Building Strategic Enterprise Context Models with *i**: A Pattern-Based Approach*

Juan Pablo Carvallo[1] and Xavier Franch[2]

[1] Universidad del Azuay
Av. 24 de Mayo 7-77, Cuenca, Ecuador
jpcarvallo@uazuay.edu.ec
[2] Universitat Politècnica de Catalunya (UPC)
c/Jordi Girona 1-3, 08034 Barcelona, Spain
franch@essi.upc.edu

Abstract. Modern enterprise engineering (EE) requires deep understanding of organizations and their interaction with their context. Because of this, in early phases of the EE process, enterprise context models are often built and used to reason about organizational needs with respects to actors in their context and vice versa. However, far from simple, this task is usually cumbersome because of knowledge and communication gaps among technical personnel performing EE activities and their administrative counterparts. In this paper, we propose the use of strategic patterns expressed with the *i** language aimed to help bridging this gap. Patterns emerged from several industrial applications of our DHARMA method, and synthesize knowledge about common enterprise strategies, e.g. CRM. Patterns have been constructed based on the well-known Porter's model of the 5 market forces and built upon *i** strategic dependency models. In this way technical and administrative knowledge and skills are synthesized in a commonly agreeable framework. The use of patterns is illustrated with an industrial example in the telecom field.

Keywords: enterprise pattern, enterprise context model, market forces, strategic dependencies, *i** framework, iStar.

1 Introduction

Modern enterprises largely rely on information systems specifically designed to manage the continuously increasing complexity of interactions with actors in their context. Architecting such systems requires deep understanding of the enterprise context and strategies. Because of this, early phases of the enterprise engineering (EE) process are usually oriented to model the enterprise context. Enterprise context models, as part of a wider enterprise architecture model, include environmental actors (i.e., actors in the context of an enterprise that interact with it) and descriptions of the relationships among them. Resulting models help understanding the purpose of enterprises on their

* This work has been partially supported by the Spanish project TIN2010-19130-C02-01.

S. Aier et al. (Eds.): TEAR 2012 and PRET 2012, LNBIP 131, pp. 40–59, 2012.

environment, what is required from them, and to reason about the way in which they will respond to the specific needs generating value.

Enterprise context models are therefore a fundamental piece that helps enterprise decision-makers to design and refine their business strategies and enterprise architects to understand what will be required from the resulting socio-technical system. However, far from easy, the construction of such models is usually a cumbersome task, mainly due to:

- Communication gaps among technical personnel, who usually lacks knowledge about business strategies, modeling, planning, and administration skills; and their administrative counterparts, with similar limitation in relation to methodological business processes and requirements elicitation, and systems modeling techniques.
- Limited knowledge of the enterprise structure, operations and strategy, which forces technical staff to spend important amounts of time studying and understanding business, to be reconciled with time constraints resulting from internal and external pressures and narrow windows of opportunity, which increases the risk of misunderstandings or misinterpretations.

One strategy to mitigate this situation is to foster knowledge reuse, designing some artifacts that may be used as templates for both technical and managerial personnel in order to improve understanding. In this paper we propose the use of patterns with this purpose. A pattern has been defined as a solution to a recurring problem in a particular context [1]. Patterns collect relevant knowledge that appears consistently throughout several similar experiences, which has been systematized and stored in an appropriated structure for its future use in analogous settings.

In our particular case, we are interested in storing enterprise context knowledge, which has been identified and systematized in the domain of business analysis and strategy. In order to make it accessible to enterprise architects, we represent this knowledge by means *i* strategic dependency* (SD) models [2]. Therefore, patterns include environmental actors and their strategic dependencies. Patterns emerged from several industrial and academic applications of the first activity of our DHARMA method [3], which requires the construction of *i**-based context models.

For building the catalogue of patterns, we distinguish two levels of abstraction. At the highest level, we make an analysis of enterprise behavior guided by the theory and the elements described in Porter's model of the 5 market forces [4]. This analysis is applicable in general to any kind of enterprise. At the lowest level, we consider enterprise strategies which describe how a particular enterprise operates, e.g., Enterprise Resource Planning (ERP), Customer Relationship Management (CRM) and Supply Chain Management (SCM) [5].

2 Background

2.1 Porter's Model of the Five Market Forces

Porter's model of the five market forces is designed to help organizations analyze the influence of 5 forces on their business and to reason about the strategies potentially available to make them profitable. Although subject of some criticism from its retractors, Porter's model remains wildly accepted, studied and used in the practice [6].

According to Porter [4], "the essence of formulating competitive strategy is relating a company to its environment". Although the environment is very broad encompassing social and economic forces, from the business point of view a key aspect in the environment of an enterprise is the industry or industries in which it competes. The state of competition in the industry depends on five competitive forces (see Fig. 1): threat of new entrants; threat of substitution; bargain power of customers; bargain power of suppliers; and rivalry among current competitors. Lately, some authors have proposed a sixth force, the government, not only because of its regulatory power, but also because it may become a potential competitor in some industries e.g. public vs. private schools. We do not consider this force in this paper.

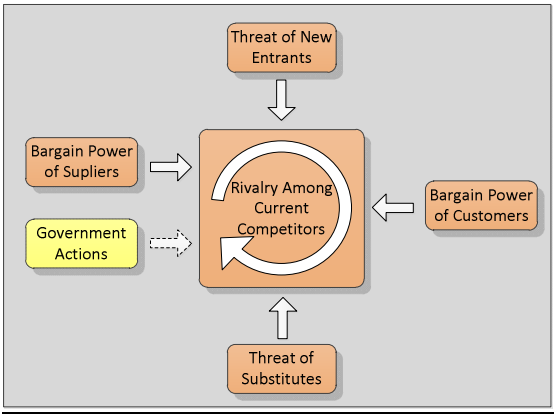

Fig. 1. Porter's model of the five market forces

2.2 The *i** Framework

The *i** framework [2] was formulated for representing, modeling and reasoning about socio-technical systems. Its modeling language is constituted basically by a set of graphic constructs which can be used in two models: the Strategic Dependency (SD) model, which allows the representation of organizational actors, and the Strategic Rationale (SR) model, which represents the internal actors' rationale. Since our patterns are defined as SD models, we focus the explanation on SD constructs.

Actors in SD models are classified in DHARMA as human, organizational, software or hardware. They can be related by *is-a* (subtyping) relationships and may have social dependencies. A *dependency* is a relationship among two actors, one of them, named *depender*, who depends for the accomplishment of some internal intention from a second actor, named *dependee*. The dependency is then characterized by an *intentional element* (*dependum*) which represents the dependency's element. The primary intentional elements are: *resource*, *task*, *goal* and *softgoal*. A softgoal represents a goal that can be partially satisfied, or a goal that requires additional agreement about how it is satisfied. They are usually introduced for representing non-functional requirements and quality concerns. Fig. 2 presents an *i** SD model in the Ecuatorian Etapatelecom company, introduced in Section 6.1.

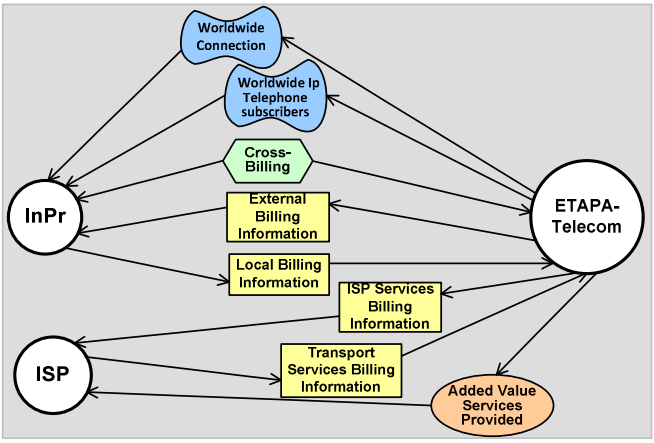

Fig. 2. An *i** SD model for the Etapatelecom case

2.3 The DHARMA Method

The DHARMA method has been used in the context of this work. It aims at the definition of enterprise architectures using the *i** framework [3]. The process resulting from the method (see Fig. 3) is initiated by modeling the enterprise context (1), then introducing the system in the organization (2), analysing its impact in detail (3) and identifying its generic enterprise architecture (4), i.e. actors that form the system, the services that must be covered by each of them and the relationships among them. It is clear, thus, that improving the initial construction of the enterprise context model would be of great help for the method applicability.

2.4 Patterns in the *i** Framework

Several approaches about the definition and use of patterns in *i** have been proposed. Among them, the closest proposals to ours are the works on social structures presented in [7][8], where the authors propose a set of social patterns, drawn from research on cooperative and distributed architectures. Several differences with our approach:

- Formalization. We will provide a formal definition of pattern instantiation and a couple of metrics to measure coverage.
- Size. In their work, patterns are intended to model different types of cooperation settings among organizations, e.g., Structure-in-5 and Joint Venture. In our approach, patterns are much more detailed and intended to model the context of particular organization instead of the relation among groups of them.
- Background. Our approach is based on theory of business administration and marked strategy whilst these approaches are based on organizational theory.

The aim of these works is to propose ontology for information systems, inspired by social and organizational structures. Our work is intended to provide guidance in early phases of the EE process, providing artifacts to bridge communication gaps among technical EE staff and administrative staff.

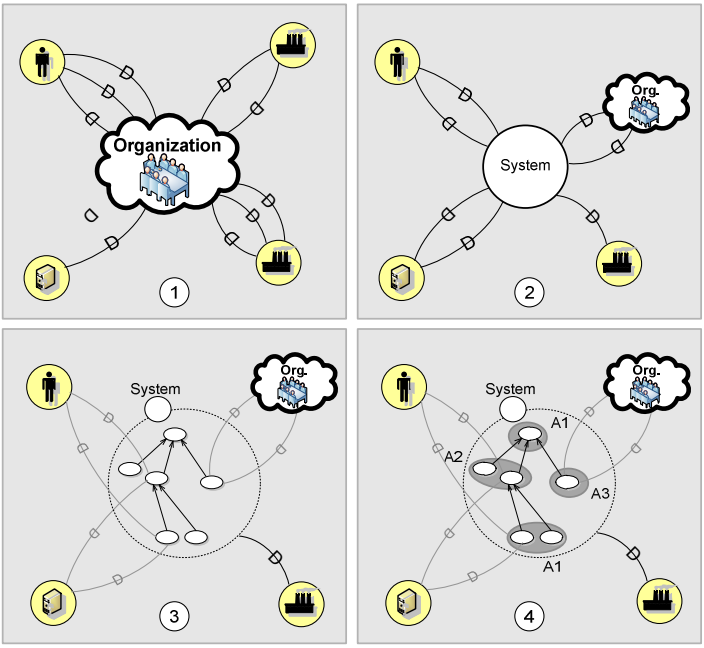

Fig. 3. The DHARMA method

3 Formal Definition of Patterns

Being aware that Alexander's definition of pattern includes several parts that may play a fundamental part in a reuse strategy, in this paper we focus on the expression of the solution part expressed as an *i** SD model. Although cumbersome, we consider this formalization a necessary step in order to be able to rigorously define the patterns and make possible some future work on analysis techniques and measures' definition.

3.1 *i** SD Models

Definition. *i** SD Model.

An *i** SD model M is a tuple M = (A, D, L) being A the set of actors, D the set of dependencies and L the set of actor specialization links.

Definition. Set of actors.

Given the *i** SD model M = (A, D, L), the set of actors A = $\{A_i\}$ is such that each A_i is a pair (name, type), with type = (Human, Organizational, Software, Hardware).

Definition. Set of dependencies.

Given the *i** SD model M = (A, D, L), the set of dependencies D (over A) is a set D = $\{d_i\}$ such that each d_i is a tuple (name, dpr, dpe, type), with dpr, dpe$\in$ A such that dpr $\neq$ dpe, and type = (Goal, Softgoal, Task, Resource).

Definition. Set of actor specialization links.

Given the *i** SD model M = (A, D, L), the set of actor specialization links L is a set L = $\{l_i\}$ such that each l_i is a pair (superactor, subactor), with superactor, subactor∈ A such that superactor ≠ subactor (in fact, cycles are not allowed).

We remark again that these definitions present some simplifications over the complete definition of the *i** SD models as available e.g. in the *i** wiki [9], but the concepts introduced here are enough for the patterns identified so far. As mentioned above, these changes align with previous work on the use of *i** in industrial settings (see e.g. [3]) which points to the fact that in practice some constructs create some confusion and act as an adoption barrier for practitioners. Also, as we will see in the next sections, we adopt three graphical conventions: first, we allow for a tabular representation of *i** models, especially useful when models grow; second, we represent direction of dependencies by arrowheads instead of the "D" convention of *i**; third, given two dependencies d1 = (n, a1, a2, t) and d2 = (n, a2, a1, t), we can draw these two dependencies using just a single graphical dependency with arrowheads in both directions. Bidirectional dependencies are useful to express mutual collaboration in actors.

3.2 Patterns and Their Application

In our proposal a context model patterns is nothing else than a plain *i** SD model defined as in Section 3.1.

Definition. Context model pattern.

Any *i** SD model M is a context model pattern.

In Fig. 4 we show an excerpt of the context model pattern for our CRM case study, that will be introduced in more detail in Section 5.

The key concept around patterns is that of instantiation. Instantiation is defined upon the notion of model correspondence, that defines how actors and dependencies of a pattern are assigned to other actors and dependencies that are defined on a different space (i.e., the actors and dependencies of the enterprise, *ent*).

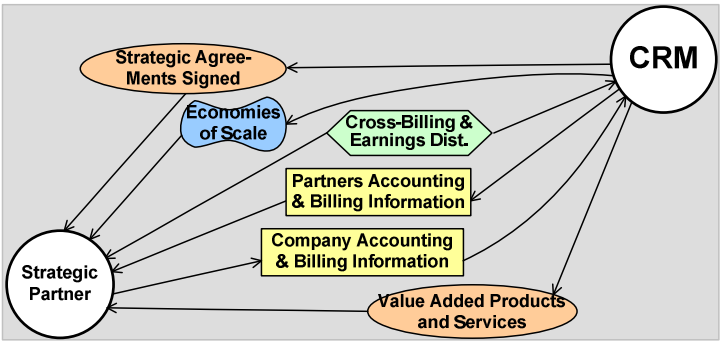

Fig. 4. Example of context model pattern (excerpt)

Definition. Model correspondence. Domain, rank.

Given the *i** SD model M = (A, D, L), and given the sets A_{ent}, D_{ent} and L_{ent}, being A_{ent} a set of actors, D_{ent} a set of dependencies (over A_{ent}) and L_{ent} a set of actor specialization links, $A_{ent} \cap A = \emptyset$ (thus $D_{ent} \cap D = \emptyset$), a model correspondence C from M to (A_{ent}, D_{ent}, L_{ent}) is a pair of correspondences, $C = (c_A, c_D)$, $c_A \subseteq (A \times A_{ent})$ and $c_D \subseteq (D \times D_{ent})$. Correctness conditions are:

- Actors appearing in dependencies must be aligned:
  $((n, a1, a2, t), (m, b1, b2, w)) \in c_D \Rightarrow (a1, b1) \in c_A \wedge (a2, b2) \in c_A)$
- If a subactor is part of the correspondence, its superactor must be too:
  $(a, b) \in c_A \Rightarrow (\forall x: (x, a) \in L: (\exists y: (x, y) \in c_A))$

Given a correspondence $c \subseteq (X \times Y)$, we define its domain and rank as:

- $dom(c) = \{x \in X: (\exists y \in Y: (x, y) \in c)$
- $rnk(c) = \{y \in Y: (\exists x \in X: (x, y) \in c)$

The correspondence is not a function, since every pattern's actor and dependency can be assigned to a zero, one or several actors and dependencies of the enterprise model.

The instantiation of a pattern consists of an application of that pattern to the model's actors and dependencies in order to create a new *i** SD model, the context model of the enterprise. That is, starting from the pattern, the actors and dependencies that are part of the correspondence are substituted by the corresponded elements, whilst actor links from the pattern are correctly preserved and new actor links are added.

Definition. Pattern instantiation. Enterprise context model.

Given an *i** SD model M = (A, D, L) acting as context model pattern, and given a model correspondence $C = (c_A, c_D)$ as above, we define the pattern instantiation of M by C as an SD model M_{inst}, the enterprise context model, $M_{inst} = (A_{inst}, D_{inst}, L_{inst})$, as:

$A_{inst} = rnk(c_A)$, $D_{inst} = rnk(c_D)$, $L_{inst} = corresponded(L, c_A) \cup L_{ent}$

where *corresponded* maps actors appearing in actor links according to the correspondence defined among actors:

$corresponded(L, c_A) = \{(x, y): (a, x) \in A \wedge (b, y) \in A \wedge (a, b) \in L\}$

Fig. 5 shows an example of instantiation involving the pattern presented in Fig. 4 and the organization model presented in Fig. 2. Pattern elements are shown semi-transparent. We can see, e.g., the pattern actor CRM instantiated into one actor the Etapatelecom company (TC); another pattern actor Strategic Partner instantiated into two actors InPr and ISP; and then several pattern dependencies instantiated, e.g. Cross -Billing into another with the same name, or Economies of Scale into two other dependencies. We remark also that a dependency from the pattern, Strategic Agreements Signed, is not instantiated. The final model is composed of the colored elements.

Two indicators may help to classify the adequacy of the patterns (pattern coverage) and the extent to which an organization adheres to known patterns (model coverage).

Definition. Pattern coverage.

Given an *i** SD model M = (A, D, L) acting as context model pattern, and given a model correspondence $C = (c_A, c_D)$, we define the coverage of M under C, cov(M, C) as the percentage of elements of M that have a correspondence defined in C:

$cov_{pat}(M, c) = (||dom(c_A)|| + ||dom(c_D)||) / (||A||+||D||) * 100$, $||S||$ being the size of S.

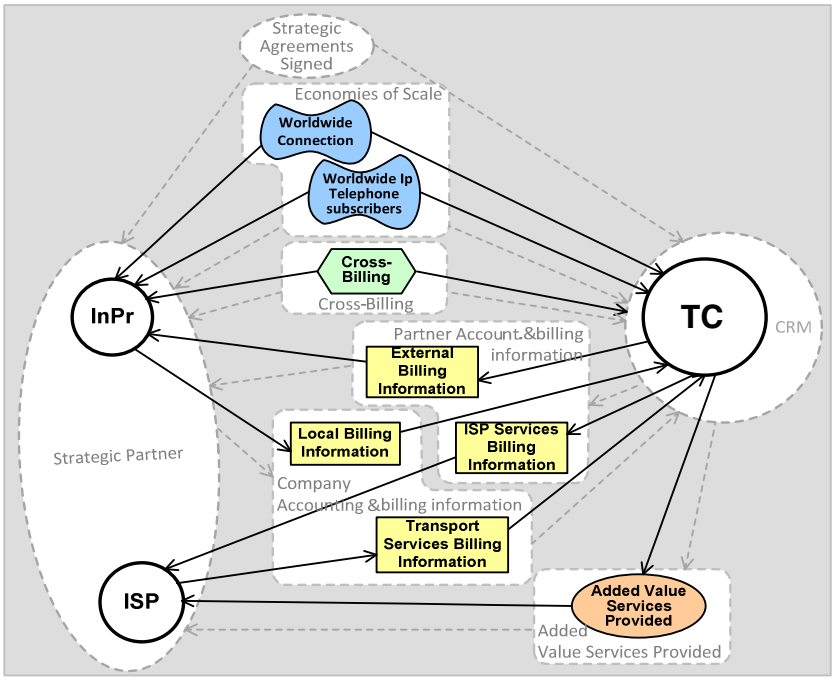

Fig. 5. Example of pattern instantiation

Definition. Model coverage.

Given an *i** SD model $M_{ent} = (A_{ent}, D_{ent}, L_{ent})$ and given a set of *i** SD models $M = \{M_k = (A_k, D_k, L_k)\}$ acting as context model patterns, and given the set of model correspondences over them $C = \{C_k = (c_{A[k]}, c_{D[k]})\}$, such that $c_{A[k]} \subseteq (A_k \times A_{ent})$ and $c_{D[k]} \subseteq (D_k \times D_{ent})$, we define the coverage of M_{ent} under M and C, $cov(M_{ent}, M, C)$, as the percentage of elements of M_{ent} that have a correspondence defined in C:

$$cov_{mod}(M, c) = ||\{a \in A_{ent}: (\exists k: a \in rnk(c_{A[k]}))|| + ||\{d \in A_{ent}: (\exists k: d \in rnk(c_{D[k]}))|| \\ / (||A_{ent}|| + ||D_{ent}||) * 100$$

In the example above, we obtain $cov_{pat}(M, c) = 87{,}5\%$ and $cov_{mode}(M, c) = 100\%$.

4 Patterns for Generic Perspective of Enterprises

Porter's model of the five market forces can be used as basis to construct enterprise context models. Each of the forces has a set of *determinants* associated to them, which describe the way in which various external agents interact with the enterprise. External agents can be modeled as actors in the context of the enterprise and their interaction with the enterprise by means of strategic dependencies, describing the intentionality between them and the enterprise. In the rest of this section, we introduce the forces in detail and propose the *i** SD pattern that describes them. For space reasons, we provide thorough details just in the first force and then an overall description for the rest of forces.

Porter's five forces together determine the intensity of the industry competition and profitability, however, each force has its own prominence in relation to each type of industry according to economical and technical characteristics. For instance, suppliers and availability of raw material may be critical for some industries e.g. cement, or aluminum processing factories, whilst location of customers may be critical for other e.g. fresh flower growers, or manufacturers of perishable agricultural products.

We have identified several context actors in relation to each force in Porter's model (see Fig. 6). They are represented in italics in the next paragraphs, which describe their relation with determinants in each force.

***1st. Force - Potential New Entrants*:** New entrants to an industry bring new capacity, the desire to gain market share and some substantial resources. The main determinants (listed in Porter's work) associated to new entrants are entry barriers:

- *Economies of scale:* Difficulties to enter the market due to higher cost of production or to gain mass production. Companies may associate with *Strategic Partners* to gain access to property technology, reduce learning curves, expand its coverage, and reduce costs of manufacturing and distribution among other.
- *Product Differentiation:* Well established firms benefit from brand identification and customer loyalty. This creates a barrier to entry, forcing entrants to expend to overcome existing customer loyalties. Association with *Strategic Partners* may help to overcome this barrier.
- *Cost Disadvantages Independent of Scale*: Some established firms have cost advantages not replicable by entrants, e.g. favorable access to raw materials. Association with *Strategic Partners* is a way to overcome cost disadvantages e.g. by providing access to raw materials.
- *Capital Requirements*: Need of *financial* resources to compete in the market, e.g. for covering start-up losses. To satisfy capital requirements 3 actors have been identified: *Owners*, *Financial Institutions* and *Shareholders*.

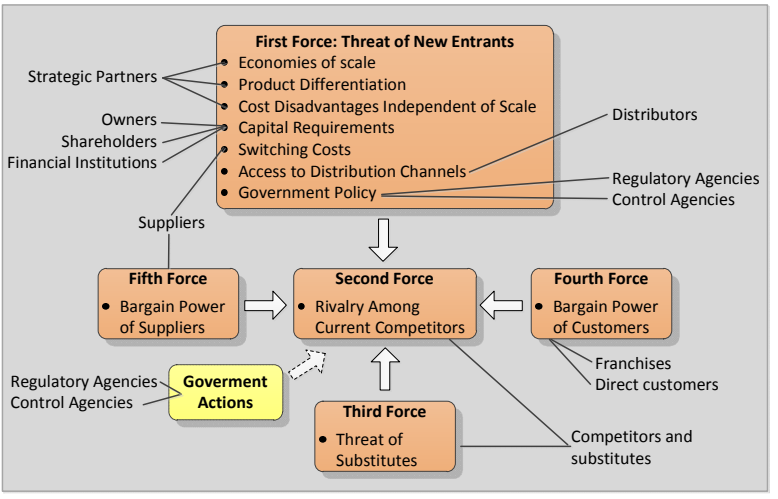

Fig. 6. Generic context actors in relation to the five market forces

- *Government Policy*. Government regulations may limit entry of new competitors, by e.g. requiring licenses or limiting access to some raw materials. We identified two actors in the environment: *Regulatory Agencies* and *Control Agencies*.
- *Access to Distribution Channels*: New *entrants* need to secure distribution of their products. In other to do so, they need to persuade channels to accept their products through price breaks and cooperative advertising, which reduces their profits. To access distribution channels companies sought to relate to *Distributors*, responsible for making product or services accessible to customers.

2nd. Force - Industry Competitors: Rivalry among existing competitors consists in the use of tactics to gain a better position in the market. Tactics may include price competition, advertising battles, new products introduction, and increased customer service or warranties. The force of the competition depends of several factors, e.g., number of equally balanced competitors, slow industry growth, etc.

A new actor in the environment has been identified in relation to this force, *Competitors*. They influence company's strategy and the decisions to be taken. Any change in the competition strategy will have an impact in the preferences of the customers; companies must be informed to react accordingly and maintain market share.

3rd Force - Substitutes: Two goods are substitutes if one of them can replace the other under some circumstances, e.g., cable TV can be replaced by internet streaming services if there is enough bandwidth available. In some sectors, products or services can impose price limits if they are considered good substitutes. Substitute availability drives customers to continuously compare offers against changing costs.

In this work, we consider substitutes part of the competition. They provide alternative products and services and influence company's strategy in a similar way than competitors. Thus, the *Competitors* SCA is renamed as *Competitors and Substitutes*.

4th. Force - Customers: Buyers compete with the industry by forcing down prices, bargaining for higher quality or more services, and playing competitors against each other at the expense of industry profitability. Bargain strength of customer's increases under some circumstances, e.g., an important buyer group concentrates or purchases large volumes relative to seller sales.

In relation to the fourth force we identify two actors *Franchises* and *Direct Customers.* These actors are indispensable for the sustainability of a company.

5th. Force - Suppliers: Suppliers can exert bargaining power over an industry by threatening to raise prices or reduce the quality of purchased goods and services. Suppliers bargain strength increases when offered products or services are scarce and buyers generate large demand, bringing them to a weaker negotiation position, particularly when there are not substitutes available or the cost of replacing them is high. On the contrary, bargain power of suppliers reduces when offered goods or services are standard, there are substitutes available or there are several suppliers offering similar products or services.

In relation to this force, we identified the environment actor *Suppliers*. Companies depend on them for the provision of row materials, machinery, means of transport or infrastructure needed for their operation, product manufacture and service provision.

Table 1 describes some dependencies identified in relation to each of these actors, represented in tabular form. The table includes the description of the dependencies,

their type and the direction with respect to the organization. The list of identified dependencies does not pretend to be exhaustive, but rather to reflect the dependencies that repeatedly appeared in several industrial experiences.

5 Enterprise Strategies: The CRM Case

In order to confront and balance market forces and to generate a profit, enterprises adopt several strategies. Not only differentiation and prices are considered, but also operational strategies, which allow them to focus in some market segments whit particular products and services. Well-known strategies, which categorize organizations in relation to the name of their core information systems, include ERP, CRM and SCM. For the sake of brevity, we are going to focus on one particular case, the CRM enterprise strategy. An excerpt of its *i** graphical representation is shown in Fig. 7 and dependencies in Table 2.

The CRM strategy is usually related to service enterprises [10]. These enterprises thrive and because of their importance, they have gained an increasingly prominent position. In pure service enterprises: the commercial function is carried without products; there is not direct human contact with clients; services are not storable; there are only user, operator and way to do; services are tailored to client needs; and the perceived quality is more malleable.

One of the most decisive issues is the treatment of all matters relating to staff, because of its intimate relationship with the process. CRM enterprises must be very careful whit personnel selection and training, but also of the conception of the services and the technologies required to support them.

Globalization, diversification and deregulation have increased competition. Today's customers have more options and enterprises must ensure that best clients remain loyal, at the time that new prospects begin to be loyal and profitable. This requires an intensive analysis of the front end applications used to interact with customers, such as billing, order recording and market segmentation, among others.

CRM enterprises usually handle three phases designed to bring them closer to customers; each of them having a different impact in the relationship:

- *Acquisition of new customers*: New customers are attracted by promoting services. Differential aspects shall be made evident for customers to purchase them. The value proposition must be backed up by excellent service and after-sales support.
- *Increase profitability of existing customers*: Improving existing relationship promotes cross-selling of services, increasing sales of services previously acquired by the client and thus profitability. The value of the proposition lies in an offer of convenience and lower cost.
- *Lifetime retention of good customers*: retention focuses on the adaptability of the services by not offering what the general market wants, but what existing customers need. The value of the proposition is to offer a service or product to the best interest of the client through a proactive relationship. Currently the leading companies focus more on retention than the attraction of new clients.

All phases of CRM are interrelated and provide the basis for a new organizational architecture, where business processes revolve around the needs of the client.

Table 1. Generic Context Dependencies

Generic Actor	Dependency Type	Dependency	Direction	Degree of Dependency
Strategic Partners	Goal	Strategic agreements signed	←	Critical
	Resource	Strategic agreements	←	Critical
	Soft Goal	Economies of scale achieved	←	Committed
	Soft Goal	Proprietary technology accessed	←	Committed
	Goal	Business knowledge and training Provided	←	Committed
	Goal	Product differentiated	←	Committed
	Goal	Value added to products and services	→	Critical
	Resource	Company accounting and cross-billing information	←	Critical
	Resource	Partners accounting and cross-billing information	→	Critical
	Task	Cross-billing and earnings distribution	←	Critical
Owners	Goal	Business started	←	Critical
	Resource	Opening capital	←	Critical
	Resource	Performance, accounting and management information	→	Critical
	Goal	Strategic decisions	←	Committed
	Goal	Strategic actions performed	→	Critical
Share-holders	Goal	Shares acquired	←	Critical
	Resource	Shares documents	→	Critical
	Resource	Performance Information	→	Critical
	Resource	Investment capital	←	Critical
	Soft Goal	Profit earned	→	Critical
Financial Institutions	Goal	Financial services provided	←	Committed
	Goal	Loans and mortgages provided	←	Committed
	Soft Goal	Convenient interest rates	←	Committed
	Soft Goal	Adequate payment terms	←	Committed
	Goal	Investments, Savings and checking accounts managed	←	Critical
	Resource	Financial Statements	←	Critical
	Goal	Financial services acquired	→	Critical
	Resource	loans and mortgages information	→	Critical
	Soft Goal	Timely payment	→	Committed
Distributors	Goal	Distribution channels accessed	←	Critical
	Goal	Products and services accessed by customers	←	Critical
	Soft Goal	Locations always supplied	←	Committed
	Resource	Products and services	→	Critical
Regulatory Agencies	Goal	Promulgation of laws and regulations	←	Critical
	Goal	Permits and licenses Issued	←	Critical
	Resource	Laws, regulations	←	Committed
	Resources	Operation licenses and permits	←	Critical
Control Agencies	Resource	Operation information	→	Critical
	Resource	Auditing information	←	Critical
	Goal	Compliance of laws and regulations validated	←	Critical
	Goal	Operation licenses and permits maintained	←	Critical
Competitors and substitutes	Resource	Information of products and strategy	←	Committed
	Resource	Market conditions	←	Committed
Franchises	Goal	Franchises granted	→	Critical
	Soft Goal	Quality of products and services preserved	←	Critical
	Goal	Massive access to customers	←	Committed
	Resource	Operation resources	→	Critical
Direct Customers	Goal	Products and services acquired	←	Critical
	Soft Goal	Quality of products and services	→	Critical
	Soft Goal	Convenient prices	→	Critical
	Resource	Products and services	→	Critical
Suppliers	Goal	Access to Specialized technology, products and services	←	Critical
	Resources	Especialiced technology, products and services	←	Critical
	Soft Goal	Costs and conditions kept stable	←	Committed
	Soft Goal	Quality of products and services	←	Committed
	Soft Goal	Timely payments	→	Committed

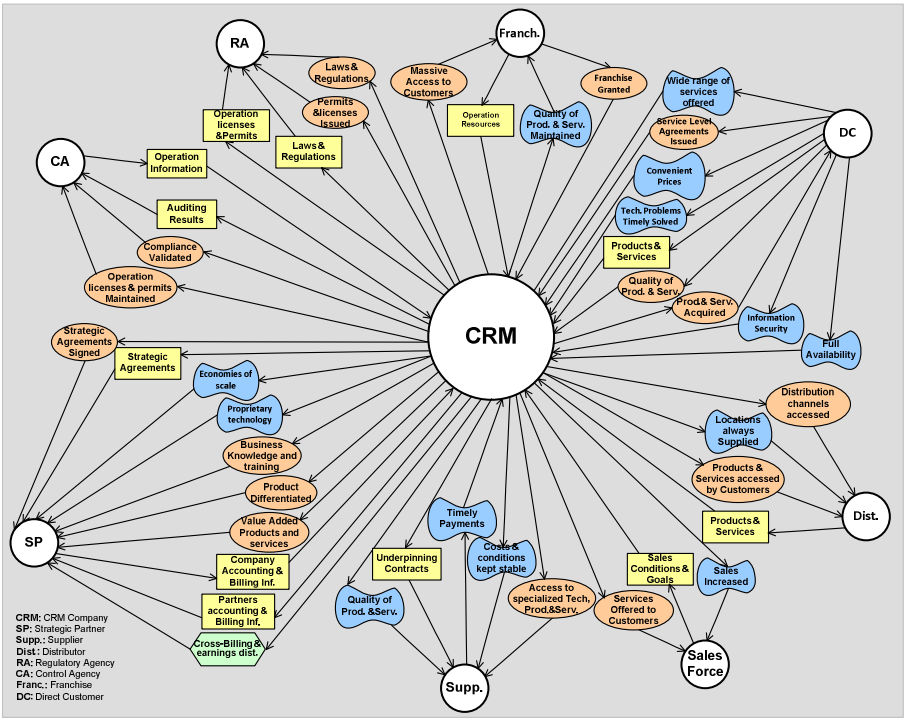

Fig. 7. Excerpt of the CRM Enterprise Context Model Pattern

The CRM patterns in built upon the actors and dependencies identified in Section 4 for Porter's model (i.e., enterprise strategy patterns are designed as an extension of Porter's *i** SD model). New actors have been identified for CRM organizations: *Market Research and strategy Organizations* and *Sales Force*. The first one helps CRM organizations to study and analyze the market, to identify specific market segments and potential services required by their customers. These organizations also help in the evaluation of customer's satisfaction and perception of quality of services. Sales forces help to promote and advertise services, to close sales and thus to increase sales.

6 The Patterns in the Practice

The patterns resulting from our work have emerged from a post-mortem analysis of two large industrial experiences related to the CRM strategy, and later validated in some academic exercises. We are currently using the ERP pattern in an ongoing industrial project. In this section, we will focus on the CRM case. The two projects are the Etapatelecom company case described in more detail in Subsection 6.1, whilst the second project was the formulation of an IT Strategic Plan for an airport. In both cases we applied the DHARMA method. We involved in the post-mortem analysis some selected stakeholders that participated in the use of DHARMA.

Table 2. CRM specific context dependencies

Generic Actor	Depen-dency Type	Dependency	Direction
Strategic Partners	Goal	Services packaged and offered as combos	←
	Goal	Complementary services provided	←
	Soft Goal	Quality of services improved	←
	Soft Goal	Access to new locations	←
Control Agencies	Resource	Customers support and complains reports	←
Direct Customers	Soft Goal	Wide range of services offered	→
	Soft Goal	Appropriated rates and fees	→
	Soft Goal	Information kept secure	→
	Soft Goal	Full availability	→
	Soft Goal	Customers fidelity	←
	Goal	Service level agreements issued	→
	Resource	SLAS	→
	Soft Goal	Timely attention and solution to Problems	→
Suppliers	Goal	Underpinning contracts established	←
	Resources	Underpinning contracts	←
Sales Force	Goal	Service offered to customers	←
	Soft Goal	Service sales closed	←
	Soft Goal	Sales Increased	←
	Resource	Sales conditions & Goals	←
Market research and strategy	Goal	Market surveys issued	←
	Goal	Marked analyzed	←
	Soft Goal	Marked segmentation	←
	Goal	Customers satisfaction evaluated	←
	Soft Goal	Customer needs and requirements identified	←

Concerning academic validation, we remark a project conducted by university students under our direction, in which the context of the institution was modeled guided by the elements in the pattern. The validation was successful, meaning that the resulting context model for the institution was completely obtained from the CRM pattern (in other words, its model coverage was 100%).

6.1 The Telecom Company Case Study

In order to illustrate the practical use of the patterns, the Etapatelecom company (TC) case has been selected. The company provides broadband Internet access and fixed telephone services in Ecuador.

To fulfill its deployment strategy, the TC had to face the selection and adoption of several technologies, and the DHARMA method was used to define its enterprise architecture. Being a utility company, the TC aligns with the CRM strategy. The context model constructed for that case (together with other cases outlined in subsection 6.3) was later used to define the patterns described in this paper. Therefore, it provides good examples to support the practical application of the patterns described in the next literal.

6.2 Applying Patterns to the TC Case

In Section 3, we have provided the formalization of the pattern concept. On top of this formalization, we need to have a casuistic that facilitates their use. Practical application of the patterns is conducted in a systematic way. Each actor and then each dependency in the pattern are reviewed in order to identify their particular instances in the context of the organization under analysis. In this way some typical cases may appear. In this subsection, we illustrate this casuistic using the TC case as if it had been defined as an instance of the CRM pattern.

When analyzing actors, three instantiation cases may emerge (see Figure 8a for the definition and Figure 9 for the examples). In the case of actors, we apply definitions of Section 3 over concrete instances of the following models:

- CRM pattern: CRM = (A_{crm}, D_{crm}, L_{crm})
- Elements of TC: (A_{tc}, D_{tc}, L_{tc})
- Correspondence to be defined: $C_{tc} = (c_A, c_D)$

1) *One-to-one actor's instantiation*. An actor in the pattern is instantiated by an actor in the context of the organization. This is the case of the *Franchise* actor included in the CRM pattern that has been instantiated by the *Telecenter Franchise* actor from TC

$$\text{Franchise} \in A_{crm} \wedge \text{TelecenterFranchise} \in A_{tc} \wedge (\text{Franchise}, \text{TelecenterFranchise}) \in c_A$$

2) *One-to-many actor's instantiation*. An actor in the pattern is instantiated by many actors in the context of the organization. This may happen in two different situations: a) several actors need to collaborate together to provide the intention of the pattern actor; b) the enterprise actor related to the GCA is very generic and is subtyped into several others using is-a. As an example of b), the *Direct Customers* actor in the CRM pattern has been instantiated by *Subscriber* which at its turn is subtyped into *Public Telephone User* and the *Home Telephone User* actors in the context of ETP.

$$\begin{aligned}&\text{DirectCustomers} \in A_{crm} \wedge \text{HomeTelephoneUsere} \in A_{tc} \wedge \\ &\text{PublicTelephoneUser} \in A_{tc} \wedge \text{Susbscriber} \in A_{tc} \wedge \\ &(\text{Subscriber}, \text{PublicTelephoneUser}) \in L_{tc} \wedge (\text{Subscriber}, \text{HomeTelephoneUser}) \in L_{tc} \wedge \\ &(\text{DirectCustomers}, \text{Subscriber}) \in c_A \wedge (\text{DirectCustomers}, \text{PublicTelephoneUser}) \in c_A \wedge \\ &(\text{DirectCustomers}, \text{HomeTelephoneUser}) \in c_A\end{aligned}$$

3) *Null actor's instantiation*. One actor in the pattern cannot be instantiated by any actor in the context of an organization. This is the case of the Sales Force CRM actor, which has not instances in the TC context. Sales force of this organization is internal, thus no relation exist with other organizations.

$$\text{SalesForce} \in A_{crm} \wedge \text{SalesForce} \notin \text{dom}(A_{tc})$$

Table 3 sums up the relation among actors in the CRM Pattern and their instances identified in the TC case (actors with no instances have been omitted due to space limitations). Fig. 9 presents an excerpt of the instantiation of the CRM organization Pattern in the TC case.

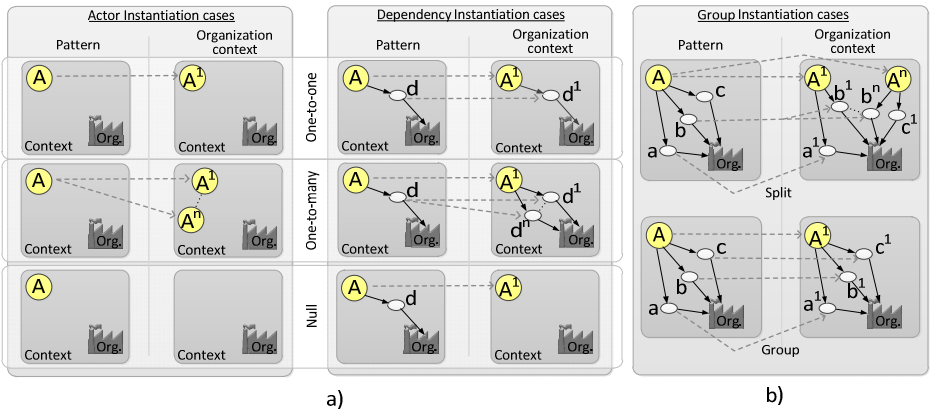

Fig. 8. (a) Actors and Dependency instantiation cases; b) dependency group instantiation cases

Similarly to actors, when analyzing individual dependencies associated to an actor in the pattern, three instantiation cases have been identified (see Figure 8a).

4) *One-to-one dependency instantiation*. A dependency associated to one actor in the pattern is instantiated by one dependency associated to an instance of the actor in the context of the enterprise. So, the softgoals *Full availability*, *Information Security*, and *Wide range of services offered* associated to the *Direct Customers* actor in the pattern, have been instantiated by the soft goals *Full availability*, *Information Security* and *World Wide Connection* associated to the *Subscriber* actor in the context of TC.

$d1=(\text{WideRangeOfServicesOffered},\text{DirectCustomer},\text{CRM},\text{softg}) \in D_{crm} \wedge$
$d2=(\text{WorldWideConnection},\text{Subscriber},\text{TC},\text{softg}) \in D_{tc} \wedge (d1, d2) \in c_D$

(Correctness condition on actors holds, e.g. $(\text{CRM}, \text{TC}) \in c_A$).

5) *One-to-many dependency instantiation*. A dependency associated to one actor in the pattern is instantiated by many dependency associated to an instance of the actor in the context of the organization. In this case, the pattern dependency is considered too coarse for its use in the enterprise context model and thus it is assigned to several dependencies that altogether provide the required intentionality. This is the case of the soft goal *Economies of scale* associated to the *Strategic Partner* actor in the pattern, which has been instantiated by the soft goals *Worldwide connection* and *Access to worldwide IP telephone subscribers* associated to the *Interconnection Provider* actor in the context of *TC*. (The formal definition is similar to the case 2) and is not included for the sake of brevity.)

6) *Null dependency instantiation*. A dependency associated to one actor in the pattern has no instances associated to an instance of the actor in the context of the organization. This is the case of the goal *Strategic Agreements Signed* associated to the *Strategic Partner* actor in the pattern, which has no instances in the case of *Interconnection Provider* actor. Interconnection is mandatory among telephone service providers in the country where TC operates, therefore this goal was not considered. From the structural

point of view and in relation to the instantiation of groups of dependencies associated to an actor in the pattern, two cases may appear (see Figure 6b; for the sake of brevity, we do not show the formal definitions).

7) *Split instantiation*. Dependencies associated to one actor in the pattern are split into groups; each group is associated to a different instance of the actor in the context of the organization. Some of the dependencies can be subject to the One to many dependencies' instantiation case, when they have to be included in more than one group. In our case study, the goal *Value added products and services provided*, and the resources *Company accounting and billing information* and *Partners accounting and billing information*, assigned to the *Strategic Partners* actor, have been grouped and assigned to the *Internet Service Providers* actor in the context of *TC*.

8) *Group Instantiation*. All of the dependencies associated to an actor in the pattern (or the ones for which instances exists in the context of the organization), are associated to a single instance of the actor in the context of the organization e.g., the *Direct User* dependencies in the pattern and the *Subscriber* dependencies in the TC context.

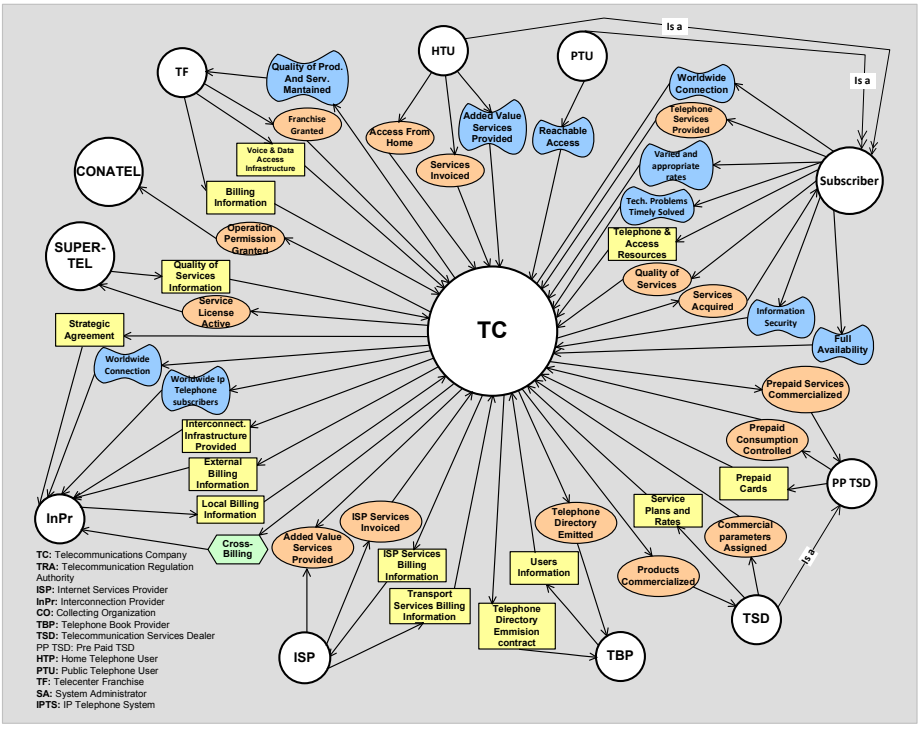

Fig. 9. Excerpt of the instantiation of the CRM organization Pattern in the TC case

Table 3. Actors in the CRM strategy pattern and their instances in the TC case

Actor in Pattern	Instance
Strategic Partners	Interconnection Provider (InPr); Internet Services Provider (ISP); Cable TV Provider (TVP)
Owners	TC Owner
Share-holders	-
Financial Institutions	Banks
Distributors	Telecommunication Services Dealer (TSD); Prepaid TSD (PP TSD)
Regulatory Agencies	National Consul of Telecommunications (NCT)
Control Agencies	Telecommunication Superintendence (ST); Internal Revenue Service; Company's Superintendence
Competitors and Substitutes	Fixed Telephone Companies; Mobile Telecom Companies
Franchises	Telecentre Franchise (TF)
Direct Customers	Subscriber; Home Teleph. User (HTU); Public Teleph. User (PTU)
Suppliers	Telephone Book Provider (TBP)
Sales Force	-
Market Research&Strategy	MR; Deloitte Touche Tohmatsu

6.3 Some Numbers

As mentioned above, the CRM pattern was obtained from two industrial cases and later validated with academic cases, in particular one of the validation cases was of similar size than the two industrial cases and is the one reported here. In Table 4 we present the three cases with the coverage of model and pattern detailed by type of element and the grand total. In the case of the two industrial cases, since this is a post-mortem analysis, it should be interpreted as what could have happen if the pattern would have existed in advance. We may see that the two coverage measures are good enough. From the point of view of the models obtained, most of the actors of the enterprise context model are bound to CRM pattern actors, except for one actor in each context model. Dependencies also are mostly result of instantiation from the pattern. From the point of view of pattern coverage, percentages show that the pattern captures the needs of an enterprise that applies the CRM strategy.

With respect to the academic validation case, we obtained the University model applying directly the CRM pattern, and the fundamental result was that the model obtained from the pattern was good enough as to consider unnecessary to add more elements, so model coverage was 100% (probably an extreme case). The coverage of the pattern was even greater than in the two industrial cases: even if the model in the academic case does not identify dependencies for all 28 actors, the numbers point out to an increase in the number of dependencies identified when using the pattern. However, this fact shall be validated with future experiences.

Table 4. Numbers in relation to the industrial and academic cases

	TC case			Airport case			Academic Case		
	nb. elements	model coverage	pattern coverage	nb. elements	model coverage	pattern coverage	nb. elements	model coverage	pattern coverage
actors	20	95%	90%	26	97%	90%	28	100%	93%
goal dep.	20	85%	83%	56	80%	91%	126	100%	87%
softg dep.	18	73%	90%	38	79%	85%	83	100%	92%
res. dep.	20	80%	84%	32	90%	82%	75	100%	86%
TOTAL	78	83%	87%	152	87%	87%	312	100%	90%

6.4 Applicability Issues

In the case studies conducted so far, we have learned some practical tips that may help in making a winning case when applying the presented approach:

- Define a multidisciplinary team before starting the process, to support the analysis. Include staff at least from financial, legal, marketing and commercial department.
- Provide basic training to participants about the modeling concepts. Conduct the training sessions in short (max. 2 hours) workshops.
- In training workshops, sketch first a simplified version of the *i** SD model representing the pattern and provide sample dependencies to clarify concepts.
- As a first step in the construction of context model of the organization, use actors in the patterns as checklist (one at the time) and identify all of their instances in the context of the organization. Include them in a two columns table: the first column for actors in the pattern and the second one their associated instances in the context of the organization.
- Next, use the dependencies associated to actors in the pattern as checklist to identify their instances in relation to each identified actor instance in the context of the organization. Sketch partial *i** SD models to record and discuss about identified dependency instances with participants.
- When working with stakeholders, as a general rule: do not try to draw perfect *i** models. Just draw quick sketches including 2 or 3 actors maximum (preferable instances of a same actor in the pattern). Graphical conventions as those included in [3] and also in this paper (e.g., tabular representation of models) can be used.
- Add additional columns to the table to include dependency description, type and direction, as shown in Tables 1 to 3. Record in these columns all dependencies identified regarding each actor.
- Do not try to address all actor instances in a single workshop. It is better to conduct several shorter meetings addressing instances of single actor (or groups of actors) in the pattern at the time.
- Draw the final *i** SD context model only when all the dependencies have been identified and tabulated.
- Draw complete *i** context models only when specifically required and with proper justifications. *i** graphic representation is great for brainstorming and discussing with meeting participants, but large models tend to be confusing and costly in terms of time. Partial diagrams are just as good and easier to draw.

7 Conclusions

In this paper, we have presented a pattern-based approach for constructing intentional context models as *i** SD diagrams. To this aim, we have analyzed a general model for enterprises, Porter's model; one particular example of enterprise strategy, the CRM strategy; we have formulated patterns for this strategy; and we have studied the results of applying the patterns to some cases.

The main advantage of our approach is its industrial applicability, in terms of theory, scalability and orientation:

- The patterns synthesize knowledge about business strategies, making it accessible to requirements engineers and helping to close the gap between them and enterprisers since they can be used as communication bridge among technical and administrative staff.
- Being based on solid theories like the Porter's model and enterprise strategies like CRM, they provide a general foundation that applies to a lot of enterprises.
- The level of detail (thanks to the solid foundation) including much more model elements that other existing proposals, makes it feasible to apply to real cases.
- It is also important to remark that the pattern-based approach has been formally defined using an algebraic formulation of *i** as baseline. This is also a differentiating characteristic compared to other pattern-based approaches.

In addition to the context model patterns presented in this work, which resulted from the application of the first activity of the DHARMA method in industrial and academic experiences, we are also working on system context patterns and hybrid systems architectural patterns for generic ERP, CRM and SCM strategies. These patterns resulted from the application of the second, third and fourth activities of the DHARMA method.

References

1. Alexander, C.: The Timeless Way of Building. Oxford Univ. Pr. (1979)
2. Yu, E.: Modelling Strategic Relationships for Process Reengineering. PhD. University of Toronto (1995)
3. Carvallo, J.P., Franch, X.: On the Use of *i** for Architecting Hybrid Systems: A Method and an Evaluation Report. In: Persson, A., Stirna, J. (eds.) PoEM 2009. LNBIP, vol. 39, pp. 38–53. Springer, Heidelberg (2009)
4. Porter, M.E.: Competitive Strategy. Free Press (1990)
5. Steinberg, M.: Enterprise Applications: A Conceptual Look at ERP, CRM, and SCM. Hill Associates Inc. (2006)
6. Coyne, K.P., Balakrishnan, S.: The McKinsey Quarterly 4 (1996)
7. Fuxman, A., Giorgini, P., Kolp, M., Mylopoulos, J.: Information Systems as Social Structures. In: FOIS 2001, pp. 10–21. ACM Press (2001)
8. Kolp, M., Giorgini, P., Mylopoulos, J.: Information Systems Development through Social Structures. In: SEKE 2002, pp. 183–190. ACM Press (2002)
9. The *i** Wiki, `http://istar.rwth-aachen.de/tiki-view_articles.php`
10. Turban, E., et al.: Information Technology for Management: Transforming Business in the Digital Economy. John Wiley & Sons (2002)
11. Anderson, K., Kerr, C.: Customer Relationships Management. Mc.Graw Hill (2002)