

A Catalogue of Reusable Context Model Elements Based on the *i** Framework

Karina Abad¹, Wilson Pérez^{1(✉)}, Juan Pablo Carvallo²,
and Xavier Franch³

¹ University of Cuenca, Cuenca, Ecuador
{karina.abadr,wilson.perez}@ucuenca.edu.ec

² University of Azuay, Cuenca, Ecuador
jpcarvallo@uazuay.edu.ec

³ Universitat Politècnica de Catalunya, Barcelona, Spain
franch@essi.upc.edu

Abstract. The definition of the context of a system is one of the most relevant activities in the early phases of information systems engineering. It allows system engineers to narrow the system scope, by defining well established system boundaries. In practice, outlining a system context model is complex and cumbersome. In order to support context modeling, in this paper we propose a catalogue of context model elements expressed in *i**, which can be reused as building blocks in the construction of context models for new systems. We describe the process used for the identification of a set of actors and dependencies recurrently appearing in several academic and industrial cases, and the process to store them into a catalogue of reusable *i** context dependencies.

Keywords: Goal-oriented model · Model reuse · iStar framework · Context model

1 Introduction

The definition of the context of a system is one of the most relevant activities in the early phases of information system engineering [1]. It allows system engineers to narrow the system scope, by defining well established boundaries in relation to the actors placed in its context (organizations, people, cyber-technical systems, etc.) and the interactions (processes) that it must support to communicate with them. The definition of system context requires at least four facets to be considered [1]:

- **Use:** the kind of users that will interact with the system and their abilities and limitations (physical or mental).
- **Object:** the system-to-be and its functional coverage.
- **System:** the platform, protocols and technologies required to run and support system operation.
- **Development:** standards and tools used to drive system construction.

In practice, modeling a system context is complex and cumbersome. It requires continuous and fluid communication among system designers, stakeholders and

managers visualizing and defining business strategy, but also notations and tools required to support and document their interaction and agreements in the form of a Context Model (CM).

In order to support this process, in the last few years we have been intensively using the *i** framework [2] to model system contexts, and proposed the DHARMA [3] method to discover the system architecture departing from these models. The application of DHARMA in a good dozen of cases led us to propose some patterns aiming at improving CM construction [4], by reusing some elements that repeatedly appeared in several of these cases. Although these patterns proved to be useful in practice [4], after conducting over 36 academic and industrial experiences, we concluded that, due to the creative nature of both managers and organizations, even in the same industry, the patterns tend to structure and behave in very dissimilar ways, making their application highly difficult. These cases also proved that that reuse of more atomic sets of elements was not only feasible, but also a way to more efficiently construct CM [5].

In this paper, we present a catalogue of reusable CM elements expressed in *i** to be used as building blocks in the construction of CM for new systems. We describe the process used to identify a set actors and related dependencies, which frequently occurred in these cases, and the process to store them into a catalogue of reusable *i** context dependencies. Dependencies can be independently reused as atomic patterns or together, in subsets of dependencies selected in relation to labels assigned to actors. These subsets of dependencies structure larger “dynamically constructed” patterns or CM chunks, which can be parametrized in relation to the specific domain.

The rest of the paper is structured as follows. Section 2 presents the background, including the DHARMA method, and some related work on pattern-based reuse. Section 3 shows the process followed for the catalogue construction. Section 4 summarizes the contents of the final catalogue. Section 5 shows how the catalogue can be used to automatize the construction of CM. Finally, Sect. 6 presents some conclusions and future work.

2 Background and Related Work

2.1 The DHARMA Method

The DHARMA (Discovering Hybrid ARchitectures by Modelling Actors) method [3] aims at the definition of enterprise architectures using the *i** framework [2], which allows representing, modeling and reasoning about socio-technical systems through graphical models based on a set of modeling constructs. DHARMA relies upon two concepts defined by Porter [6]: (1) the model of market forces designed to reason about potential available strategies and how to make them profitable and helpful in the analysis of the influences of market forces; and (2) a value chain that includes primary and support activities. The process consists in four activities (see Fig. 1). Activity 1 models 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. Activity 2 places a

system-to-be into the organization and analyzes the impact that it has over the elements in the CM. Strategic dependencies identified in the previous activity (internal and context), are inspected to determine which of them may be totally or partially satisfied by system. In Activity 3, dependencies included in the CM are analyzed and decomposed into a hierarchy of goals required to satisfy them. The goals represent the services that the system 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).

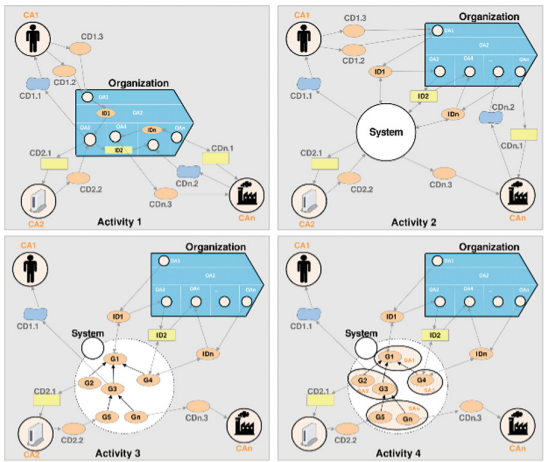


Fig. 1. The DHARMA method

2.2 Related Work on Reuse Through Patterns

The reuse of requirements through patterns has been proposed and used broadly in the field of requirements engineering [7]. Most of them focus on non-functional requirements (NFR) as in [8, 9], where a set of defined patterns is presented; these patterns aim to capture and reuse some specific aspects linked to data security. The PABRE framework [10] makes use of patterns in order to define requirements expressed in natural language. The catalogue used by PABRE was grounded on real software requirement documents and applies to both functional requirements and NFRs.

In the *i** community, we find several approaches too. In [11] an evaluation was carried out around the application of patterns over *i** models, trying to find out if those patterns improve CM construction, finding that their application allow to define elements with a broader coverage. Nevertheless, their construction requires a deep understanding and effort; therefore, in this work it was not possible to demonstrate that their application decreases the complexity of the construction of CM. With a similar aim of exploring pattern application, in [4] we proposed a set of patterns based in the Porter’s model and some strategies, specifically the CRM strategy, and we formulated patterns for this strategy, which are formally described and oriented to industrial

applicability. In [12], we have proposed a first approach for reusing elements through a catalogue of CM elements and showed how to automate the construction of i^* SD based-CM, starting the construction of CM from a solid base instead of departing from scratch.

In this work, we aim to provide guidance in early phases of the Enterprise Engineering process, providing artifacts to bridge communication gaps among technical Enterprise Engineering staff and administrative staff, starting from previous works and improving them. In relation to [10], our contribution is focused in the exclusive use of the i^* notation instead of natural language (which provides an adequate level of abstraction for modeling CM). We improved the catalogue of common elements presented in [12] to include more CM to the analysis, this paper presents the process to construct such catalogue; we also introduce and show the use of parametric actors and dependencies.

3 Catalogue Construction

In this section, we present the process performed to construct the catalogue of CM elements. The process starts with the data collection, then we analyze separately actors and dependencies. Due to the large number of CM elements, each analysis was performed in many steps, as explained below.

3.1 Data Collection

The models upon which we based this analysis were constructed by university students in their final grade project, acting as junior consultants in 36 organizations. The students were trained in the construction of CM, specifically in the i^* notation and the DHARMA method, according to the scope, objectives and activities proposed in such method. Since DHARMA is relatively new in the context of its application, we consider usable the CM constructed by the students according to the study carried out in [13], where the authors concluded that the performance observed in students and professionals is similar when the approach is new. The models were created for the organizations through formal agreements among them and the University of Cuenca. In the study, 27 organizations were small sized, 6 medium sized and 3 large sized. This distribution largely corresponds with the reality of the country where the studies were conducted, whose industrial network is composed by small companies as majority (97, 94%) [14]. The organizations were classified according to NACE Rev2's domains [15] identifying: 12 manufacturing, 16 wholesales, 2 human health, 4 education, 1 transportation and 1 financial.

In each organization, the modelling process started with the construction of CM and finished with the identification of the system architecture required to support its operation. The junior consultants worked in pairs in order to complete each DHARMA activity. To perform Activity 1 of the DHARMA method, an interview with the interlocutor assigned by each organization was conducted; its objective was to get information enough as to allow the identification of actors inside and outside the organization, and to discover the relations among them. The junior consultants

performed the process manually and the final product of this activity was a set of i^* diagrams and its tabular representation according to DHARMA. Each group was able to identify around 31 actors and 58 dependencies per organization in average, with a maximum of 50 actors and 113 dependencies and a minimum of 17 actors and 20 dependencies.

3.2 Actors Analysis

The analysis of actors started by considering two generic groups as defined by Porter [6], namely: 8 external context actors (*Suppliers, Consumers, Strategic Partners, Distributors, Financial Institutions, Regulatory Agencies, Control Agencies and Competitors*) and 9 internal context actors (*Inbound Logistics, Operations, Outbound Logistics, Marketing and Sales, Services, Infrastructure, Human Resources Management, Technology Development and Procurement*). To define the catalogue of actors extending this initial group, we followed a three-step process, described below.

First, the 36 CM were integrated into a single data space, where each actor in a CM was classified as one of the 8 external or 9 internal context actors enumerated above (see results in Table 1).

A total of 1109 actors were found, from which 886 are external and 223 are internal. The most common types of actors are *Suppliers* and *Customers* with a 38.89% appearance rate. On the other hand, the least common actor is *Services*, included only in 5 organizational models.

Second, we proceeded to unify all the actors duplicated in the models, so that we obtained a set of instances for each internal and external actor. Table 2 shows an excerpt of actors identified as *Supplier* instances and their occurrences in the 36 organizations. We can see instances like *Basic services supplier, Technology supplier*, etc. Summarizing, from the 886 external actors, only 203 were unique instances and from the 223 internal actors, only 77 were unique instances.

Third, after getting the instances, we realized that some of them had common characteristics, being able to group them into categories, obtaining a hierarchical structure; the categories are called dimensions and are composed by the specific instances found in the previous step. As an example, consider actors categorized under the *Supplier* generic actor, which defines three dimensions (see Table 3): *Type* (Goods, Hardware, Basic services, Technology, etc.); *Volume* (Wholesaler and Retailer); and *Location* (Local, International and National).

3.3 Dependencies Analysis

The analysis of dependencies was focused on their description and type in order to define a catalogue of dependencies. This process was performed in two steps, as described below.

First, similarly to actors' identification, the CM corresponding to the 36 organizations were integrated into a single data space and the dependencies of each organization were analyzed based on their type (goal, softgoals, resources and tasks). Results are shown in Table 4. A total of 2095 dependencies were found (1351 dependencies related to external actors and 744 dependencies related to internal actors);

Table 1. Actors identified per organization. Organization size (S – Small, M – Medium and L - Large). Domain classification (M - Manufacturing, E - Education, T - Transportation, H - Human Health, W - Wholesale and F - Financial activities)

Organization	External Context Actor											Internal Context Actor										
	Id	Size	Domain	Suppliers	Consumers	Control Agencies	Regulatory Agencies	Distributors	Strategic Partners	Competitors	Financial Institutions	Total	Inbound logistics	Outbound logistics	Operations	Services	Firm Infrastructure	Technological development	Procurement	Human Resource	Marketing and sales	Total
Org1	S	M	9	3	7	3	0	0	0	0	22	1	1	1	0	1	0	1	1	1	1	7
Org2	S	M	5	3	7	3	1	2	3	0	24	1	0	1	0	1	0	0	1	2	6	
Org3	S	M	3	5	7	5	2	3	0	1	26	1	1	1	0	0	0	0	0	1	4	
Org4	S	M	8	5	6	3	2	1	0	1	26	1	1	1	1	0	1	0	1	2	8	
Org5	S	W	6	7	2	3	1	2	1	1	23	0	0	0	0	0	0	0	0	0	0	
Org6	S	W	5	6	2	2	1	1	0	1	18	1	1	1	1	0	0	0	0	1	5	
Org7	S	W	4	7	3	2	1	0	1	0	18	1	1	0	0	0	0	0	1	2	5	
Org8	S	E	4	4	3	2	1	2	0	1	17	0	0	0	0	0	0	0	0	0	0	
Org9	S	W	6	7	4	3	2	1	2	2	27	1	1	1	1	0	0	0	1	2	7	
Org10	S	W	3	5	4	4	0	0	0	0	16	1	0	0	0	1	1	0	0	1	4	
Org11	S	W	10	12	7	3	4	3	2	3	44	1	0	1	0	2	0	0	1	1	6	
Org12	S	W	4	3	2	1	2	0	0	1	13	1	1	1	1	0	0	0	1	2	7	
Org13	S	M	3	6	5	2	3	0	3	2	24	1	0	1	0	1	0	0	0	1	4	
Org14	S	W	8	5	6	3	2	1	0	1	26	1	0	1	0	2	0	0	0	1	5	
Org15	M	W	8	9	7	3	3	4	2	2	38	1	1	0	0	2	0	1	1	3	9	
Org16	M	W	7	5	6	3	2	1	0	0	24	2	1	0	0	2	0	2	0	3	10	
Org17	S	W	7	5	3	4	2	2	1	1	25	0	0	1	0	2	0	2	1	1	7	
Org18	S	M	5	7	2	2	1	1	1	1	20	1	1	1	0	0	0	0	1	2	6	
Org19	S	W	9	3	7	3	0	0	0	0	22	0	0	0	0	1	0	1	0	1	3	
Org20	S	W	8	5	6	3	2	1	0	1	26	1	1	1	0	1	0	0	1	2	7	
Org21	M	E	9	8	6	3	4	4	2	2	38	0	0	0	0	2	3	0	0	0	5	
Org22	S	F	8	3	7	3	0	0	0	1	22	0	0	1	0	3	1	0	1	2	8	
Org23	S	W	6	5	5	2	3	0	3	1	25	1	0	2	0	1	1	0	1	1	7	
Org24	S	W	8	3	7	3	0	0	0	0	21	0	0	1	0	1	0	0	0	1	3	
Org25	S	M	7	5	6	3	3	4	2	2	32	1	0	1	0	2	0	0	1	1	6	
Org26	S	E	5	6	2	3	1	2	1	1	21	0	0	3	0	1	0	0	0	2	6	
Org27	L	M	7	7	5	3	4	4	2	1	33	1	1	1	0	4	0	1	1	1	10	
Org28	S	E	9	8	6	3	4	4	2	2	38	0	0	3	0	1	0	0	1	0	5	
Org29	S	M	6	6	2	2	1	0	1	0	18	1	1	8	0	2	0	0	1	1	14	
Org30	S	M	8	5	6	3	2	1	0	1	26	1	1	2	0	1	0	0	0	1	2	8
Org31	S	W	5	4	2	1	2	0	1	1	16	1	0	0	0	4	1	0	1	0	7	
Org32	M	M	8	3	7	3	0	0	0	0	21	1	0	0	0	3	0	0	0	0	4	
Org33	M	H	5	4	2	1	2	0	0	0	14	1	0	3	0	0	0	0	0	1	5	
Org34	M	H	7	7	5	3	4	4	2	1	33	1	0	2	1	2	1	1	1	4	13	
Org35	L	M	6	7	6	3	2	1	1	0	26	2	0	2	0	3	1	1	1	2	12	
Org36	L	T	8	5	6	3	0	0	1	0	23	0	0	0	0	0	0	0	0	0	0	
Total			234	198	176	99	64	49	34	32	886	28	14	42	5	46	10	10	21	47	223	

from them, 862 dependencies are goals, 537 softgoals, 619 resources and 77 tasks. The scarce use of task dependencies is probably due to their high level of prescriptiveness, which is something that does not match well with the activity of context modeling.

Table 2. Excerpt of actors identified and their occurrence in the 36 cases conducted

Generic Actor	Actor	Org1	Org2	Org3	Org4	Org5	Org6	Org7	Org8	Org9	Org10	Org11	Org12	Org13	Org14	Org15	Org16	Org17	Org18	Org19	Org20	Org21	Org22	Org23	Org24	Org25	Org26	Org27	Org28	Org29	Org30	Org31	Org32	Org33	Org34	Org35	Org36	Total	Percentage					
		Supplier	Hardware									X	X	X	X							X																				3	8%	
Basic services																																									14	39%		
Technology							X															X																			1	3%		
Transport									X			X																														3	8%	
Wholesale											X																															1	3%	
Local	X			X	X					X		X			X								X																				7	19%
National				X	X			X		X				X	X								X																X	X		9	25%	
International								X		X					X			X																X					X			7	19%	

Table 3. Dimensions found for the *Supplier* generic actor.

Generic actor	Dimension	Actor instances
Supplier	Type	Goods
		Hardware
		Basic services
		Technology
		...
		Transport
	Volume	Wholesaler
		Retailer
	Location	Local
		National
		International

Table 4. Dependencies identified per organization

Type	Org1	Org2	Org3	Org4	Org5	Org6	Org7	Org8	Org9	Org10	Org11	Org12	Org13	Org14	Org15	Org16	Org17	Org18	Org19	Org20	Org21	Org22	Org23	Org24	Org25	Org26	Org27	Org28	Org29	Org30	Org31	Org32	Org33	Org34	Org35	Org36	Total	
Goal	16	31	22	36	15	39	26	19	39	14	29	25	15	29	29	32	16	14	16	9	40	30	28	29	17	24	46	20	30	28	13	16	10	9	15	36	862	
Softgoal	11	10	9	19	11	12	2	4	18	6	15	22	8	19	23	15	10	21	11	9	19	18	21	17	13	14	32	22	23	33	4	7	7	10	22	20	537	
Resource	18	12	0	26	10	10	2	13	22	5	15	18	14	25	22	32	16	24	13	11	21	11	24	39	15	27	32	12	21	35	8	5	3	14	26	18	619	
Task	0	0	0	1	0	2	0	0	7	2	0	2	4	2	4	6	0	0	1	2	1	4	3	1	1	7	3	1	7	8	1	0	0	0	0	5	2	77
Total	45	53	31	82	36	63	30	36	86	27	59	67	41	75	78	85	42	59	41	31	81	63	76	86	46	72	113	55	81	104	26	28	20	33	68	76	2095	

Second, we proceeded to group the 2095 dependencies with their respective generic actors. In this step, the duplicated dependencies were omitted, finding 994 dependencies (408 dependencies linked to external actors and 586 to internal actors). Summing up, from the 862 goal dependencies identified in the previous step (Table 4), 449 are unique instances; from the 537 softgoals, 249 are unique instances; from the 619 resources, 254 are unique instances; and from the 77 tasks, 42 are unique instances.

3.4 Synonyms in Actors and Dependencies

During the data analysis, we found some actors and dependencies representing the same entity or idea but written differently, that is, synonyms. An example is the occurrence of two actors, *Final client* and *Final customer*, in two different organizations; for dependencies, consider the dependencies *Timely delivery* and *On-time delivery* found in two different organizations. In summary, we found 13 synonyms in external actors, 51 in dependencies linked to external actors, 32 synonyms in internal actors and 86 in dependencies linked to internal actors. For dependencies, 68 were goals, 48 softgoals and 21 resources. To simplify the catalogue, we decided to create a section focused in those findings. The main idea is that in later stages, through the use of semantic technologies, analyze the actors and dependencies entered by the user as part of a CM and inform him that a similar actor or dependency has been previously defined with different words (if that is the case), and let him decide which of them is the best option.

Figure 2 shows graphically how the total number of actors and dependencies shrank little by little after each step described in Sects. 3.2 and 3.3, including the results of synonyms, obtaining a total of 235 actors, from them 190 are external context actors and 45 internal context actors. Additionally, 857 dependencies were identified (381 goals, 201 softgoals, 233 resources and 42 tasks).

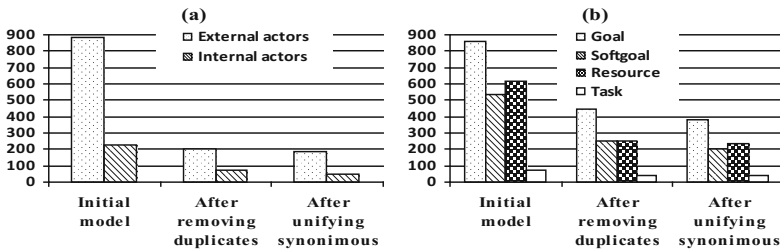


Fig. 2. Number of elements shrunk on each step. (a) Actors, (b) Dependencies

3.5 Parametric Actors and Dependencies

Even after the consolidation of actors and dependencies as explained in the previous subsections, we found groups of actors or dependencies identified in different organizations but all those sharing similar characteristics. To make explicit this similarity and also to make the catalogue more compact, we incorporated parameters to the definition of actors and dependencies. As an example of parametric actors, consider an organization where the actor *Primary Students* has been identified, and a second organization where the actor *High School Students* emerged, both of them with similar relationships in their respective organization. That allowed us to group them in one category, with the possibility to parametrize them, that is, the parametric actor is

defined as *<type-of-student> Students*, where the parameter *<type-of-student>* can be instantiated as *Primary* or *High school*. Other possible case of parametrization can be found with actors sharing characteristics as the sector, but differing in the industry, it means, the parametric actor could be *Supplier of <services>* and the tag *<services>* can be parametrized as *basic services, telecommunications, security, etc.*

The same occurs in dependencies. For instance, let's consider the dependency *<Products> acquired*; the parameter *<Products>* can be replaced by *furniture, clothes, equipment, etc.*, according to the industry of the depender or dependee. Sometimes, parametric dependencies can be associated to parametric actors, for example, the parametric actor *Supplier of <services>* (mentioned in the paragraph above) is associated to the parametric dependency *<Specific documents>*, if the actor is instantiated as *Supplier of transport services*, the dependency has to be instantiated as *Waybill* or if the actor is *Supplier of medical services*, the dependency has to be instantiated as *Medical record*.

A total of 41 actors in the catalogue were identified as parametric (22 external and 19 internal) and 63 dependencies (35 dependencies linked to external actors and 28 to internal actors), where 25 are goals, 6 softgoals, 28 resources and 4 tasks.

4 The Catalogue of Context Model Elements

After organizing the data and identifying unique instances, parameters and synonyms of actors and dependencies, we provided structure to the catalogue. It was organized into two sections, one for actors and a second one for dependencies. The actors' section has a total of 235 instances, from which 190 are external context actors and 45 internal context actors, structured in 4 hierarchical levels as explained below:

- **Fist level:** composed by the 17 Porter generic context actors, 8 external and 9 internal, introduced in Sect. 3.2.
- **Second level:** Each internal and external actor is decomposed into subactors (defined as “Dimensions” in Table 3). From them, 17 are dimensions of external context actors, and 7 are dimensions of internal context actors.
- **Third level:** Contains a total of 39 instances on external context actors and 9 instances on internal context actors (see column “Actor instances” in Table 3).
- **Fourth level:** This level contains the parameters that can be used as instances of parametric actors defined in the third level and has 190 external context actors and 45 internal context actors.

From the point of view of the dependencies, the catalogue contains a total of 857 dependencies, from which 381 are goals (126 dependencies linked to external actors and 255 to internal actors); 201 softgoals (107 dependencies linked to external actors and 94 to internal actors); 233 resources (109 dependencies linked to external actors and 124 to internal actors); and 42 tasks (15 dependencies linked to external actors and 27 to internal actors).

Table 5. Catalogue excerpt

Generic actor	Dimension	Actor instances	Dependency	Type	Direction	
Customer	Frequency or volume	Potential	Widespread promotions	Goal	>	
			Promotional samples	Resource	<	
		New	Membership card provided	Goal	>	
			Special introduction prices provided	Softgoal	>	
			Membership card	Resource	>	
			Personal information registered	Goal	<	
		Important	VIP benefits granted	Goal	>	
			Personalized attention	Softgoal	>	
			VIP card	Resource	>	
			Important high volume order placed	Goal	<	
		Distribution channel	Wholesaler	Product availability guaranteed	Goal	<
				Product distribution agreement signed	Softgoal	<
	Increase sales through the distribution chain			Softgoal	<	
	Product distribution agreement			Resource	<	
	Product distribution chain achieved			Softgoal	>	
	Retailer		Restocking in small quantities provided	Goal	>	
			Approach consumers through a specific location	Softgoal	<	
			Increase sales through individual stores	Softgoal	<	
	Specificmarket segment		Specialized customer service infrastructure	Softgoal	>	
			Trained staff for specific needs	Softgoal	>	
			Specific documents	Resource	>	
	Payment method		Credit	Deferred payments	Goal	>
		Credit flexibility		Softgoal	>	
		Acceptance of various credit cards		Softgoal	>	
		Voucher		Resource	>	
		Warranty documents		Resource	<	
		Cash	Cash rebates	Goal	>	
Money			Resource	<		
Supplier	Type	Goods	Health service obtained	Goal	>	
		Services	Home service	Goal	>	
	Volume	Wholesaler	Large purchase orders	Goal	>	
		Retailer	Restocking in small quantities provided	Goal	<	
	Location	Local	Cash payment discounts	Goal	>	
		National	Deferred payments	Goal	>	
		International				

The catalogue (currently in Spanish only) can be accessed at a given URL address¹. Table 5 shows an excerpt of the catalogue, presenting the dimension, actors and dependencies identified for the *Customer* and *Supplier* actors.

5 Catalogue Use

In this section, we show a use case where the catalogue of CM elements was applied, the example shows how the catalogue was applied over a small organization which included some of the elements listed in the catalogue (see Sect. 5.1) and Sect. 5.2 shows some statistics about the reusability level achieved when applying the catalogue.

5.1 Use Case

To validate the catalogue, we proceeded to model a small organization of the academic sector that develops scientific, professional and technical activities. The modeling process was performed in 4 phases, where, each phase was modeled using the catalogue:

1. *Identification of Internal Actors*: We performed an interview with the manager of the organization, asking him to check the internal actors from the catalogue that represent departmental areas in the organization. This step exposes all the actors inside the organization.
2. *Identification of External Actors*: Each internal actor identified in Step 1 was interviewed and asked to identify the external actors from the catalogue with which it interacts in a daily basis.
3. *Establishing dependencies between Actors*: Once identified both, external and internal actors, we checked the dependencies from the catalogue that included each pair of actors, obtaining a set of dependencies validated by the internal actor being interviewed.
4. *Constructing the i^* diagrams*: we generated a set of i^* diagrams that included actors and dependencies identified for the organization. The i^* diagrams resulting were validated with the manager of the organization.

The results obtained in this use case contain a total of 39 actors (33 external and 6 internal context actors) and 185 dependencies (101 goals, 34 softgoals, 44 resources, and 6 tasks). In each phase were identified new elements (actors and dependencies), this elements were added to the catalogue. Regarding actors, 33 out of 39 are included in the catalogue (30 external and 3 internal) and 6 actors were identified during the interviews (3 external and 3 internal). Regarding dependencies, 176 out of 185 dependencies were included in the catalogue (97 goals, 34 softgoals, 42 resources and 3 tasks) and 9 dependencies were identified during the interviews (4 goals, 2 resources and 3 tasks). Figure 3 show the catalogue usage statistics.

¹ https://www.dropbox.com/sh/7jnws7vqwwhnv8/AADqaiHx_vDj-gi6mk_5pg_Ca?dl=0.

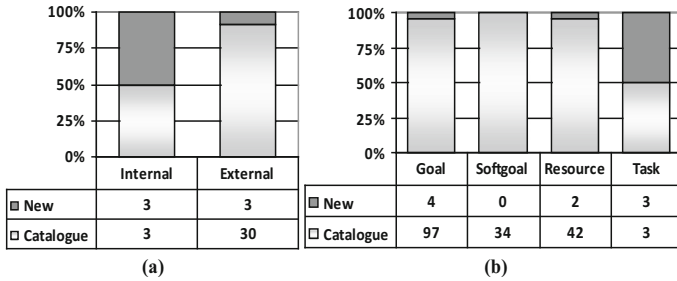


Fig. 3. Use of the catalogue

The use of the catalogue in this use case facilitated the identification of actors and dependencies through the reuse of elements already identified, where the consultants selected the elements of the catalogue that matched those of the organization. In this way, it allowed to streamline and create stable models taking advantage of the knowledge base, compared to the time it would take to model from scratch.

5.2 Reusability

As stated by Porter [6], organizations share common elements in their context (e.g. external actors derived from the model of market forces and internal actors derived from the value chain) it can be considered as fact that some activities performed by these actors share commonalities, this is the reason why we have decided to analyze many academic and industrial cases. The analysis leads us to confirm our beliefs, many global activities were the same among different organizations. However, as we drilled down to more specific industry segments, more specific elements started to emerge, which, at some point, required an exhaustive analysis about including them or not in the catalogue, at the end, we decided to create parametric actors and dependencies. Although the construction of the catalogue was very laborious, the final result is satisfactory since after making a first use (explained in previous section) we noticed that its application helps to create CM with the following characteristics:

1. Due to the wide range of industry segments of the organizations analyzed in this work, we consider that the catalogue is representative, almost complete, and the consultant can select elements from a wide variety of possibilities.
2. As actors and dependencies are categorized and include definitions of type, it is easier for the consultants to clarify concepts regarding to types and hierarchies.
3. When performing the interview with the responsible of each organization area, the consultant will have a solid knowledge base that will help him to understand the organizational context without starting from scratch and reusing elements identified in the catalogue.

6 Conclusions and Future Work

In this work, we have presented the study performed in over 36 industrial IS architectural in which junior consultants used the *i** language, in particular Strategic Dependency models (SD), to build organizational context models. The models were analyzed, classified and organized in order to get a catalogue of reusable context model elements to serve as a basis for the construction of *i** SD-based CM in future case studies. The results obtained in this study are important; the catalogue contains a total of 235 actors (190 external context actors and 45 internal context actor) and 857 dependencies (381 are dependencies of type goal, 201 are dependencies of type soft-goal, 233 dependencies of type resource, and 42 dependencies of type task). Also, we validated the catalogue through its application in a small organization, finding that most of the elements included in the final CM (>80%) are contained in our catalogue.

Work in progress includes the further refinement of the existing catalogue; we plan to conduct a similar study focused on *i** SR models, which are used in later phases of the DHARMA method, analyzing goal decomposition, means-end links, etc. Also, it is worth mentioning that we have created an ontology network [16] to add semantics to our catalogue and we also have developed a tool to support the application of the catalogue [17]; the tool makes easier the application of the catalogue, filtering its elements and generating *i** models automatically; nevertheless, we aim to improve the ontology and the tool to support the construction of CM avoiding synonyms in actors and dependencies.

References

1. Pohl, K.: Requirements Engineering: Fundamentals, Principles, and Techniques. Springer, Heidelberg (2010)
2. Yu, E.: Modelling strategic relationships for process reengineering. Ph.D. thesis, University of Toronto, Department of Computer Science, Canada (1995)
3. Carvallo, J.P., Franch, X.: Descubriendo la arquitectura de sistemas de software híbridos: un enfoque basado en Modelos *i**. In: WER (2009)
4. Carvallo, J.P., Franch, X.: Building strategic enterprise context models with *i**: a pattern-based approach. In: Aier, S., Ekstedt, M., Matthes, F., Proper, E., Sanz, Jorge L. (eds.) PRET/TEAR -2012. LNBIP, vol. 131, pp. 40–59. Springer, Heidelberg (2012). doi:10.1007/978-3-642-34163-2_3
5. Abad, K., Pérez, W., Carvallo, J.P., Franch, X.: *i** in practice: identifying frequent problems in its application. In: ACM SAC (2017)
6. Porter, M.: Competitive Strategy. Free Press, New York (1980)
7. Withall, S.J.: Software Requirement Patterns. Microsoft Press (2007)
8. Supakkul, S., Hill, T., Chung, L.: An NFR pattern approach to dealing with NFR. In: RE (2010)
9. Ruiz-López, T., Garrido, J., Supakkul, S., Chung, L.: A pattern approach to dealing with NFRs in ubiquitous systems. In: CEUR-WS (2013)
10. Renault, S., Méndez, O., Franch, X., Quer, C.: Constructing and using software requirement patterns. In: RCIS (2009)

11. Strohmaier, M., Horkoff, J., Yu, E., Aranda, J., Easterbrook, S.: Can patterns improve *i** modeling? Two exploratory studies. In: Paech, B., Rolland, C. (eds.) REFSQ 2008. LNCS, vol. 5025, pp. 153–167. Springer, Heidelberg (2008). doi:[10.1007/978-3-540-69062-7_16](https://doi.org/10.1007/978-3-540-69062-7_16)
12. Abad, K., Carvallo, J.P., Peña, C.: *i*Star in practice: on the identification of reusable SD context models elements. In: *i*Star (2015)
13. Salman, I., Misirli, A.T., Juristo, N.: Are students representatives of professionals in software engineering experiments? In: ICSE (2015)
14. Instituto Nacional de Estadísticas y Censos. Ecuador en cifras. <http://aplicaciones2.ecuadorencifras.gob.ec/dashboard2/pagina3.php>
15. Office for Official Publications of the European Communities. NACE Rev 2. Statistical classification of economic activities in the European Community (2008)
16. Pérez, W., Abad, K., Carvallo, J.P., Espinoza, M., Saquicela, V.: Ontología DHARMA para la construcción de Arquitectura de Sistemas Empresariales. *Revista Maskana* 7, 177–185 (2016)
17. Abad, K., Pérez W., Carvallo, J.P.: Managing *i**-based reusable context models elements through a semantic repository. In: *i*Star (2016)