Supporting i* model integration through an ontology-based approach

Karen Najera[1,2], Anna Perini[2], Alicia Martinez[1], and Hugo Estrada[1]

[1] National Center of Research and Technological Development - CENIDET
Interior Internado Palmira S/N, 62490 Cuernavaca, Morelos, Mexico
{karen.najera, amartinez, hestrada}@cenidet.edu.mx
[2] Fondazione Bruno Kessler - IRST, Center for Information Technology (CIT)
Via Sommarive, 18, 38050 Trento, Italy
{knhernandez, perini}@fbk.eu

Abstract. The i* framework is widely used for organizational modeling. It has been applied in different application domains, hence many i* variants have been proposed. Sharing information and integration of models expressed in i* variants imply interoperability problems. The interoperability has been approached at different levels, e.g. through unified metamodels, or with an interchange format for representing i* models. As a preliminary step toward addressing the interoperability problem, our aim in this paper is to investigate the role of an ontology-based metamodel to realize integration of models expressed in i* variants, bringing the advantages of ontologies to the organizational modeling domain. We describe the ontology-based metamodel of the i* framework and the process followed to build it, exploiting Model Driven Engineering ideas. Moreover, we describe a first application of our approach to define a tool-supported process aiming at generating ontologies corresponding to models expressed in the i* language.

Keywords: i*, conceptual modeling, ontologies, model-driven engineering, model transformations.

1 Introduction

The i* framework [10] is a well known organizational modeling technique, that inspired several studies and extensions. It uses strategic relationships to model the social and intentional context of an organization. Nowadays, many research projects use the i* framework in different application domains, hence many i* variants have been proposed, such as Tropos [6], Service-oriented i* [4] and so on.

Since models are created with particular variants, sharing information and integrating models expressed in different i* variants imply interoperability problems. The interoperability has been approached at different levels, e.g. through the definition of a unified metamodel(e.g. [2]) and [8]), or with the introduction of interchange format for representing i* models such as iStarML [3]. As a preliminary step toward addressing the interoperability problem, our aim in this paper is to investigate the role of an ontology-based metamodel to realize integration of models expressed in i* variants, bringing the advantages of ontologies to the organizational modeling domain.

Recent literature [9], [7] put in relationship ontologies and the layered architecture used in the Model Driven Engineering (MDE) approach (where models, metamodels and metametamodels correspond to the M1, M2 and M3 layers, respectively) with the purpose of bridging models and metamodels with ontologies. The authors specify the advantages of using ontologies, namely: ontology linking service, where models and metamodels are transformed in terms of ontologies to improve interoperability; querying, automated reasoning and others. In our work, in addition to model integration, we aim at providing a solution, which permits bringing ontologies advantages to the organizational modeling domain. In this research work we have developed the metaontology of the i* framework. It has been built using the standard semantic web language OWL [1] exploiting MDE ideas. As a first application of our approach, we described a tool-supported process aiming at generating ontologies corresponding to models expressed in the i* language.

2 Objectives

The main objective of this research work is to propitiate the integrability of models represented in the i* modeling language or represented in any of its variants through the use of ontologies. For the accomplishment of this main objective we have identified three specific objectives: the first corresponds to the development of a metaontology (which we have called OntoiStar) for representing the i* metamodel; the second corresponds to the development of a methodology for guiding the process of integrating additional concepts of i* variants into OntoiStar for improving the interoperability; and the last one is related to the use of OntoiStar as the underlying baseline for the automatic transformation of a model represented in any i* variant into ontologies derived from the concepts of OntoiStar. At present we have addressed the first objective and partially the third one which will be totally covered after we achieve the second objective (since now we have only implemented the automatic transformation from i* models into OntoiStar).

3 Scientific contributions

We aim to show that the use of ontologies is an appropriate way for supporting integrability of models expressed in i* variants and a promising approach towards tackling the i* variants interoperability problem. For that reason, our scientific contributions addressed so far are related with a semi-automated approach to generate organizational ontologies from an i* model (both the strategic dependency and the strategic rationale model). We first developed the metaontology OntoiStar which corresponds to the ontological representation of the i* metamodel; and then we developed a tool to automatically transform an i* model to instances of OntoiStar. OntoiStar has been built using the OWL language [1], as it is the standard semantic web language, the organizational knowledge can be shared to be understandable not only for humans but also for software system to automatically discover the meaning of business resources defined in the models. OWL allows to define axioms in OntoiStar for defining the semantic of each i* variant and the definition of syntactic constraints. Therefore it is possible to analyze the syntactic correctness of i* models. Moreover, OWL supports inference rules which we will apply for avoiding the loss of information caused by differences in the integrated i* variants.

The constructs included in the i* metamodel were taken from two i* metamodel proposals [2] [8], which consist of mainly the common constructs of the i* variants. We developed OntoiStar based on those common constructs and selected characteristics described below of each proposal. We have applied a transformation language bridge approach [9] based on MDE. MDE is a software development methodology that recognizes models as a key role for describing the system to be developed. It is based on layered architectures, where models, metamodels and metametamodels correspond to the M1, M2 and M3 layers, respectively. In Fig. 1 we present the two layered architecture of this approach. On the left side, the i* modeling language architecture, where the i* metamodel is located in the M2 layer, and it is described by its metametamodel (represented in the Unified Modeling Language) in the M3 layer, and on the right side, our proposed ontology architecture, where the resultant OntoiStar has been located in the M2 layer and it is described by the OWL metamodel. The transformation bridge then is defined in the M3 layer. It contains the mapping rules between concepts from the i* metametamodel, such classes and associations and concepts from the OWL metamodel, such classes and properties. The transformation bridge is applied in the layer M2, transforming the i* metamodel into the metaontology: OntoiStar. For transforming an i* model to instances of OntoiStar (in the layer M1), we propose an automatic transformation tool.

Applying this approach we generate a logical knowledge base, where the terminological part is provided by the metaontology OntoiStar and the assertional part corresponds to a specific organization description represented in an i* model, which is mapped as instances of ontoiStar.

The transformation bridge is defined as follows:

(1) Identifying constructs from the i* metamodel and from the OWL language.

1. We adopted from the metamodels presented in [2] and [8] the common characteristics, including concepts, relations and attributes. The specific characteristics adopted for each one are described in Table 1.
2. The main constructs of the OWL language are: Class, Object property and Data property and the axioms: ObjectPropertyDomain, ObjectPropertyRange and DataPropertyDomain.

(2) Defining the relationships between constructs from the i* metamodel and the OWL language. Based on this definition, we proposed the following transformation rules:

Table 1. Specific characteristics adopted from i* metamodels

Reference model for i* [2]	Unified metamodel for i* [8]
Dependum class.	All concept relationships representation as classes and associations.
Attributes: Label and Type, from Node class and IntentionalElement class respectively.	The high abstraction level class iStarRelationship.
ContributionType enumeration types: '+' and '-'.	The iStarRelationship subclasses: ActorRelationship, DependencyRelationship and InternalElementRelationship.
Class properties (except the disjoint property between DependableNode and IntentionalElement).	

1. Each concept, concept relationship and enumeration class included in the i* metamodel is represented as a class in OWL.
2. Each association included in the i* metamodel is represented as an object property in OWL. Where its domain corresponds to the association source and its range corresponds to the association target.
3. Each class property included in the i* metamodel is represented with axioms in OWL.
4. Each enumeration element included in the i* metamodel is represented as a class instance of the owner enumeration class in OWL.
5. Attributes. There are two types of attributes:
   (a) Each enumeration type attribute included in the i* metamodel is represented as an object property in OWL. Where its domain corresponds to the owner class and its range corresponds to the enumeration class.

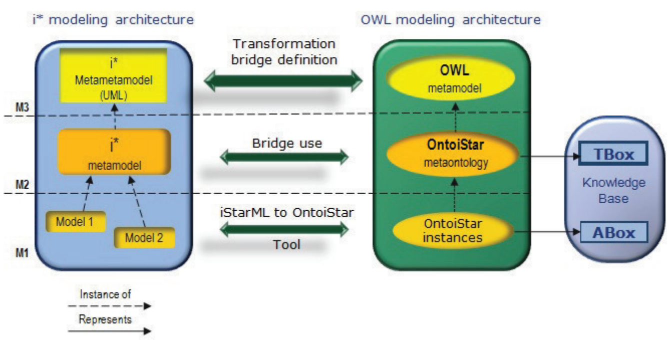

Fig. 1. Transformation bridge

Fig. 2. Fragment of the OntoiStar taxonomy

(b) Each primitive data type attribute included in the i* metamodel is represented as a data property in OWL. Where its domain corresponds to the owner class and its range corresponds to the primitive data type.

We implemented OntoiStar using the ontology editor and knowledge-base framework Protègè [5]. A Fragment of the OntoiStar taxonomy is presented in Fig. 2.

3.1 Generating ontologies corresponding to i* models

A software tool has been developed where given the i* model represented in iStarML format and the metaontology OntoiStar, the tool populates the metaontology with instances of i* elements belonging to a specific organizational model. For doing that, mapping rules between the iStarML elements (tags and attributes) and the concepts in OntoiStar (classes and properties) have been defined. The tool overview is presented in Fig. 3. The first step is the development of the i* model which can be realized with i* modelers or editors that enables producing a model specified in the iStarML format. The automatic transformation process consists of parsing the iStarML file and, according to the defined mapping rules implemented in the tool *iStarMLtoOntoiStar*, instantiating the corresponding classes and properties in the metaontology OntoiStar. The output of the tool contains the metaontology OntoiStar with instances representing the knowledge content in the i* model. It could be edited with an ontology editor or it could be the input of an ontology based application.

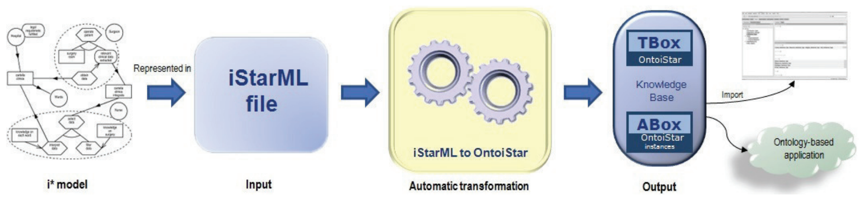

Fig. 3. Tool overview

4 Conclusion

In this paper we presented a semi-automated approach to generate organizational ontologies from an i* model. Specifically we described how we developed the metaontology of i* modeling language (OntoiStar) and the transformation process that we have applied to derive from i* models their corresponding OWL ontologies. As the basis of our future work, OntoiStar contains the common and relevant characteristics of i* variants obtained from two i* metamodel proposals.

5 Ongoing and future work

Currently, we are addressing the integration of i* variants to OntoiStar. The methodology is applied to the i* variants: Tropos and Service oriented i*. Moreover, we are also extending the tool for covering the automatic transformation for models of those variants. In the future the methodology and the automatic transformation will be applied to more i* variants. Towards addressing the interoperability problem in a more systematic way, we will investigate the transformation of the knowledge contained in the metaontology OntoiStar to any of the i* variants already integrated in OntoiStar. We plan the use of inference rules for avoiding the loss of information caused by differences in the i* variants. Empirical evaluation of the effectiveness of ontologies to solve the interoperability between i* variants will follow.

References

1. S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. OWL Web Ontology Language Reference. Technical report, W3C, http://www.w3.org/TR/owl-ref/, February 2004.
2. C. Cares, X. Franch, E. Mayol, and C. Quer. A Reference Model for i*. In *Social Modeling for Requirements Engineering*, pages 573–606. MIT Press, 2010.
3. C. Cares, X. Franch, A. Perini, and A. Susi. Towards interoperability of i* models using istarml. *Computer Standards & Interfaces*, 33(1):69–79, 2011.
4. H. Estrada. *A service-oriented approach for the i* framework*. PhD thesis, Valencia University of Technology, Valencia, Spain, 2008.
5. J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, and N. F. Noy. The evolution of Protégé: an environment for knowledge-based systems development. *Int. J. Hum.-Comput. Stud.*, 58(1):89–123, 2003.
6. P. Giorgini, J. Mylopoulos, A. Perini, and A. Susi. The Tropos Methodology and Software Development Environment. In *Social Modeling for Requirements Engineering*, pages 405–423. MIT Press, 2010.
7. B. Henderson-Sellers. Bridging metamodels and ontologies in software engineering. *Journal of Systems and Software*, 84(2):301–313, 2011.
8. M. Lucena, E. Santos, C. T. L. L. Silva, F. M. R. Alencar, M. J. Silva, and J. Castro. Towards a unified metamodel for i*. In *Research Challenges in Information Science*, pages 237–246, 2008.
9. S. Staab, T. Walter, G. Gröner, and F. S. Parreiras. Model driven engineering with ontology technologies. In *Reasoning Web*, pages 62–98, 2010.
10. E. S.-K. Yu. *Modelling strategic relationships for process reengineering*. PhD thesis, University of Toronto, Toronto, Ont., Canada, 1996.