Towards a Communicational Perspective for Enterprise Information Systems Modelling*

Arturo González[1], Sergio España[2], and Óscar Pastor[2]

[1] Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
agdelrio@dsic.upv.es
[2] Centro de Investigación en Métodos de Producción de Software
Universidad Politécnica de Valencia
{sergio.espana,opastor}@pros.upv.es

Abstract. This paper presents an overview of Communication Analysis, an approach to enterprise information systems (ISs) modelling that adopts a communicational perspective. It consists of a method and its underlying requirements structure. The proposed approach is conceived to be conceptually sound (it is founded on settled knowledge from various scientific fields), practically prescriptive (it offers guidance and criteria) and flexible (it offers specific strategies to deal with static and dynamic aspects of ISs). Both the method and the structure have been proved to be successful in complex projects.

Keywords: Enterprise Information Systems, Requirements Engineering.

1 Introduction

Enterprise modelling covers the set of activities, methods, and tools related to developing models for various aspects of an enterprise [1]. In this paper, the term enterprise refers to an organisational system that needs to manage its activity and its knowledge. An enterprise model is a consistent set of complementary models (a.k.a. viewpoints) describing the various facets of an enterprise to satisfy some purpose of some business users (a.k.a. stakeholders) [1]. Proposals for enterprise modelling may differ in their purpose, the content of the model, the quality of formalism, the level of abstraction, and the span of existence [2]. Concerning information systems (ISs), the purpose of an enterprise model can be to serve as a requirements specification for the information system (IS) computerisation [3]. This paper focuses on this purpose. We consider requirements elicitation to be a process that involves not only gathering requirements and constraints related to a software product (solution viewpoint), but also understanding and specifying organisational work practice (problem viewpoint).

* Research supported by the Spanish Ministry of Science and Innovation (MSI) project SESAMO (TIN2007-62894), the MSI FPU grant (AP2006-02323), and FEDER.

J. Stirna and A. Persson (Eds.): PoEM 2008, LNBIP 15, pp. 62–76, 2008.

This paper proposes a requirements elicitation[1] method and a requirements structure that have been specifically conceived for the development of enterprise ISs. We refer as enterprise IS to a system of people, processes, data and material resources that supports the management and daily activity of an enterprise, by providing the strategic, tactical and operational levels with the information that they need for their performance. We share the view of Langefors that the IS is a support for organisational communication [4]. This paper proposes Communication Analysis as an approach to IS modelling. Communication Analysis adopts a communicational perspective by focusing on communicative interactions while eliciting requirements. Communication Analysis covers several enterprise model viewpoints (e.g. resources, activities, information, organisation). In case of necessity, it can be complemented with other viewpoints (e.g. economic, optimisation).

Special emphasis is placed on the structure of the requirements specification (RS). We consider that the lack of a proper structure is one of the factors that hinder the industrial adoption of Requirements Engineering (RE) proposals. In this paper, we argue about the concept of requirement and we enumerate issues related to their structure. E.g., a requirements structure should facilitate the elicitation process by supporting fact finding: we advocate binding together method and structure. The proposed requirements structure is founded on Systems Theory and Communication Theory, among other fields of science. The method borrows concepts and revises techniques from RE and Software Engineering, among other disciplines. Furthermore, specific modelling techniques are proposed to deal with communicational aspects of ISs (e.g. Communicative Events Diagram and Communication Structures). Communication Analysis (the method and the requirements structure) is the result of many years of applied research. The approach has been successfully put in practice in several enterprise ISs development projects.

The paper is outlined as follows. Section 2 motivates the work; several issues that hinder the industrial adoption of RE proposals are discussed. Section 3 introduces Communication Analysis; the underlying requirements structure and the main activities that compose the method are explained. Section 4 discusses the qualities of the proposal with respect to the motivating issues. Section 5 reviews related work. Section 6 presents the conclusions and future work.

2 Motivation: Recurrent Problems in Requirements Engineering

There is no agreement in ISs development project failure rates. The widely cited 1994 Chaos Report suggests that 84% of projects are unsuccessful [5]. The report uses an ambiguous concept of failure and, therefore, its validity has been questioned [6]. A 2006 Chaos Report highlights “a major up-tick” (sic) in success rates: the rate of unsuccessful projects drops to 65% [7]. In any case, it is recognised that information and communication technologies development is a risky activity. Various factors are considered to cause failure. E.g. project complexity increases risk significantly [8], and an inadequate requirements practice is also considered a major factor [9].

[1] We refer to the discovery and description of requirements as *elicitation*. We acknowledge that some of the terms that we use appear sometimes in the literature with a different meaning.

The sizeable scientific production in the area of RE indicates that we are dealing with an open problem. Many challenges issued by Bubenko in 1995 [10] are still valid today. As some studies show [11][12], new requirements elicitation techniques keep on being proposed but they do not live up to industry expectations. Many factors hinder RE industry adoption [13][14]; among these factors we will focus on those related to the inherent complexity of RE. We claim that there exist differences in criteria about the concept of requirement itself. We consider that the following issues are source of bad practice in requirements specification (RS).

1. **Requirements should offer an external view of the system under development.** Many authors agree on the fact that requirements should describe the user perception of the system [15][16]. Requirements are considered as functions and characteristics of a system that can be perceived externally [17]. Yet, something as basic as information needs is not the essential element of many RS proposals. Instead of determining what information the users need to carry out their tasks, many RS proposals emphasise the viewpoint of how the application is used. An IS is a socio-technological system [18], a medium that supports the communications of an organisation [4]. Therefore, we consider that input and output messages are the external concept par excellence. We believe that IS requirements elicitation should be organised around the communicative interactions between the system and its environment.
2. **Requirements should differentiate the problem space and the solution space.** Our stance is that input and output messages are the basis of an IS RS. They define user's needs (i.e. problem space), independently of the implementation technology. The way in which messages are built and conveyed, and the technological support that allows treating these messages efficiently determine how the problem is solved (i.e. solution space). E.g. an order (input message) requested by a client can be 'solved' by introducing data via a keyboard, scanning bar codes, reading magnetic cards, etc. Whatever the solution chosen, the input message is invariable.
3. **Requirements elicitation practices should facilitate user-developer communication.** So as to establish shared knowledge. However, in many cases, stakeholders do not understand the RS language. The lack of shared knowledge between business experts and the development team hinders aligning the software with the needs of the enterprise.
4. **Requirements specifications should be well structured.** It is necessary to determine a structure for the set of requirements (i.e. for the RS document) and a structure for each individual requirement. In many cases, the structure of the set of requirements consists of a enumerated list of requirements and the structure for each individual requirement consists of a description template [19][20][21]. Templates facilitate a cognitive reasoning process that is *local* to a given requirement; that is, they allow to interact with the user in order to ask for details of the requirement that the analyst could miss without the template. The structure of the set of requirements should facilitate the cognitive reasoning of requirements that are *adjacent* to a given one; that is, when a stakeholder expresses a requirement, the structure of the set should promote the discovery of requirements that are related to the initial one (by helping the analyst ask the proper questions). However, enumerated lists are poor from the methodological and cognitive points of view. Solely enumerating facts hinders systematising fact finding. It is hard to

support an interactive and structured search process by means of enumeration, since the search space is not sequential. We consider that a proper structure for the set of requirements would improve requirements elicitation and management.

5. **Methods should be adaptable (and should provide flexible guidelines).** In a development project, there appear distinct types of problems that may need distinct problem-solving approaches. E.g. each kind of stakeholder recounts knowledge that can be related to derived-information needs (e.g. managerial staff tend to need listings and printouts), or to input-information needs (e.g. operational staff tend to need business forms). The way to elicit requirements is contingent upon the characteristics of the problem. When designing a requirements elicitation method it is advisable not to force a single orientation. The method should guide the analyst to tackle each problem with the proper reasoning tool. Note that flexibility does not mean indefiniteness: requirement elicitation guidelines should be prescriptive.
6. **The (inevitable) ambiguity of the concept of requirement should be overcome.** Some authors have given widely-accepted definitions of term requirement. But those definitions do not help much to deal with requirements. For instance, it is common to express that a RS should specify *what* the system does, instead of *how* the system does it. In fact, we did so above (see issue 2). However, the difference between the *what* and the *how* is not objective, as Davis shows in [17]. Standard definitions of requirement (e.g. a condition or capability needed by a user to solve a problem or achieve an objective [15]) neither specify the scope, granularity or structure of a requirement. To overcome this problem, the concept of requirement needs to be reinforced with instructive RS exemplifications, prescriptive elicitation guidelines and a requirements structure that facilitate the process. The need for requirements structure is argued in issue 3 and is a major contribution of this paper (see Section 3). Regarding exemplification, it is the instances (the examples) that complete the meaning of a concept [22]. Unfortunately, authors are not in the habit of illustrating their work with instructive examples that serve as reference. As stated in [12], examples found in RE literature are usually irrelevant.
7. **Requirements elicitation methods and structures should provide an integrated perspective of all the viewpoints of the analysed system.** In the current RE scene there exists a wide range of requirements elicitation techniques. Each technique is focused at a specific viewpoint of the system. Some techniques offer a similar viewpoint and can be considered mutually exclusive (e.g. BPMN and UML activity diagrams). Other techniques offer complementary viewpoints and can be used combinedly (e.g. UML class diagrams and state-transition diagrams). Methods should choose a set of combinable techniques and exploit their complementarity in order to integrate their viewpoints. This integration would foster the requirements structure navigability.

The above-mentioned issues are the subset of RE problems that motivates our work. We do not claim to have found a solution to each and every RE problem. Our main concern is providing a method that allows dealing with complexity and providing a requirements structure that is properly organised. Next section presents the proposed approach.

3 Requirements Structure Underlying Communication Analysis

There are, at least, three systems involved in ISs development [23]. The enterprise (a.k.a. organisational system) is a social system that needs the information that the IS provides in order to achieve its goals. The IS is a system composed of heterogeneous agents and it is intended to facilitate the enterprise work practice by means of supporting information tasks (e.g. acquiring, storing, retrieving, distributing information). The subject system[2] is the portion of the world that the enterprise wants to observe, control or influence [18]; the enterprise does so by means of performing communicative interactions and storing relevant facts about the subject system. RS should be organised around the set of interactions/messages that enterprise actors need in order to carry out their tasks. Communicative interactions constitute *what* the enterprise needs (the problem); therefore we consider them to be the main requirements to be discovered and described. The rest of requirements qualify communicative interactions by stating aspects, qualities, constraints, etc. of the communication. These solution requirements constitute *how* the system has to be implemented or has to perform. Communication and information requirements are related to *efficacy*; that is, the adequacy of the information supplied to achieve a task. Solution requirements relate to *efficiency*, to the minimisation of operation and usage costs. E.g. usability requirements and response-time constraints are intended to reduce costs. We further differentiate two kinds of solution requirements: usage requirements offer an external viewpoint and lead to interface design; operational requirements offer an internal viewpoint and lead to internal component design.

We propose a layered structure for requirements that covers both the problem and the solution spaces. This structure is the backbone that supports requirements elicitation (both the discovery and the description of requirements). The first level of the requirements structure corresponds to the systemic levels involved in enterprise ISs modelling. Each requirement is ascribed to one systemic level. We will also refer to the proposed systemic levels as requirements levels. The Communication Analysis method can be explained in terms of a composition of activities that are associated to the requirements levels. Figure 1 shows the requirements levels (to the left) and the Communication Analysis workflow (to the right). Today it is widely accepted that an iterative and incremental software lifecycle facilitates software development. However, the workflow is presented sequentially for the sake of understandability; it can be considered that this flow can be repeated on each development process iteration (in Figure 1, this fact is informally expressed by means of a spiral).

This structure offers two dimensions. One dimension is related to the static-dynamic duality (horizontal axis of Figure 1). The other dimension is related to refinement (vertical axis of Figure 1).

In organisations, a duality appears among dynamic and static aspects: business interactions are such things because they affect business objects and, in the opposite way, certain objects are considered business objects because organisational interactions deal with them. Communication Analysis eases to deal with this duality by offering techniques for interaction analysis (activities 2, 4 and 6) and for the analysis of business objects (activities 3, 5 and 7). This way, following a systemic

[2] In the literature, the *subject system* is sometimes refered to as *Universe of Discourse*.

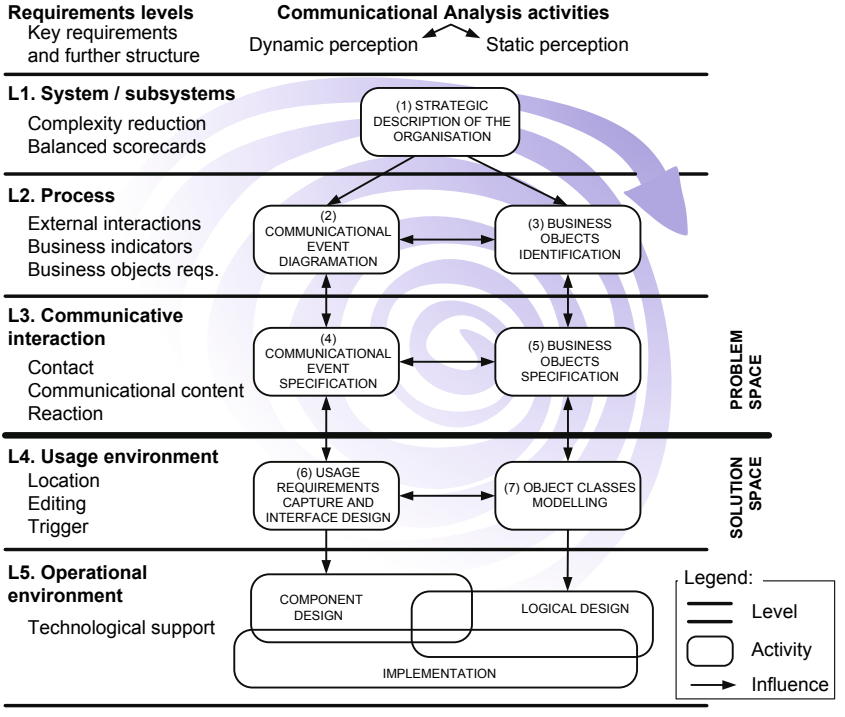

Fig. 1. The requirements structure and the Communication Analysis workflow

approach, Communication Analysis allows to use stepwise refinements techniques in a twofold perspective. From a dynamic perspective, analysing business processes and obtaining, from the process specification, the business objects structure. From a static perspective, discovering the business objects structure and, then, reasoning the communicative interactions that allow the users to deal with those business objects. This intertwining of both perspectives makes the method flexible and contingent.

We consider each elicited requirement to be of one of these kinds:

- A communicative interaction between the information system and its environment. E.g. a balanced scorecard, the reception of an order from a customer.
- A refinable aggregate of interactions. E.g. the accounting department (seen as a set of interactions), the set of communications related to sales management.
- Part of the specification of a communicative interaction. E.g. a message structure description, the description of a particular data field.
- A characteristic of a (set of) requirement(s) of any the previous kinds. E.g. statements such as “every amount shall be stored with 9 decimals”, “every listing for the accounting department shall have a timestamp”.

Each and every requirement is associated to a specific interaction or to a set of interactions. This way, the communicational perspective of the Communication Analysis forces every requirement to be subordinated to communicative interactions.

During the first stages[3] of requirements elicitation, the analyst seeks to obtain the repertoire of communicative interactions that the users need for their work practice. In later stages, the analyst (or designer) confronts the design of the interface that will support the editing of messages. Also, the memory of the system is designed to ensure the persistence to the communicative interactions. Lastly, the technological architecture is designed and the software application is implemented. In many ways, the techniques that we propose are an evolution and enhancement of existing techniques. For instance, the analysis of external interactions (done during activity 2) is a revision of business process modelling techniques. Next, the requirements structure and the method are explained in more detail. For the sake of brevity, only the techniques with a clear communicational nature are exemplified; namely the Communicative Events Diagram and Communication Structures.

L1. System/Subsystems Level

The first requirements level proposed is related to the point of view of business strategy; i.e. the information needed to manage the enterprise. The two main kinds of requirements ascribed to this level are: (a) communicative interactions (mainly outputs) that enable strategic management and decision making; (b) refinable aggregates of communicative interactions in the shape of subsystems.

Activity 1 consists of creating a strategic description of the organisation. Strategic requirements allow the definition of balanced scorecards [24]. In addition, when big enterprises are confronted, it is convenient to decompose the problem into subsystems or organisational areas. This way the analyst obtains aggregates of interactions that are more manageable, and problem complexity is reduced.

L2. Process Level

Communication is the essence of business process analysis in Communication Analysis. The analyst seeks to discover the set of communicative interactions of the enterprise. The method offers unity criteria to help determine the granularity of communicative interactions. Then, these interactions are ordered in space-time.

Activity 2 consists of determining the set of external interactions of the IS with its environment. By external we mean that we seek to describe system behaviour and not system composition. Communication Analysis offers a set of unity criteria that allow to objectify the appropriate granularity of communicative interactions. The term *communicative event* is used to refer to a complete communicative interaction (an interaction that satisfies the unity criteria)[4]. The elicited communicative events are interrelated by means of temporal precedence relations, creating a *Communicative Events Diagram*[5]. This diagram specifies the set of interactions of a business process, highlighting the information needs of the involved stakeholders.

---

[3] We refer as *stage* to the set of those activities that are related to the same requirements level.

[4] Unity criteria for the communicative event are not explained herein for reasons of space.

[5] Note that the Communicative Events Diagram actually specifies *classes of* communicative events; we omit this qualifier to keep the name shorter (just as the Activity Diagram does).

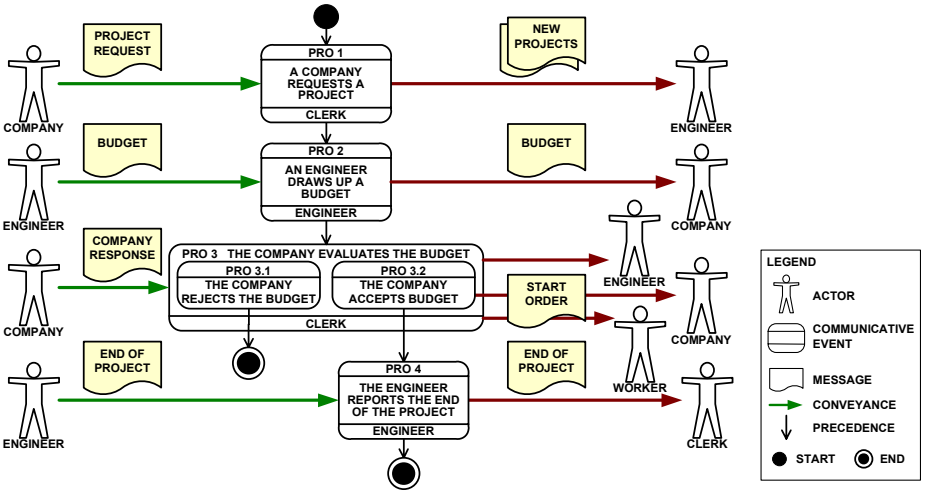

Fig. 2. Communicative Events Diagram of a projects office

Consider the case of a projects office that carries out industrial electrical installations[6]. A company (customer) contacts the projects office clerk, who opens a project record where data about the requested project and the customer company is registered. Every morning, the electrical engineer checks new projects, s/he visits the company and, depending on the customer needs, s/he draws up a budget and attaches it to the project record. A copy of this budget is sent to the company. The company either accepts or rejects the budget. If the budget is accepted, the clerk issues an order to start the project. Once the project is completed, the engineer reports this fact so the clerk can manage the payment. Figure 2 shows the Communicative Events Diagram that specifies the projects office IS, from a communicational perspective.

Activity 3 consists of identifying business objects[7]. Interviews with stakeholders (mainly representative users) are conducted and an ontological approach is followed. The aim is to identify the static perceptions that the users have of their business elements. During activity 3, business objects are identified and outlined; their detailed description is done during activity 5. Business process indicators are also specified. These indicators allow an organisation to monitor their running and performance (e.g. the state of an order, number of outstanding orders waiting to be served, mean response time to an order). Some business indicators appear during activity 3, others appear during activity 5.

The bidirectional arrow between activities 2 and 3 represents that the identification of business objects may involve the identification of new communicative events and vice versa. Communication Analysis is flexible and adaptable in this sense.

[6] This exemplification is neither big nor detailed, but it is relevant to enterprise information systems; bigger examples are to be offered in future publications.

[7] Herein, we use the term *object* where other authors would use the expression *class of objects*.

L3. Communicative Interaction Level

If the business process analysis done in the level above was intended to discover communicative events, now it is the turn for describing them.

Activity 4 consists of the specification of communicative events. Requirements related to communicative events are further structured in three wide classes: contact, communication and reaction requirements. *Contact requirements* are related to the triggering of the event by an actor who wishes to communicate something to the IS. *Communication requirements* specify the contents of the message being communicated to or from the IS. Practice has shown us that *Communication Structures*, a form of structured text, is convenient to describe messages. Figure 3 shows the Communication Structure associated to the communicative event PRO1. The structure of message fields lies vertically (at the right). Many other details of the fields can be arranged horizontally; e.g. the information acquisition operation, the data domain, the link with the business object, an example value provided by users. *Reaction requirements* are related to the IS reaction to the conveyed message; that is, it has to do with processing the information, updating the system memory and distributing the reaction result to other actors so that they can act accordingly.

Activity 5 consists on the specification of business objects. Business objects are complex aggregates of properties. At this stage, business objects are not specified from a (typical) object-oriented perspective. Object orientation splits business knowledge into homogeneously identifiable aggregates of properties. For instance, users consider an order to be a single business object, but object orientation splits an order into several object classes (e.g. order head and order lines). During activity 5, the structure of properties of each business object is specified. Then, each property (either elementary or complex) is linked with communicative interactions; i.e., the corresponding Communication Structure column is filled (B.OBJECT in Figure 3).

The bidirectional arrow between activities 4 and 5 represents that new business-object requirements may arise from the specification of communicational content, and vice versa.

PRO1 A company requests a project				
FIELD	OPERATION	DOMAIN	B.OBJECT	EXAMPLE VALUE
NEW PROJECT=			PROJECT	
< project code+	g projcode()	char[7]	< code+	P0034/08
request date+	i	date_time	req_date+	24-04-2008
COMPANY=				
< VAT number+	s Company<vat>	char[16]	vat+	19.345.631-Q
company name+	d Company	char[124]		Delicioso Olive Oil, SA
address+	d Company	char[256]		Polígono Sur, prc.64, nº 7
contact person+	i	char[64]	contact+	Sergio Pastor González
contact phone#+	<d Company + i>	char[16]	phone+	963870000 ext 83534
>+				
project description+	i	char[124]	description	Install air conditioning
>			>	unit in a 1000m2 warehouse

LEGEND a=<b+c > aggregation g generation i input s selection d derivation

Fig. 3. Communication Structure associated to the Communicative Event PRO1

L4. Usage Environment Level

In the usage environment, the essential interactions are related to the *editing* of the messages that are associated to communicative events. In order to do so, the user needs to navigate through the interface and to *locate* the corresponding editorial environment, to *trigger* a start signal to begin the editing process, and to trigger an end signal that initiates the expected reaction of the system. Editing, location and trigger are the types of interactions that guide usage requirements elicitation. Also, requirements related to ergonomics are addressed (e.g. layout, look and feel).

Activity 6 consists of capturing usage environment requirements in order to design the interface. Communicative events are restructured in editorial environments since a system interface usually supports several external interactions combinedly (both for interface usability reasons and for component reuse). Each editorial environment is then modelled in terms of abstract interface patterns (e.g. a grid) that are arranged in containers (e.g. a window form). Communication Analysis considers the interface to be a message editor and displayer, and it offers techniques that allow to reason the interface from the Communication Structures. Furthermore, a strong traceability is achieved among the interface patterns and the Communication Structures.

Activity 7 consists of modelling classes of objects. The business objects specified during activity 5 need to be fragmented in order to computerise them. The IS memory is specified in terms of object classes (as object-oriented and relational models do).

The bidirectional arrow between activities 6 and 7 represents that interface design often originates new classes of objects, and that decisions taken during object class modelling influence interface design.

L5. Operational Environment Level

This level is very conditioned to the chosen implementation technology. We only draft the activities associated to this level for the sake of brevity, and because these activities are clearly out of the scope of this paper[8]. This level concerns the design of software components that support the interface, the reaction, and the system persistence. The software architecture is chosen depending on several factors; e.g. technological and budgetary constraints. Lastly, the software is implemented.

At this point in the article, the Communication Analysis workflow (i.e. the activities) and the first level of the underlying requirement structure (i.e. the requirements levels) have been presented. Also, further levels of the requirements structure have been mentioned (e.g. L3: contact, communicational content, and reaction), although not explained in detail due to space restrictions. It is worth insisting that Communication Analysis induces an external viewpoint of the system; the proposed RE techniques elicit (aggregates of) messages/interactions. Next section argues how our proposal contributes to deal with the issues enumerated in Section 2.

4 Qualities of the Proposal

Structure levels L1 and L2 are related to communicational requirements (e.g. balanced scorecards, communicative events). At these levels, the analyst refines sets

[8] To focus the paper on requirements elicitation techniques, other development process activities -such as project management, testing and deployment- are neither addressed.

of communicative interactions (i.e. groups of messages). Level L3 sets the limit for communicative interaction refinement, by offering unity criteria that allow to determine the granularity of communicative events with as much objectivity as possible. At level L3, messages are described. Levels L4 and forth involve taking design decisions; system composition is specified. For instance, communicative interactions are reorganised around editorial environments, which in turn are composed of interface components. In short, each level allows the refinement of upper level requirements and the addition of requirements of that specific level. In the following, it is argued how the issues enumerated in the motivation (see Section 2) can be ameliorated or even solved by following the approach proposed in this paper.

Communication Analysis offers an external point of view; that is, the initial (and main) focus is put on discovering and describing the messages communicated between the IS and its environment (motivating issue 1).

The separation of the requirements associated to the problem space from those of the solution space (issue 2) allows an enterprise to concentrate on specifying the organisational knowledge (the problem), and then to outsource the technological solution. Software development providers can propose interface prototypes taking as input the problem specification and (if there exist) organisational standards regarding interface style guides, usability guidelines, etc. Furthermore, the specification of communicational and informational needs (problem) tends to be stable whereas technological designs and constraints (solution) are usually more volatile. The separation of problem and solution spaces facilitates requirements management, allowing this activity to focus on those requirements that change frequently over time.

Regarding the facilitation of user-developer communication (issue 3), Communication Analysis promotes two communication-enhancing techniques. Firstly, the definition of a glossary of business objects (during activity 3). Secondly, the specification of each communicative event using as input business forms used (or even designed) by users (during activity 4). This way, the user informational model is faithfully represented. See, however, some comments regarding this issue in conclusions and further work (Section 6).

Moreover, the requirements structure and the unity criteria diminish the ambiguity of the concept of requirement and allow an effective methodological guidance for requirements elicitation (issue 6). Our experience with the use of this method has demonstrated that it facilitates the ascription of each discovered requirement to the place in the structure where it belongs, as well as the determination of a proper granularity for the specified communicative interactions.

The concept of communicative interaction and its granularity are maintained at every requirements level. This conceptual homogeneity allows every requirement to be strictly linked to an interaction or to a set of interactions. This prescriptive interrelation in the requirements structure facilitates organising, searching for and tracing requirements (issue 4). The proposed requirements levels allow to search for requirements of a certain systemic level (searching or classifying requirements using a different requirement attribute may need tool support). In current traceability[9]

[9] In this paper, we refer to the aspect of requirements traceability that concerns understanding how high-level requirements are transformed into low-level requirements. It is therefore primarily concerned with the relationships between layers of information [26].

proposals, the interrelation is set a posteriori by means of linking requirement identifiers with structures meant for aggregation [25]. Explicit mappings among problem-space and solution-space requirements come standard in our approach.

Communication Analysis allows forward and reverse elicitation strategies (issues 5 and 7). A *forward* strategy consists of first eliciting input communicative interactions (those that report new information to the IS) and then inferring output communication needs. A *reverse* strategy consists of considering output interactions (e.g. listings and printouts) as complex objects; then the analyst defines derivation formulas until s/he finds the input interactions that provide the information included in the object (or in a part of it). These strategies deserve to be explained in detail in future work.

The method and requirements structure proposed in this paper have been put in practice in several complex and broad scope projects. Communication Analysis has been adopted (a) by the Valencia Port Authority, (b) by the Infrastructure and Transport Ministry of the Valencian Regional Government, (c) and by Anecoop S. Coop.[10]. Anecoop is a second degree cooperative and the major commercialisation agent in the Spanish fruit and vegetables sector. Anecoop intermediates between its associated cooperatives (>100) and its worldwide clients. Therefore, its IS is highly complex, since it has to deal with disparate information needs (e.g. invoices conforming to different tax laws). Communication Analysis has changed the way Anecoop tackles ISs development. Anecoop has attained a successful software development process. A 2-6 people team elicits requirements. All the implementation effort is outsourced. They have implemented a software framework to support their internal business processes and they have also deployed a web service-based B2B solution to work with their associates. At the moment, Anecoop leads a big project aimed to unify the internal processes of their associates. Obviously, this success can not be attributed only to the requirements practice; other practices as project management and risk management have also been undertaken.

5 Related Works

We claim that requirements elicitation and systems analysis methods have to be founded on the knowledge structures that they intend to discover and describe. It is not sufficient to offer a requirements structure without offering a sound method to provide the content, and vice versa. We consider that there exist comprehensive surveys of requirements elicitation techniques [14][27][28][29], so we focus the study of the state of the art on proposals that offer structure and taxonomy for requirements.

Some proposals consider systemic aspects that are characteristic of IS development [30][31][32][33][34][35]. For instance, the following levels of the Requirements Abstraction Model (RAM) [35] are similar to the levels proposed in this paper: Product level and L1; Function level and L3; Component level and L5. The difference between the RAM and our approach is that the RAM has been conceived for market-driven product development and it lacks a proper structure for supporting business process-related requirements. Some proposals offer structure for requirements of an specific level [19][20][35][36]. For example, Volere [20] offers an atomic requirement specification template. Now then, this template is the same independently of the type

[10] (a) Autoridad Portuaria de Valencia; (b) Consellería de Infraestructuras y Transporte.

of requirement. In this paper we propose the use of specific structures and notations for specific types of requirements (e.g. Communication Structures for message-related requirements). Some proposals support refinement [19][20][35].

Information systems must support enterprise communication. However, aspects related to communication are barely present in the above proposals. Some frameworks deal with communication [33], but Communication Analysis confers more pre-eminence to these aspects. In any case, the communicational strategy for IS analysis and design is not an unexplored idea. There exist proposals based on speech act theory [37] and the language action perspective [38] [39]. For example, Cronholm and Goldkuhl propose Communication Analysis as a perspective and method for requirements elicitation [39]. Their proposal coincides with ours in the communicational perspective. With regards to the method, we coincide in highlighting the importance of studying enterprise documents. However, discovery guidelines and the requirements structure are different. Cronholm and Goldkuhl focus their method on enterprise documents and use them as a guide for discovering requirements. The communicative situations in which a document is used are subordinated to the document. Therefore, the requirements structure that they propose (graphically materialised in the Document Activity Diagram) can be said to be document-centred. The method and structure proposed in this paper focuses on the communicative situations (that we refer to as communicative events) and messages are specified by means of Communication Structures. As a result, our requirements structure (graphically materialised in the Communicative Events Diagram) is communicative interaction-centred. Communicative Events Diagram is a revision of business process modelling techniques. The Communication Structures technique is influenced by DeMarco's proposal for data structures in Structured Analysis [40].

6 Conclusions and Future Work

This work is motivated by the inherent complexity of enterprise information systems (ISs) requirements elicitation, and by the lack of requirement elicitation techniques widely-adopted by the industry. We consider that this paper offers interesting insights into ISs requirements. The contribution of this paper is double. Firstly, it presents Communication Analysis, an approach to ISs development that is based on the communications between the IS and its environment. Secondly, it provides a structure for requirements. The requirements structure is founded on systemic concepts; that is, it takes into consideration the diverse systemic levels that are involved in computerised ISs development. Each and every requirement is ascribed to a specific level. Whenever a requirement needs to be refined, this is done in subsequent levels. This way, the structure allows to organise requirements in order to support and facilitate both requirements discovery and description. Also, the requirements structure has a clear communicational perspective. It fits perfectly Communication Analysis. This paper describes the Communication Analysis workflow. The activities are related to their corresponding systemic level. The method is flexible and contingent in the sense that it provides specific techniques to deal both with dynamic aspects (e.g. business processes are viewed as flows of communicative interactions) and with static aspects (e.g. business objects) of the IS. Two modelling techniques for communicative interactions are proposed: Communicative Events Diagram and

Communication Structures. These techniques have been drafted for reasons of available space; they will be tackled in more detail in future publications.

The way in which our proposal satisfies the motivating issues is argued. In any case, we plan to demonstrate the advantages of Communication Analysis empirically. We consider that sharing our industrial experience using the method will be of interest (qualitative research). Also, we are currently carrying out lab experiments to assess specific characteristics of the method, such as the unity criteria (includes quantitative research). Practice has shown us that the communicational perspective facilitates user-developer communication and mutual understanding, but we acknowledge that other influential factors deserve deeper research. E.g. in projects where stakeholders have markedly differing opinions, the analyst must play the role of facilitator.

Currently, several tools that will support Communication Analysis are under development. Also, we are researching the integration with model-based software production methods (e.g. the OO-Method [41]), in order to take advantage of automatic software generation.

References

1. Vernadat, F.: Enterprise Modeling and Integration: Principles and Applications. Springer, Heidelberg (1996)
2. Szegheo, O.: Introduction to Enterprise Modeling. In: Enterprise Modeling: Improving global Industrial Competitiveness, pp. 21–32. Kluwer, Dordrecht (2000)
3. Bubenko, J.A., Kirikova, M.: Enterprise Modelling: Improving the Quality of Requirements Specifications. In: Information Research Seminar IRIS 17, Olou, Finland (1994)
4. Langefors, B.: Theoretical Analysis of Information Systems, Studentlitteratur, Lund (1977)
5. The Standish Group International, Inc.: CHAOS Report 1994 (1994)
6. Glass, R.L.: The Standish Report: Does it Really Describe a Software Crisis? Commun. ACM 49(8), 15–16 (2006)
7. The Standish Group International, Inc.: CHAOS Summary 2008 (2008)
8. Daniels, C.B., LaMarsh, W.J.: Complexity as a Cause of Failure in Information Technology Project Management. In: IEEE Intl. Conf. on System of Systems Engineering (SoSE 2007) (2007)
9. IT Cortex Project Failure Statistics, `http://www.it-cortex.com/Stat_Failure_Cause.htm`
10. Bubenko, J.A.: Challenges in Requirements Engineering. In: 2nd IEEE International Symposium on Requirements Engineering. IEEE Computer Society, Los Alamitos (1995)
11. Juristo, N., Moreno, A., et al.: Is the European Industry Moving Toward Solving Requirements Engineering Problems? IEEE Software 19(6), 70–77 (2002)
12. Kaindl, H., Brinkkemper, S., et al.: Requirements Engineering and Technology Transfer: Obstacles, Incentives and Improvement Agenda. Requir. Eng. 7(3), 113–123 (2002)
13. Morris, P., Masera, M., et al.: Requirements Engineering and Industrial Uptake. In: 3rd International Conference on Requirements Engineering: Putting Requirements Engineering to Practice, Colorado Springs, Colorado, USA. IEEE Computer Society, Los Alamitos (1998)
14. Zowghi, D., Coulin, C.: Requirements Elicitation: A Survey of Techniques, Approaches and Tools. In: Aurum, A., Wohlin, C. (eds.) Engineering and Managing Software Requirements, pp. 19–46. Springer, New York (2005)
15. IEEE: Standard glossary of software engineering terminology IEEE Std 610.12-1990 (1990)
16. Zave, P., Jackson, M.: Four Dark Corners of Requirements Engineering. ACM T Softw. Eng. Meth. 6(1), 1–30 (1997)

17. Davis, A.M.: Software Requirements: Analysis and Specification. Prentice Hall, Englewood Cliffs (1990)
18. Lockemann, P.C., Mayr, H.C.: Information System Design: Techniques and Software Support. In: Kugler, H.-J. (ed.) IFIP 1986. North-Holland, Amsterdam (1986)
19. IEEE-STD-830-1998 Recommended Practice for Software Requirements Specifications (1998)
20. Volere: Requirements Specification Template & Atomic Requirements Template, Atlantic Systems Guild (2006), http://www.volere.co.uk
21. Davis, A.M.: Just Enough Requirements Management, Dorset House (2005)
22. Sager, J.C.: A Practical Course in Terminology Processing, 23 p. John Benjamins Publishing, Amsterdam/Philadelphia (1990)
23. Pastor, O., González, A., España, S.: Conceptual Alignment of Software Production Methods. In: Krogstie, J., Opdahl, A.L., Brinkkemper, S. (eds.) Conceptual Modelling in Information Systems Engineering. Springer, Berlin (2007)
24. Kaplan, R.S., Norton, D.P.: The Balanced Scorecard: Translating Strategy into Action. Harvard Business School Press (1996)
25. Ramesh, B., Jarke, M.: Toward Reference Models for Requirements Traceability. IEEE T Software Eng. 27(1), 58–93 (2001)
26. Hull, E., Jackson, K., Dick, J.: Requirements Engineering. Springer, Heidelberg (2004)
27. Goguen, J.A., Linde, C.: Techniques for Requirements Elicitation. In: IEEE International Symposium on Requirements Engineering, San Diego, CA, USA, pp. 152–164 (1993)
28. Sommerville, I., Kotonya, G.: Requirements Engineering: Processes and Techniques. Wiley, Chichester (1998)
29. Wieringa, R.: A Survey of Structured and Object-Oriented Software Specification Methods and Techniques. ACM Comput. Surv. 30(4), 459–527 (1998)
30. UN/CEFACT's Modeling Methodology - UMM Meta Model - public draft 2.0 (2008)
31. Wieringa, R.: Requirements Engineering: Frameworks for Understanding. Wiley, Chichester (1996)
32. ISO/IEC: 10746-1:1998 Information Technology - ODP Reference Model Overview (1998)
33. Wieringa, R.: Design Methods for Reactive Systems: Yourdon, Statemate and the UML. Morgan Kaufmann, San Francisco (2002)
34. Myers, D., Hathaway, T.: On Business Requirements and Technical Specifications: A Requirements Taxonomy. Requirements Solutions Group LLC (2005)
35. Gorschek, T., Wohlin, C.: Requirements Abstraction Model. Requir. Eng. 11(1), 79–101 (2005)
36. White, S., Edwards, M.: A Requirements Taxonomy for Specifying Complex Systems. In: 1st Intl Conf on Eng of Complex Computer Systems. IEEE Computer Society, Los Alamitos (1995)
37. Falb, J., Popp, R., et al.: Fully-automatic generation of user interfaces for multiple devices from a high-level model based on communicative acts. In: 40th Annual Hawaii International Conference on System Sciences. IEEE Computer Society, Los Alamitos (2007)
38. Dietz, J.L.G.: Enterprise Ontology: Theory and Methodology. Springer, Heidelberg (2006)
39. Cronholm, S., Goldkunhl, G.: Communication Analysis as Perspective and Method for Requirements Engineering. In: Mate, J.L., Silva, A. (eds.) Requirements Engineering for Socio-Technical Systems. Idea Group Inc. (2004)
40. Demarco, T.: Structured Analysis and System Specification. Yourdon Press (1979)
41. Pastor, O., Molina, J.C.: Model-Driven Architecture in Practice: A Software Production Environment Based on Conceptual Modeling. Springer, New York (2007)