an initial way the scalability requirement. Finally, for analysis support, we include new properties in some elements in order to define metrics associated to them.

Table 1. Proposal of new iStarML 2.0 elements

Requirement	iStarML 1.0's element to review [9]	Proposal to iStarML 2.0	Interpretation under iStar 2.0
To include iStar2.0's actors	<actor> Attribute type: type={agent, role, position, *}	type={agent, role, *}	*position* value should be processed like any other instance of *
To include iStar2.0's intentional elements	<ielement> Attribute type: type={goal, softgoal, task, resource, *}	type={goal, task, resource, quality, *}	*softgoal* value should be processed like any other instance of *
To include iStar2.0's intentional element link types	<ielementLink> Attribute type: type={decomposition, means-end, contribution}	type={decomposition, means-end, contribution, refinement, qualification, neededby}	Deprecated types should be replaced by new ones (means-end and decomposition by refinement)
Requirement	iStarML 1.0's element to review	Proposal to iStarML 2.0	Guide to compliance with iStar 2.0
To include iStar2.0's actor links	<actorLink> Attribute actorLink-type actorLink-type=	actorLink-type= {is_part_of, is_a, instance_of, plays, covers, occupies,	All the actor links, except of the is_a, have been replaced by participates_in. The

	{is_part_of, is_a, instance_of, plays, covers, occupies, <string> }	participates-in, <string> }	previous links can be processed as an instance of <string>
To enable interoperability of unfinished models (traceability)	<dependency> Definition: *dependerTag* {*dependerTag*} {*dependeeTag*}	{dependerTag | dependeeTag}	
To enable scalability at actor's rationality level	<boundary>	New attribute: [style={"open"|"closed"}]	The boundary and target intentional elements from wants association must be hidden when style is closed
To enable scalability at intentional element level	<ielement>	New attribute: [style={"open"|"closed"}]	Source intentional elements from refines association must be hidden when style is "closed".
To enable modularity at participant actor level	<actor>	New attribute: [style={"open"|"closed"}]	Actors in participates-in association (source) must be hidden when style is "closed".
To enable analysis	<actor> and <ielement>	New contained tag <metric> Metric attributes: name=<name> value=<value> [type=<metricType>]	

4 Conclusions and Future Work

The formulation of a new standard is a collective task of several stakeholders in order to solve a common problem under a shared perspective. This have addressed the formulation of iStarML 2.0 at an early stage. Firstly, we have reviewed the design principles of iStarML 1.0 and gathered from the *i** research community their main evolving requirements at modelling time. A key milestone was the generation of a consensuated iStar2.0 metamodel which has motivated four of the nine extensions to iStarML 2.0. The other extensions are

related to modularity, one to enable *i** model assessment and propagation analysis and one to deal with a particular element of traceability.

The future work includes promoting a shared standard proposal and to provide basic tools to deal with generation, reading, parsing, analysis and evolution of *i** models represented in iStarML 2.0

Acknowledgments. This work has been supported by the Spanish project EOSSAC (TIN2013-44641-P).

References

[1] Carlos Cares, Xavier Franch, Anna Perini, Angelo Susi: Towards Interoperability of *i** Models using iStarML. In Computer Standards & Interfaces 33(1): 69-79.

[2] Lotman, Yuri M: "On the semiosphere." Σημειωτκή-Sign Systems Studies 1: 205-229, 2005

[3] Fabiano Dalpiaz, Xavier Franch, Jennifer Horkoff. *iStar 2.0 Language Guide*. arXiv:1605.07767, May 2016

[4] Franch, X.: Incorporating Modules into the *i** Framework. In International Conference on Advanced Information Systems Engineering (CAiSE), LNCS 6051: 439-454. Springer Berlin Heidelberg, 2010

[5] Horkoff J, Yu E. Interactive goal model analysis for early requirements engineering. Requirements Engineering. 2016 Mar 1;21(1):29-61.

[6] Franch X, Grau G. Towards a catalogue of patterns for defining metrics over *i** models. In Int. Conf. on Advanced Information Systems Engineering (CAiSE 2008) Jun 16 (pp. 197-212). Springer Berlin Heidelberg.

[7] Serrano M, Leite J. A rich traceability model for social interactions. In Proc. of the 6th Int. Workshop on Traceability in Emerging Forms of Software Engineering, May 23 (pp. 63-66): 2011, ACM

[8] Lampathaki F, Mouzakitis S, Gionis G, Charalabidis Y, Askounis D. Business to business interoperability: A current review of XML data integration standards. Computer Standards & Interfaces. 2009 Nov 30;31(6):1045-55.

[9] iStarML. The *i** Mark-up Language: Reference Guide. August 2007. Avaliable online at http://www.upc.edu/gessi/istar/tools/istarml/resources.html (last access Juny 2016)