Teaching Information Systems: an *i**-based approach

Juan Pablo Carvallo

Computer Science Department
University of Cuenca, Ecuador

pablo.carvallo@ucuenca.edu.ec

Abstract. Modern enterprises rely on information systems (IS) both to support their operation and provide information required to endorse strategic decisions. Because of their increasing complexity, such systems are usually constructed by integrating software components of different nature and origins, into hybrid systems, for which architectural design plays a fundamental role. If well engineered, it becomes the blueprint required for IS to add value to the organization. However, IS architecting is not an easy task, mainly due to the fact that engineers often lack of knowledge related to business strategy and operations, but also proper training in methods and techniques required to conduct IS architectural design in a well-structured way, without losing focus on business strategy. In this paper we present an *i**-based approach to lecture information systems, which keeps in mid both business strategy and technology.

1 Introduction

Modern enterprises rely their strategy on Information Systems (IS) required to, support and orchestrate their operation, provide information to endorse decision making, and eventually add value to their businesses. Because of their complexity, such systems are currently built by integrating components of different nature into *Hybrid Systems* [1], encompassing third-party *Off-The-Shelf components* [2], *legacy* and *bespoke systems*. IS architecture plays a fundamental role. It is used to guide components construction and/or selection, and integration into a dependable systems, in which components interact in a reliable and coordinated way to support business strategy.

However, IS architecting is not an easy task; engineers are usually trained in methods and technologies required to design and build components from the scratch, but often lack knowledge related to *business strategy* and *operations*. This leads them to prioritize technology over strategy and thus, to assemble IS that do not fully comply with needs or add value to organizations. It also leads them to adopt merely operative, subordinated roles, instead of visionary, managerial and strategic ones, hampering professional growth.

It is therefore highly relevant to improve their training to include curricula covering these aspects. This paper presents an *i** based approach to lecture IS, intended to develop skills related to business modeling and strategy, and the understanding of their impact on IS architecture and vice versa.

The remaining of this paper is structured as follows; section 2 introduces the context and the course; section 3 describes the lecturing method; Section 4 gives some concluding remarks.

2 The context and objectives of the course

Cuenca University is one of the top five ranked universities in Ecuador. Its Engineering Faculty offers major degrees in several engineering fields including Systems Engineering. This five years long program includes, in addition to general education, courses mainly related to the ACM software engineering curricula [3], but also some in relation to Computer Science. Information Systems is a core course in the Systems Engineering program. Lectured in the second semester of the fourth academic year, it encompasses 64 core hours of classroom activities. Its learning objectives are:

- Understand the managerial role of the systems engineers and its responsibility as value generators for the organization.
- Identify the responsibility (dependencies) of an organization in relation to its context and the strategy required to fulfill it.
- Understand enterprise organization (value chain –see next section-) and the roles of organizational areas to support enterprise strategy.
- Identify the benefits that a IS could bring to the organization in order to support its strategy (external requirements) and operation (internal requirements)
- Define the IS architecture required to support decision making at different organizational levels, and the goals for software components required to implement it.

The main goal of the course is getting students to understand and keep focus on business strategy, in order to design IS architectures required to construct systems that generate value to the organization and the actors in its context.

3 The lecturing method

The course is conducted in a semi-industrial way; students are required to complete analysis and modelling activities in industrial settings (organizations that agreed to cooperate with the program). Students work in couples to complete assignments, supported by responsible of the assigned organizations. At the end of academic period, organizations are provided with all documents produced in the course. Academic process starts by constructing *Context Models* (CM) of the organization and ends with the identification of the IS architecture (actors that structure the system, services that must be covered by each of them and their relationships). The course is divided into 7 units (in addition to its introduction). The first three units provided basic information in relation to methods, models and notations to be used, whilst units 4 through 7 are intended to guide students on the core assignments, making strong use of *i** models and modelling techniques (see Fig. 1). The next paragraphs resume activities conducted in each of these units.

- **Unit 1: The business strategy framework.** Strategic framework is based in Porter's models of *market forces* and *value chain* [6]. The first is designed to analyze the influence of five competitive forces on business context (*threat of new entrants*; *threat of substitution*; *bargain power of customers*; *bargain power of suppliers*; and *rivalry among current competitors*) and reason about strategies to make organizations profitable. To balance the forces, enterprises adopt an internal organization known as Value Chain, which encompasses five *primary* (*inbound logistics*; *operations*; *outbound logistics*; *marketing & sales*; and *support*) and four *support* (*Infrastructure*; *human resources management*; *technology development*; and *procurement*) value activities (VA), required to generate value and eventually a *margin* (difference among total value generated and cost of performing VA). Primary activities are core and specific of business whilst supporting ones are transversal to them.

  Students are required to analyze organizational chart and identify organization's value chain, by aligning *Organizational Areas* (OA) with value activities included in Porter's generic value chain model (see Fig.2). Next, they are required to analyze the organization in relation to each market force, with support of representatives of each OA, and identify *Context Actors* (CA) in relation to them, e.g. types of suppliers (services/raw materials/supplies, domestic/international, etc.), types of consumers (local/national/international, cash/credit, etc.). We advise students to acknowledge four types of actor's hardware, software, organizations and people.

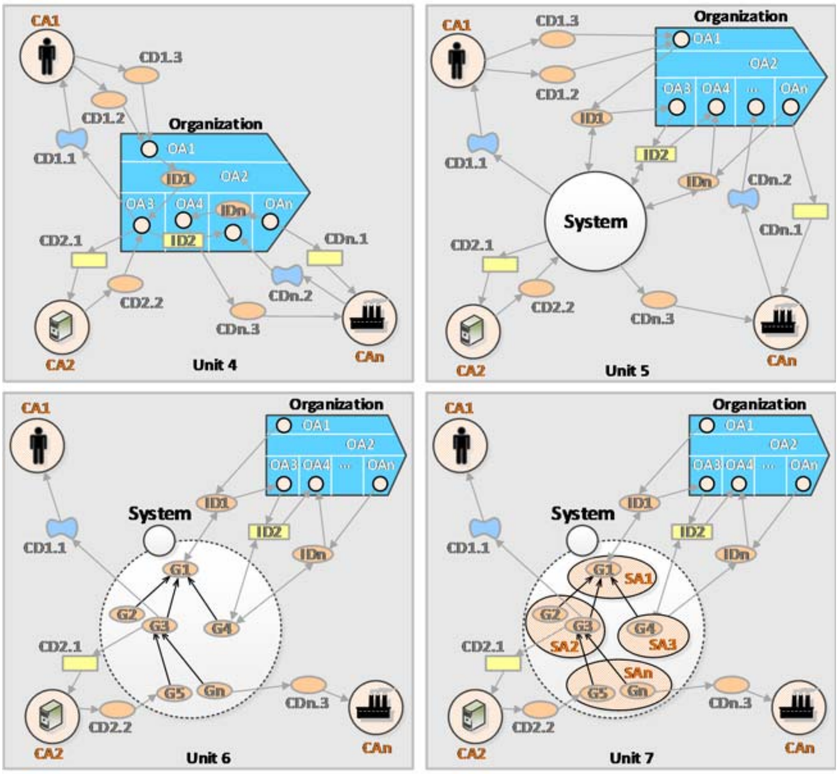

Fig. 1. Sketch of the method proposed in units 4 to 7.

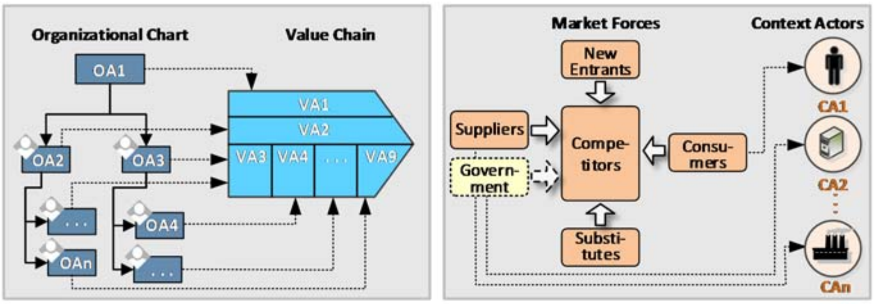

Fig. 2. Mapping of OA to VA (left) and identification of CA in relation to market forces (right).

- **Unit 2 and 3: Introductory concepts.** Unit 2 provides basic training on composite hybrid systems [1], types of components that can be used for their construction, empirics to support the decision whether to acquire or build a component from the scratch and a quick review of integration techniques. Unit 3, introduces students to *i** modelling notation; SD and SR models and their constructs are reviewed and several examples are provided. The course adopts the simplified version of the *i** metamodel reported in [5] e.g. actors are treated in a generic manner, without distinguishing roles, positions and agents; use only *is-a* and *is-part-of* actor links; avoid task dependencies since they are too low level (focus on goals that can be operationalised by them); use just goals as intentional elements and goal decompositions as intentional elements links, inside actors' boundaries, etc.
- **Unit 4: Modelling the enterprise context.** The organization and its strategy are analysed in detail, to identify its role inside the context. CA identified in unit 1 are examined in relation to each OA in the value chain, to identify strategic needs among them (*Context Dependencies –CD-*). Also OAs are analysed in relation to each other to identify their strategic interactions (*Internal Dependencies –ID-)*. Students are asked to construct an *i** SD context model from the perspective of each OA, including their related CA and OAs as well as their CD and ID. Resulting models are combined into a single enterprise Context Model (see Unit 4 of Fig. 1). Guidelines included in [5] are provided to help identify CD and ID. Graphical guidelines are also provided e.g., change the graphical representation of dependencies from standard curved lines with oriented "D"s, to straight lines with arrowheads (see Fig. 1).
- **Unit 5: Modelling the environment of the system**. An IS-to-be is placed into the organization, and the impact that it has over the context is analysed. Strategic dependencies identified in unit 4 (internal and external), are examined to determine which of them may be totally or partially satisfied by IS. These dependencies are redirected inside the *i** SD diagram to the IS. The model includes the organization itself as an actor in IS environment, its needs are modelled as strategic dependencies over the IS (see Unit 5 of Fig. 1). Context Model resulting from unit 4 is transformed to its tabular form as proposed in [6] (see first 5 columns of table 1), to help manage size. Columns "*Total*" and "*Partial*" are used to state which dependencies can be fully or partially automated. These dependencies structure the Context Model of the IS (SCM). "*Why*" column is used to state reasons why dependencies are considered to be automatable (for teacher validation purposes).

- **Unit 6: Decomposition of system goals and identification of system actors**. Dependencies included in the SCM are analysed 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, using means-end links of type goal-goal (representing then a decomposition of objectives into sub-objectives) (see Unit 6 of Fig. 1). Column "*How*" is added to the table resulting from unit 5, to state goals and sub-goas in the tabular representation (see table 1). As a hint for goal identification we ask students to think about basic data to be stored, transactions and processes required to manage and transform it and, basic queries and reports to be obtained. Sub-goals are refined until they represent services atomic enough, such that it does not makes sense to further reduce them. The process is complete when all the CD and ID have been considered and linked to appropriated sub-goals required for their fulfillment.
- **Unit 7: Identification of system architecture**. Finally, goals included in the SR model are analysed and systematically grouped into *System Actors* (SA). Objectives are clustered into services, according to an analysis of the strategic dependencies with the environment and an exploration of software components marketplace. Relationships between SA that form IS architecture are described according to the direction of the means-end links that exist among the objectives included inside them (See Unit 7 of Fig. 1). Additional columns are added to the table, one for each SA identified (see table 1). Goals and sub-goal are linked to SA by placing a capital letter "I" in the proper cell (SA column; goal/sub-goal row) if they have to be implemented in SA, and a "D" when they require data or supporting functionality from other SAs (interoperability requirements). SA are not software components; they represent atomic software domains for which several situations may occur: there can be a software component covering the functionality of several SA; the functionality of a single SA is covered by several software components for ubiquity reasons e.g. mobile and local applications; or there can be cases for which no software components exist, leading to the need of bespoke software.

Table 1. Sketch of tabular represetnation of *i** SD and SR models in relation to Fig. 1

Actor 1	Type	Direction	Dependency	Actor 2	Total	Parcial	Why	How	SA1	SA2	SA3	…	SAn
CA1	☁	←	CD1.1	OA3		X	Just 1	G3	D	I			
CA1	⬭	→	CD1.2	OA1									
CA1	⬭	→	CD1.3	OA1									
CA2	▭	←	CD2.1	OA3		X	Just 2	G3	D	I			
CA2	⬭	→	CD2.2	OA3		X	Just 3	G5		D			I
…													
CAn	▭	←	CDn.1	Oan									
CAn	☁	→	CDn.2	…									
CAn	⬭	←	CDn.3	Can	X		Just 4	Gn		D			I
OA1	⬭	→	ID1	OA3	X		Just 5	G1	I				
								G2	D	I			
								G3	D	I			
								G4	D		I		
OA3	▭	→	ID2	…		X	Just 6	G4	D		I		
…													
OAn	⬭	→	Idn	OA4		X	Just 7	G4	D		I		

4 Concluding remarks

This paper presents an *i**-based approach for lecturing IS and IS architecture. The approach helps students to keep focus on business strategy and technology at a time. It starts by modelling the environment of the organization and concludes with the definition of the IS architecture required to support its operation and the strategic interactions with actors in its context. The lecturing method makes intensive use of *i** models; after conducting the course in several editions, we have found that the method provides several benefits for students:

1. It helps to organize their work making IS architecting more structured.
2. The learning process constantly addresses strategic needs, helping students to keep focus on them.
3. It makes evident the interaction among IS components, consequently architectural design can be used to support the selection and/or construction of components that may integrate more seamlessly.
4. The architecture oriented nature the method, allows for the definition of IT strategic plans. Projects can be better justified to managers, in terms of the services that they will provide to CA and OA, and how they satisfy strategic internal and external needs.

Although we find the lecturing method very valuable and the evaluation of students is usually positive, there are problems of different nature that we have to address: *Notational* (organic nature of graphical notation, difficulties to scale and poor tool support, problems typically associated to *i** models); *Semantic* (confusion among dependency types, their direction or the correct way to state their descriptions); *Decomposition and groping of goals*; and *Organizational* (lack of organizations willing to cooperate, or size of the ones willing to do it -too large o too small-).

References

1. *Proceedings of the 7th International Intl. Conference on Composition-Based Software Systems* (ICCBSS), IEEE, 2008.
2. J. Li et al. *A State-of-the-Practice Survey of Risk Management in Development with Off-the-Shelf Software Components*. IEEE TSE, 34(2), 2008.
3. The Joint Task Force for Computing Curricula 2005. *Computing Curricula 2015.* ACM and IEEE, 2006. Available at https://www.acm.org/education/education/curric_vols/CC2005-March06Final.pdf
4. M.E. Porter. *Competitive Strategy*. Free Press, 1980.
5. Carvallo, J.P., Franch, X. *On the Use of i* for Architecting Hybrid Systems: A Method and an Evaluation Report*. PoEM 2009. LNBIP, Springer, 2009.
6. Carvallo, J.P., Franch, X. *Building Strategic Enterprise Context Models with i*: A Pattern-Based Approach*. TEAR 2012. LNBIP, Springer, 2012.
7. C. Ayala et al. A Comparative Analysis of i*-based Agent-Oriented Modeling Languages. SEKE 2005.