

# Managing *i*\*-based Reusable Context Models Elements through a Semantic Repository

Karina Abad, Wilson Pérez, Juan Pablo Carvallo

Computer Science Department, Universidad de Cuenca, Ecuador

{karina.abadr, wilson.perez,  
pablo.carvallo}@ucuenca.edu.ec

**Abstract.** In previous works, we have presented a catalogue of patterns aimed at improving context models' construction by departing from a set of predefined elements, instead of starting from scratch. Patterns were identified from *i*\*-based context models constructed for several industrial cases (where the DHARMA method was used), packaged and stored in the form of *i*\* SD actors and dependencies. In this paper we go one step further and present a framework, based in semantic technologies, designed to improve utilization of the catalog, which has been deployed as a semantic repository. The framework makes easier the construction of *i*\*-based context models, by recommending knowledge elements stored in the repository.

**Keywords:** DHARMA Method, semantic repository, iStar Framework

## 1 Introduction

Modern enterprises rely on Information Systems (IS) specifically designed to manage the increasing interactions with their context. Design of such systems requires deep understanding of the enterprise context and strategies. To support this process, we have proposed the DHARMA method [4], aimed at discovering IS architecture departing from the construction of Context Models (CM) expressed in *i*\*. The application of the first activities of this method in several industrial and academic cases, allowed us to construct a catalog of patterns of context elements, represented as *i*\* Strategic Dependency models, which include generic environmental actors and their associated strategic dependencies [3]. Patterns in the catalogue are used to improve CM construction, by reusing the stored intentional elements in new modes, instead of starting from scratch. In this paper we go one step forward and introduce the Semantic DHARMA (S-DHARMA) framework, designed to automate the application of the DHARMA method, making use of semantic techniques and an evolution of the catalogue of patterns, which has been redefined as a semantic repository.

## 2 Related Works and Literature Review

There are several tools supporting *i\**-based modeling and its variants (TROPOS, Service Oriented *i\**, etc.), e.g., OpenOME [5], E4J [7] and iStarTool [6], all of them allow users to create and validate *i\** models. In [3], authors conduct a systematic comparison of *i\** modelling tools, focusing in both, syntactic and semantic support, in order to help users to choose the most adequate tool. However, all these tools need user interaction in order to construct *i\** models. In [10], a collection of RESTful Web Services intended to manipulate existing *i\** models is presented. More specifically, one service creates, edits, deletes and stores iStarML files in an eXistDB; a second service creates SVG files and shows them in a Web browser as an *i\** diagram, achieving semi-automatic generation of *i\** SD models. In our approach, we aim at the automatic generation of *i\** models from tables resulting from each DHARMA activity.

In [8], the development of the ontological meta-model for *i\** variants is presented; this meta-model contributes to the development of a tool for the automatic generation of organizational ontologies [9]. Departing from an iStarML file, the tool joins the ontological meta-model with domain models in order to extend the vocabulary, the meta-model in [8] will be the basis for this project.

## 3 The DHARMA Method

The DHARMA (Discovering Hybrid ARchitectures by Modelling Actors) method (see Fig.1), aims at the definition of enterprise architectures using the *i\** framework [14]. The process is initiated in activity 1 by modeling the enterprise context. The organization and its strategy are carefully analyzed, to identify its role inside the context, making evident the Context Actors (CA) and the Organizational Areas (OA) structuring the organization. *i\** SD models are built and used to support reasoning and represent results from this activity. Next, in activity 2, an IS-to-be is placed into the organization and the impact that it has over the elements in the CM is analyzed. Strategic dependencies identified in the previous activity (internal and context), are inspected to determine which of them may be totally or partially satisfied by IS. In activity 3, dependencies included in the IS CM are analyzed and decomposed into a hierarchy of goals required to satisfy them. The goals represent the services that IS must provide, to support interaction with CA and OA activities. An *i\** SR diagram for the system is built. Finally, activity 4, is used to identify the generic architecture of the system (system actors that structure the system, the services -goals- that must be covered by each of them and the relationships among them).

## 4 Semantic DHARMA

S-DHARMA is a web framework that allows the creation and maintenance of *i\** models, following the activities defined in the DHARMA method. Its main objective is to make easier the construction of organizational CMs and to visualize the models resulting from each activity of the method.

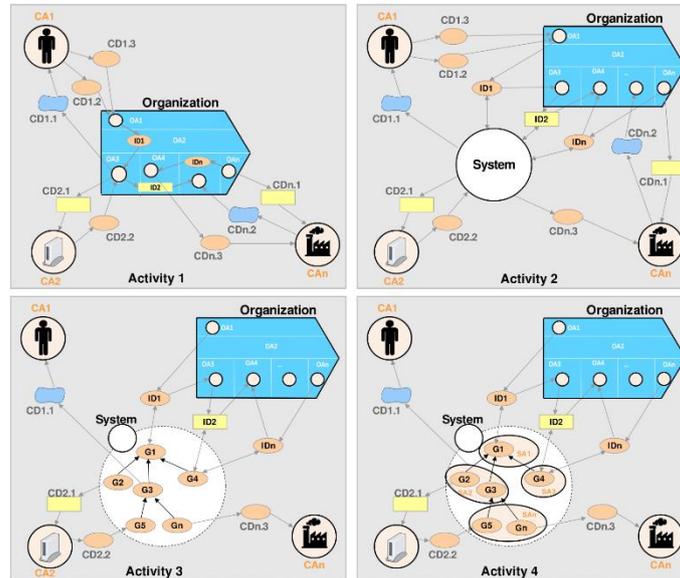


Fig. 1. The DHARMA Method

As starting point, the framework recommends the dependencies stored in a semantic repository, and then, in a refinement phase, the user can create additional dependencies or modify existing ones using the tabular form of *i*\* SD model presented in [4]. The framework makes use of OntoStar [8] and the DHARMA ontology [1], thus, supports all the elements of the *i*\* notation.

#### 4.1 Architecture

Architecture of the semantic DHARMA framework (see Fig.2) is defined in three layers, client, server and data layers. The **client layer** allows the interaction with the users, and gives him the possibility to create, edit, store, and visualize context models based in the *i*\* notation, it has been constructed using JavaScript and makes use of the D3.js library to generate *i*\* diagrams.

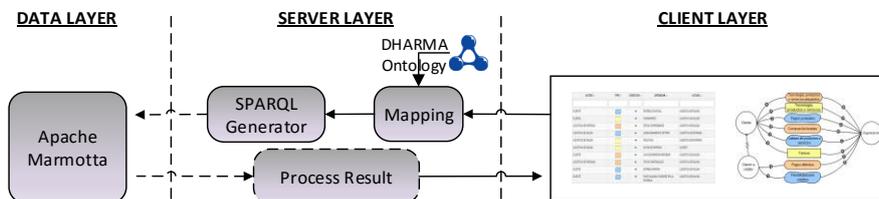


Fig. 2. Semantic DHARMA Architecture

The **server layer** works as a bridge between the client and the data layers. It receives requests from the client layer in JSON format by invoking RESTful Web Services, and

then, through a mapping process between the data and the DHARMA ontology, creates SPARQL queries [13], which are sent to the data layer through Apache Marmotta [11] Services. The S-DHARMA ontology conceptualizes the knowledge of the DHARMA method in an ontological network (see fig. 3), which integrates different domains and their vocabularies, such as Offer-job and Classification [12] ontologies, to cover concepts of size, sector and industry of an organization. The OntoStar+ ontology is the basis of the S-DHARMA ontology, it contains concepts of actors, dependencies, intentional types, etc. The ValueChain [1] ontology is used to cover requirements of the OAs, and to manage features related to the activities of the DHARMA method, such as the definition of relations between actors, organization, dependencies, areas, direction of the dependencies, and the concept of coverage used in activity 2 of the DHARMA method, to define if a dependency can be totally or partially satisfied by the IS, etc.

The **data layer** stores and retrieves information from the triplestore based in Apache Marmotta. The SPARQL queries generated in the server layer, either for insert, update or delete, are processed in this layer and results are returned to the server in JSON format. Immediately, the server layer retrieves the response to the client-layer.

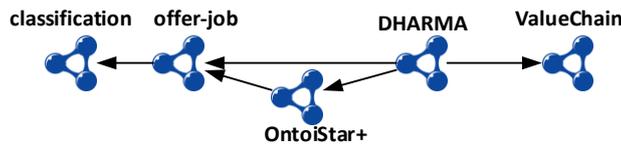


Fig. 3. DHARMA ontology network

#### 4.2 Using the framework

As mentioned before, the S-DHARMA framework supports the activities of the DHARMA method; in this subsection we will focus on the support for activities 1 and 2. The first activity of the DHARMA method, “Context Modelling”, requires a deep analysis of the organization and its OAs, so, as first step, the organization must be created, by inputting its name, size, sector and industry. Immediately the framework will show a group of OAs of interest for the organization under study, which are automatically recommended and retrieved from the catalog and displayed as a tree, which enables a *consultor* (user conducting the modeling activities) for its selection and the creation of new OAs as needed. Next, the description of each OA and the specification of a responsible for each of them is required; this information is stored in order to identify the administrative stakeholder to be interviewed by the consultant during the modeling process. Then, the analysis and modeling of each OA should be performed; the process starts by selecting a group of actors, external and internal, with whom the OAs interact (also retrieved and recommended from the semantic repository and displayed as a tree). After that, the framework queries the semantic repository (see figure 4a for a query example) and presents a group of generic dependencies of interest for each pair of related actors (also retrieved from the semantic repository), the user can accept or not the dependencies suggested.

After saving the dependencies, the framework will show the tabular form of the CM (see fig.4b), and additional options to support a refinement phase, where the user can

either create specific dependencies needed to improve the model, or parametrize dependencies and actors. The framework also includes an option to show *i*\* diagrams for each pair of actors modelled in this activity (see fig. 4c).

The second activity of the DHARMA method, “System Context Modelling”, consists in the identification of the dependencies that can be solved by the IS, so, the framework allows its identification using radio buttons to specify whether a dependency can be totally or partially covered by the IS. After that, new *i*\* diagrams can be generated, including the IS as an actor.

```
SELECT distinct ?URI ?Actor1 ?Dependum ?Actor2 ?IntentionalType ?direction WHERE {
GRAPH <http://www.ucaenca.edu.ec/graph/DHARMA/production>
{
?DependerLink OntoStar:has_Dependency_DependerLink_target_ref ?DependerE .
?DependerLink OntoStar:has_Dependency_DependerLink_source_ref ?URI .
?DependeeLink OntoStar:has_Dependency_DependeeLink_target_ref ?DependeeE .
?DependeeLink OntoStar:has_Dependency_DependeeLink_source_ref ?URI .
?URI OntoStar:Node_sannotation ?direction .
?URI dharma:has_Dependency_Area_source_ref ?URIArea .
?DependerE OntoStar:Node_label ?Actor2 .
?DependeeE OntoStar:Node_label ?Actor1 .
?DependumLink OntoStar:has_Dependency_DependumLink_target_ref ?DependumE .
?DependumE OntoStar:Node_label ?Dependum .
?DependumE OntoStar:has_IntentionalElement_IntentionalType ?IntentionalTypeE .
?IntentionalTypeE rdfs:label ?IntentionalType .
}}

```

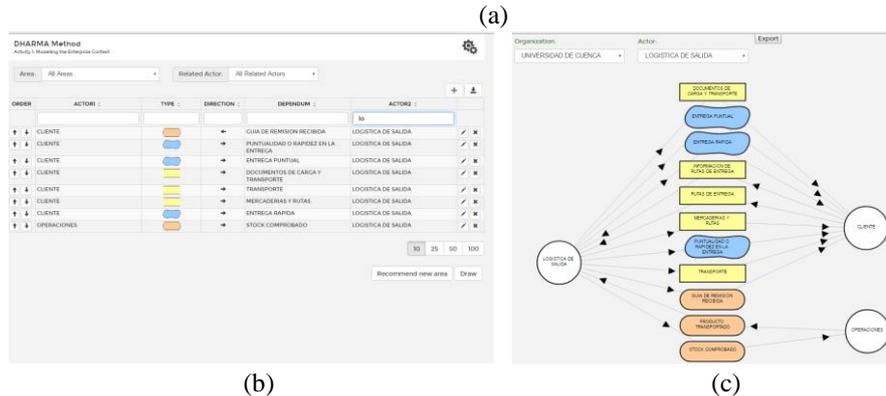


Fig. 4. Screenshots of the Semantic DHARMA Framework

## 5 Conclusions and Future Work

In this paper, we presented the S-DHARMA framework, a web framework to support the construction of CM based in the activities of the DHARMA method. Its interactive interface, helps users to create models in short time, reducing typical errors commonly associated to the *i*\* notation. The framework makes use of semantic technologies that allow the retrieval, recommendation and instantiation of *i*\* SD context models elements which have been annotated, packaged and stored in a semantic repository.

Currently, the S-DHARMA framework supports the construction of SD models required by activity 1 and 2 of the DHARMA method. As future work, we plan to extend it to support SR models and the identification of the system architecture required by activities 3 and 4 of the method. Also, we pretend to incorporate semantic reasoners

and include synonymy modules to avoid dependency duplication. Furthermore, we pretend to adapt the constructors in order to implement the iStar2.0 standard and allow the user to import and export *i\** models in existing formats, such as iStarML. Finally, we pretend to incorporate the framework in academic activities aiming at improving the DHARMA Method and the *i\** framework teaching-learning process.

## Acknowledgment

This work has been conducted with the support of a grant from the Research Direction of the University of Cuenca (DIUC), in the framework of the project “Hacia la identificación, validación, y enriquecimiento semántico de patrones de modelos de contexto a ser utilizados como base para la definición de arquitecturas de sistemas de información empresariales”.

## References

1. Abad, K., Carvallo, J. P., Espinoza, M., Saquicela, V., *Hacia la Creación de un Repositorio Semántico de Modelos de Contexto Basados en i\* y el método DHARMA*. Revista Ibérica de Sistemas y Tecnologías de la Información, 2016, pp. 41-56.
2. Abad, K., Carvallo, J. P., Peña, C. *iStar in Practice: On the identification of reusable SD Context Models Elements*. Proceedings of *i\** Workshop 2015, pp. 43-48
3. Almeida, C., Goulao, M., Araújo, J. *A Systematic Comparison of i\* Modelling Tools Based on Syntactic and Well-formedness Rules*. Proceedings of *i\** workshop 2013.
4. Carvallo, J. P., Franch, X. *Descubriendo la arquitectura de sistemas de software híbridos: un enfoque basado en Modelos i\**. Proceedings of RE'09, pp. 45-56.
5. Horkoff, J., Yijun, Y., Yu, E. *OpenOME: An Open-source Goal and Agent-Oriented Model Drawing and Analysis Tool*. CEUR Workshop Proceedings, vol. 766, 2011.
6. Malta, A., Soares, M., Santos, E., Paes, J., Alencar, F., Castro, J. *iStarTool: Modeling requirements using the i\* framework*, vol. 766, 2011, pp. 163-165.
7. Merlin, L. P., De Borba Silva, A. L., Santander, V. F. A., Da Silva, I. F., Castro, J. *Integrating the E4J editor to the JGOOSE tool*. CIBSE 2015 – XVIII Ibero-American Conference on Software Engineering, 2015, pp. 646-659.
8. Najera, K., *An ontology-based approach for integrating i\* variants*. Master's thesis, National Center of Research and Technological Development, Cuernavaca, Morelos, Mexico, 2011. [www.tagoon.semanticbuilder.com/NajeraThesis.pdf](http://www.tagoon.semanticbuilder.com/NajeraThesis.pdf).
9. Najera, K., Vazquez, B., et. al.. *TAGOON+: Generation and Integration of Organizational Ontologies*. Proceedings of *i\** workshop 2013, pp. 125-127.
10. Petrushyna, Z., Ruppert, A., Klamma, R., Renzel, D., Jarke, M. *i\*-REST: Light-Weight i\* Modeling with RESTful Web Services*. Proceedings of *i\** workshop 2014.
11. The Apache Software Foundation, Apache Marmotta, <http://marmotta.apache.org/> (28/6/16)
12. Villazón-Terrazas, Boris; Ramírez, Jaime; Gómez-Pérez Asunción, Human Resources Management Ontology, <http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/ontologies/99-hrmontology/index.html> (last accessed: 28/06/2016)
13. W3C, SPARQL Query Language for RDF, <https://www.w3.org/TR/rdf-sparql-query/> (last accessed: 28/06/2016)
14. Yu, E., *Modelling strategic relationships for process reengineering*, PhD thesis, University of Toronto, Department of Computer Science, Canadá, 1995.