Improving Requirements Engineering by Quality Modelling – A Quality-Based Requirements Engineering Framework

Paolo Donzelli

Computer Science Department
University of Maryland, College Park, 20742, MD (USA)
donzelli@cs.umd.edu

Paolo Bresciani

ITC-irst, Istituto per la Ricerca Scientifica e Tecnologica,
Via Sommarive, 38050 Trento-Povo (Italy)
brescian@itc.it

In this paper we describe a structured, goal-oriented, agent-based Requirements Engineering Framework, where quality modelling is adopted to enhance the capability of advanced agent- and goal-based requirements engineering techniques to deal with and resolve soft organizational and system issues. The framework assists and drives analysts and stakeholders towards an early, combined definition of desired system functionality and corresponding quality attributes, by providing an environment within which they can easily cooperate. We present examples from a real project to illustrate our framework.

ACM Classification: D.2.1 (Software – Software Engineering – Requirements/Specifications)

1. INTRODUCTION

Requirements Engineering (RE) (Van Lamsweerde, 2000) is concerned with the definition, analysis, and formalization of the requirements that a potential information system must have to accomplish specific organizational needs. Traditionally, RE has focused mainly on the system and its interactions with the users. Only recently, RE has adopted a broader perspective, i.e., to explicitly take into account the application context of the system during the analysis process (Bresciani *et al*, 2004; Bubenko, 1993; Castro *et al*, 2000; Fickas and Helm, 1992; Offen, 2002; Yu, 1997).

The successful implementation of a new system relies, in fact, on a firm understanding of its application context, and, above all, on our ability to: 1) *transform the needs of the application context into the requirements for the new system*; and 2) *redesign the application context around the new system* to better exploit its capabilities and avoid negative reactions from users.

Both of these are very difficult tasks, for a number of reasons.

First, people are the most valuable asset of an organization, thus, during any innovation process, their needs and perspectives have to be carefully taken into consideration. Put another way, while trying to derive the requirements for a new system or to redesign the application context, the analysts have to be able to clearly identify and address the expectations, the goals, and the possible

Manuscript received: 16 December 2003
Communicating Editor: Didar Zowghi

reactions of all the potential stakeholders (Alexander and Robertson, 2004) who are directly or indirectly involved (e.g., final users, customers, business and information technology managers, system developers, etc.).

Second, the new system and its application context represent two inseparable entities. In other terms, the analysts cannot treat the problem of deriving requirements for the new system, and the problem of redesigning the application context as two separate issues. The new system and its application context are, in fact, so strictly intertwined that they need to be treated and analyzed as a whole—a larger *socio-technical system*—encompassing both the new information system and its application context (Donzelli and Moulding, 1999; Fickas and Helm, 1992; Offen, 2002). The analysts, thus, need to be able to deal with the resulting socio-technical system, to understand its overall needs, and identify the most appropriate *socio–technical requirements* (Dorr *et al*, 2003; Offen, 2002), specifying the desired characteristics of both the technical component (system requirements), and social component (organizational structures and personnel tasks).

Finally, it is important to note that, for socio-technical systems, the designation of the needs (objectives) is in itself problematic, consisting often of “soft”, ill-structured issues (Checkland, 1993; Donzelli and Moulding, 1999), which may be formulated and interpreted differently according to the actors who perceive them. For example, different actors in the same organizational setting could have very different interpretations of the apparently common need of improving the quality of the working environment. The analysts, thus, while trying to derive the requirements for the new system, or to redesign the application context, need to be able to explicitly resolve the softness and uncertainty of socio-technical systems, to produce well-defined and “implementable” socio-technical requirements.

In Requirements Engineering, goal- and agent-based approaches (Anton and Potts, 1998; Bresciani and Donzelli, 2004; Bresciani *et al*, 2001; Bresciani *et al*, 2004; Chung *et al*, 2000; Dardenne *et al*, 1993; D’Inverno and Luck, 1997; Mylopoulos and Castro, 2000; Mylopoulos *et al*, 2001; Yu and Mylopoulos, 1996) have the capability of effectively dealing with socio-technical systems, because the needs emerging from the organizational context can be directly linked to the requirements of the technological system. In fact, when applied to model a socio-technical system, the notions of *Agent, Goal*, and *Intentional Dependency*, allow the analysts to refine (in a smooth and controlled manner) high-level organizational needs into detailed descriptions of the system to be implemented (Bresciani *et al*, 2004; Bresciani *et al*, 2001).

Available agent- and goal-based approaches, however, lack the ability to turn soft issues into precisely defined, agreed-upon, and “implementable” solutions. Although they do provide mechanisms to reason about soft issues and to qualitatively compare alternative solutions, they lack the ability to explicitly “resolve” soft issues into precise requirements.

To address this need, this paper introduces a quality-based Requirements Engineering Framework (REF), explicitly devised to support the requirements engineering process for complex socio-technical systems. Initially introduced to support the requirements engineering process for synthetic environments (Donzelli, 2004; Donzelli and Moulding, 1999), REF has been subsequently applied to organizational information systems (Antonelli *et al*, 2001; Bresciani and Donzelli, 2004; Donzelli and Setola, 2001).

The basic idea behind REF is to combine the advantages provided by the organizational context modelling mechanisms of agent, goal and dependency, and the advantages provided by quality modelling techniques (Basili and Weiss, 1984; Dorr *et al*, 2003; Fenton and Pleeger, 1997; ISO/IEC 9126-1, 2001; McCall *et al*, 1977; Van Solingen and Berghout, 1999). The combination permits the capture of stakeholder perception of quality (soft) issues from the outset of a new project.

This paper is organized as follows. Section 2 outlines the main characteristics of REF, and compares it with related work. Section 3 discusses the need to incorporate a quality modelling approach into REF, while Section 4 outlines the REF application process. Section 5 briefly introduces the case study, an eGovernment project, while Section 6 details the REF application process by showing extracts from the case study. Finally, results are discussed in the Section 7.

2. OUTLINE OF REF AND RELATED WORK

REF has been designed to capture and refine high-level organizational goals, and to transform them into system and organizational requirements, while taking into account the specific needs, experiences, and the perspectives of the involved stakeholders.

In REF, the application context is modelled as a network of interacting *agents* (which may represent any kind of active entity, e.g., teams, humans, machines and software systems, including the target system), collaborating or conflicting in order to achieve both individual and organizational goals. Any agent may generate its own goals, may operate to achieve goals on the behalf of some other agents, may decide to collaborate with other agents for a specific goal, and clash on other ones.

The notion of *agent* allows the stakeholder to be brought into the picture. Because they are modeled as agents in the application domain, stakeholders can visualize their involvement in the new context, and so be in the position of better understanding what could happen with the introduction of the new system, how they might be affected, and what might be required of them. Their reactions and suggestions can be evaluated in advance: domain experts may identify new ways of organizing the application context around the new system to better exploit its capabilities; high-level organizational roles may more easily assess the impact, potential benefits, and risks of the new system; direct users can more concretely think in terms of the functions and properties that the new systems should have in order to be valuable in their activities.

The notion of *goal* allows the analysts and the stakeholders to explicitly model, and reason about, the rationale behind the organization (actual or desired) behaviour. By visualizing goals, REF allows analysts and stakeholders to constantly base their decisions on organizational needs, and, therefore, focus on the "right" organizational system.

In REF, two types of goals are adopted, *hard* and *soft* goals (Chung *et al*, 2000; Donzelli and Moulding, 1999; Yu and Mylopoulos, 1996). A goal is *hard* when its achievement criterion is sharply defined (e.g., the goal "*make the documents available*" is hard, being easy to check whether or not it has been achieved: are the documents available, or not?). Instead, a goal is *soft* when it is up to the goal originator, or to an agreement between the involved agents, to decide when it has been achieved (e.g., the goal "*make the documents easily accessible and promptly available*" is a soft goal: as soon as we introduce concepts such as "easily accessible" and "promptly", different persons usually have different opinions).

Introducing the notion of soft goal increases the flexibility of the decision-making process and our ability of dealing with the fuzziness typical of organizational domains. Soft goals are, in fact, a useful tool to represent (and reason on) soft issues while interacting with the stakeholders.

REF builds on the most recent goal- and agent-based requirements engineering approaches (Anton and Potts, 1998; Bresciani *et al*, 2001; Chung *et al*, 2000; Dardenne *et al*, 1993; D'Inverno and Luck, 1997; Mylopoulos and Castro, 2000). Among these approaches there is *i**, a framework for modelling and redesign intentional relationships among actors in an organizational setting (Yu, 1997; Yu and Mylopoulos, 1996). The differences between REF and other approaches mainly derived from two of its original design goals (Donzelli, 2004; Donzelli and Moulding, 1999).

The first REF design goal was to obtain active stakeholder participation, crucial to the success of the requirements engineering process participation (Alexander and Robertson, 2004).

Like any knowledge elicitation activity, in fact, the requirements engineering process is a cooperative learning process, where all participants learn something through continuous interaction. The analysts learn more about the application domain, stakeholder needs, and the specific application context; the stakeholders learn more about what the new system can help them to achieve, while trying to articulate and make their needs explicit, as well as their knowledge, experience and ideas. The quality of its outcome, thus, strictly depends upon the level of involvement of the stakeholders.

With this perspective, REF attempts to eliminate the barriers that can result from the complexity of the notation or its underlying concepts. In particular, as illustrated in Section 6, REF adopts a very basic notation (especially when compared with similar approaches, e.g., *i** (Yu, 1997; Yu and Mylopoulos, 1996) and F3 (Bubenko, 1993)), and develops through a clear-cut, top-down application process. The trade-off between expressiveness and simplicity is discussed in Bresciani and Donzelli (2004); the advantages of REF simplicity for rapid information system identification are illustrated in Donzelli and Bresciani (2004), while the positive results in terms of stakeholder participation have been demonstrated by various case studies (Antonelli *et al*, 2001; Bresciani *et al*, 2004; Donzelli and Setola, 2001).

The second REF design goal was to enable analysts to turn soft issues into implementable solutions, in order to establish a clear link between the objectives and needs of the encompassing application context (most of which are soft issues) and the proposed system and organizational requirements.

Unlike other approaches, REF emphasizes the operational role that soft goals can play in deriving the requirements of a new system. While soft goals, in fact, are usually applied to support mainly qualitative reasoning between alternatives, REF attempts to go beyond this possibility. For example, in *i** (Yu, 1997; Yu and Mylopoulos, 1996), Tropos (Mylopoulos and Castro, 2000; Bresciani *et al*, 2004), and the non-functional requirements (NFR) framework (Chung *et al*, 2000), soft goals are used to rank (i.e., values range from very satisfying to very dissatisfying), and, hence, compare, different potential system solutions. REF, instead, recognizes the need for explicitly "resolving" soft goals, by turning them into precisely defined, manageable, and "implementable" requirements. For this, REF encompasses a soft goal analysis approach inspired by quality modelling techniques. This is illustrated in the following Section.

3. ENCOMPASSING QUALITY MODELLING

To explicitly resolve a soft goal (i.e., to turn it into well defined requirements), an analyst needs to first identify and extract the hard core of the goal (e.g., the unambiguous goals, actions and objects that may be hidden in the soft goal), and then operationalize the remaining "softness". To clarify this point, let us use a simple example. In particular, let us assume that a document management system has to achieve the following soft goal: "*make the documents easily accessible and promptly available*". Now, we want to see how it can be turned into a set of precise requirements that the system have to satisfy.

First, the analyst has to extract the hard core of the soft goal. In this case, the only item that can be unambiguously defined is the goal "*make the document available*", which in REF can be modeled as a hard goal. Then, the analyst needs to operationalize the remaining softness, i.e., to transform the "purely soft" attributes of "*easily accessible*" and "*promptly available*" into precise requirements.

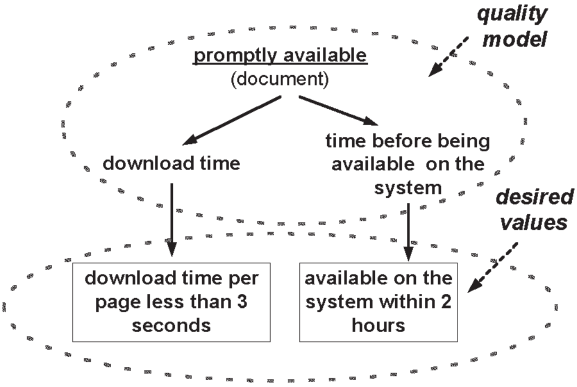

quality model
promptly available
(document)
download time
time before being available on the system
desired values
download time per page less than 3 seconds
available on the system within 2 hours

Figure 1: The Quality Model for "*promptly available*"

Such operationalization requires a two-step process:

1) Step one. The analyst needs to **build a quality model** (Dorr *et al*, 2003; Fenton and Pleeger, 1997; ISO/IEC 9126-1, 2001) of the soft concept, by identifying which characteristics the stakeholders judge important for evaluating it. For example, for evaluating how easily a document is accessible, a stakeholder can identify as relevant specific system characteristics such as the length of the procedure, or the time necessary to learn how to access the documents. Similarly, for judging how promptly a document is available (Figure 1 – quality model), the stakeholder can recognize as relevant the time required for a document to be available on the system once received by the organization, and the time necessary to download it.
2) Step two. The analyst needs to **populate the quality model** to capture the system behaviour which stakeholders desire. In other words, for each system characteristic identified in the quality model, the analyst needs to associate the value desired by the stakeholders. So, for example (Figure 1 – desired values), the stakeholders may require that a document be available on the system within 2 hours, or that the download time per page should be 3 seconds. Similarly, for a document to be easily accessible, stakeholders might require that the average access procedure should not take more than 30 seconds, and that the system should adopt an already known interface which requires no training time.

By populating the quality model, we obtain the set of precise requirements that the document management system will have to satisfy in order to achieve the soft goal: "*make the documents easily accessible and promptly available*". In fact, put in terms of requirements, the hard goal "*make the document available*" leads to the functional requirement "the system shall allow the user to access a document". The operationalization of the soft attributes *easily* and *promptly* (in "*easily accessible*" and "*promptly available*", respectively), performed through the two-step process described above, leads, instead, to a well-defined set of constraints (i.e., non-functional requirements), limiting the way in which the functional requirement could be implemented by the system: a) the system adopts an already known interface; b) the procedure to access a document lasts 30 seconds as average; c) a document is available on the system within 2 hours; d) the download time per page is less than 3 seconds. Such a refinement is schematized in Figure 2.

As it is evident from the previous example, building a quality model is a crucial step in refining a soft goal.

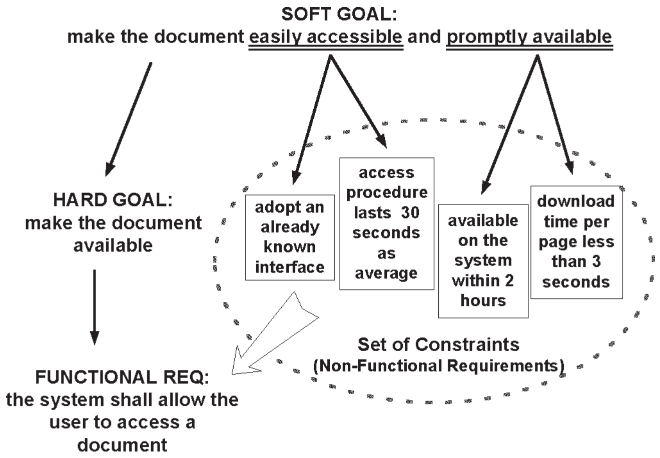

SOFT GOAL:
make the document easily accessible and promptly available
HARD GOAL:
make the document available
FUNCTIONAL REQ:
the system shall allow the user to access a document
adopt an already known interface
access procedure lasts 30 seconds as average
available on the system within 2 hours
download time per page less than 3 seconds
Set of Constraints
(Non-Functional Requirements)

Figure 2: Example of soft goal refinement

One way to build a quality model is to turn to models already available in literature, for example, the McCall model (McCall *et al*, 1977), or the models in the ISO/IEC 9126 standard (ISO/IEC 9126-1, 2001), which provides a framework for the evaluation of software quality by defining six product quality attributes (e.g., reliability, usability, portability, etc.), and some sub-attribute characteristics. The analysts, in fact, can use such models (whole or in part) as example templates or guidelines to refine the soft concepts of interest. As downside, however, they show little flexibility and are not always easy to customize when attempting to accommodate the specific needs of the context under analysis.

Therefore, to refine soft goals REF adopts an approach inspired by a more sophisticated empirical quality modelling/measuring method, i.e., the Goal Question Metric (GQM) approach (Basili and Weiss, 1984; Van Solingen and Berghout, 1999). GQM represents, in fact, a very pragmatic way of dealing with soft issues, providing both analysts and stakeholders with a simple tool for reasoning about soft goals. In addition, as described below, it can be easily integrated with the REF modelling concepts of goal and agent.

First, let us see how GQM can be used to build a quality model. GQM is based on the idea that measures have to be identified starting from the goal that has to be achieved. Once identified, the goal is refined into several questions, to break down the issue into its major components; then, each question is refined into one or more metrics. Put another way, measures are obtained by applying a question-answer mechanism: ask which "Questions" we should be able to answer in order to achieve the "Goal", and then ask what "Metrics" we should apply to be able to answer those questions.

GQM requires that a goal be stated precisely, and provides a template to specify goals of increasing complexity, from the more elementary goal of characterizing, to the more articulated ones of understanding, monitoring, predicting, and improving a software development activity or the related artifacts. For our purposes, we need to focus only on the characterization goal.

To define a *characterization goal*, GQM requires specifying the object of study, the quality attribute, the viewpoint, and the context. Thus, for example, in GQM a characterization goal could be: "characterize how promptly a document is available (the quality attribute) on a document management system (the object of study), for the user (the viewpoint) in our organization (the context)".

According to GQM, now we have to identify the questions we should be able to answer in order to achieve the characterization goal. Possible questions from users in our organization could be:

a) "How long does it take to download a document?" b) "After a document has been received by the organization, how long do I have to wait before it becomes available on the system?". Then, GQM requires the identification of the metrics to apply in order to answer the questions. Possible metrics could be: a) the download time; b) the time before a document is available on the system. By means of the GQM question-answering mechanism, we have identified which metrics on a document management system we should apply in order to characterize how promptly a document is available to users.

It is crucial to note that these metrics (the download time, and the time before a document is available on the system) represent the characteristics of the system that the users judge as relevant to assessing how promptly a document is available. In other terms, they represent the quality model of the soft attribute "promptly available" (compare with Figure 1). Thus, while GQM was devised to support measurement activities, by creating a clear link between the needs of the organization (goals) and the metrics to apply, GQM can also be adopted in requirements engineering to identify the quality model that describes a soft (quality) attribute. The only difference being that, once the set of metrics (quality model) has been identified, it is not applied to measure the object of study i.e. the implemented system. Rather, the stakeholders have to associate to each metric the desired value (or range of values), according to the corresponding measurement scale (e.g., from ratio, to ordinal and nominal as, for example, described in Fenton and Pleeger (1997)). These values are the set of constraints (i.e., non-functional requirements) that operationalize the soft (quality) attribute.

The GQM example application is schematized in Figure 3.

Now, let us see how GQM can be integrated with the REF modelling concepts of goal and agent. REF models the application context as a network of agents interacting by exchanging goals. An agent may generate his own goals and then collaborate with, or delegate to, other agents to achieve them. For each goal, REF clearly identifies all the elements that are required by a GQM characterization goal: the object of study (the object or actions to which the goal refers to), the quality attribute (the soft issue), the viewpoint (represented by the agent or agents involved), and the context (the modeled application context).

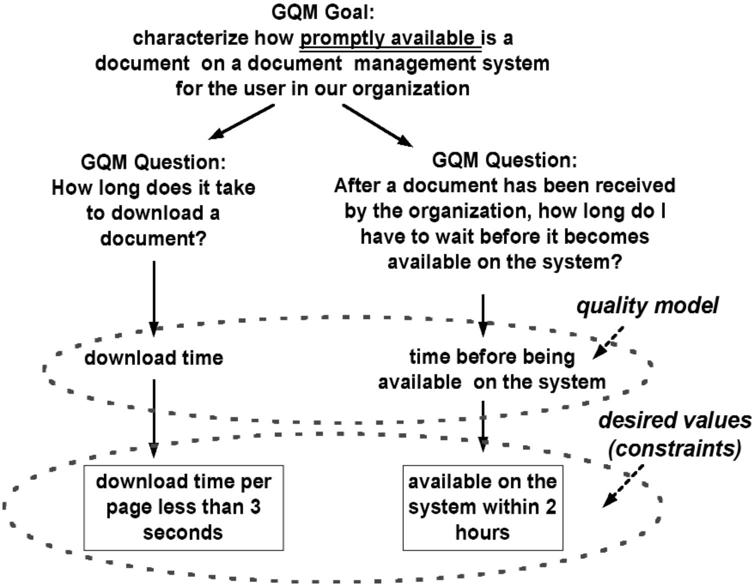

GQM Goal:
characterize how promptly available is a document on a document management system for the user in our organization
GQM Question:
How long does it take to download a document?
GQM Question:
After a document has been received by the organization, how long do I have to wait before it becomes available on the system?
quality model
download time
time before being available on the system
desired values (constraints)
download time per page less than 3 seconds
available on the system within 2 hours

Figure 3: Example application of GQM to build a quality model

REF adopts GQM to obtain quality models tailored to the specific context and stakeholders' needs, suitable to operationalize the soft goals into well-defined constraints. It is important to note that other constructive quality modelling methods have been suggested recently. One is the efficiency modelling method described in Dorr *et al* (2003). This could also be adopted as a guideline in refining soft issues.

4. THE REF APPLICATION PROCESS

The REF application process is designed to support the knowledge elicitation and refinement activities underlying the identification of system requirements starting from high-level organizational needs. Its top-down and cyclical structure has been devised, in fact, to facilitate these tasks by allowing analysts to incrementally acquire more detailed organizational knowledge (Donzelli and Moulding, 1999). The process leads analysts to produce a sequence of artifacts that progressively model the system requirements and its organizational context. Each artifact represents an extension/refinement/integration of the previous ones. The artifacts are of two kinds: organization models and hard and soft goal models.

As shown in Figure 4, REF process begins with a *start-up step*. Here, the analyst has to sketch an initial model of the organization with enough detail to define the scope of the problem, the initial high-level goals, the main agents involved, and their interactions.

Then, the process evolves in a cyclic way through the following two phases:

Goal modelling phase (Figure 4.a), during which the goals (hard and soft) discovered during organization modelling are refined. The refinement is performed by applying, as elicitation technique, a multi-step question-answering mechanism: the analysts repetitively ask the stakeholders (agents) what they need to achieve (subordinate goals), to perform (tasks), or to have available (resources) in

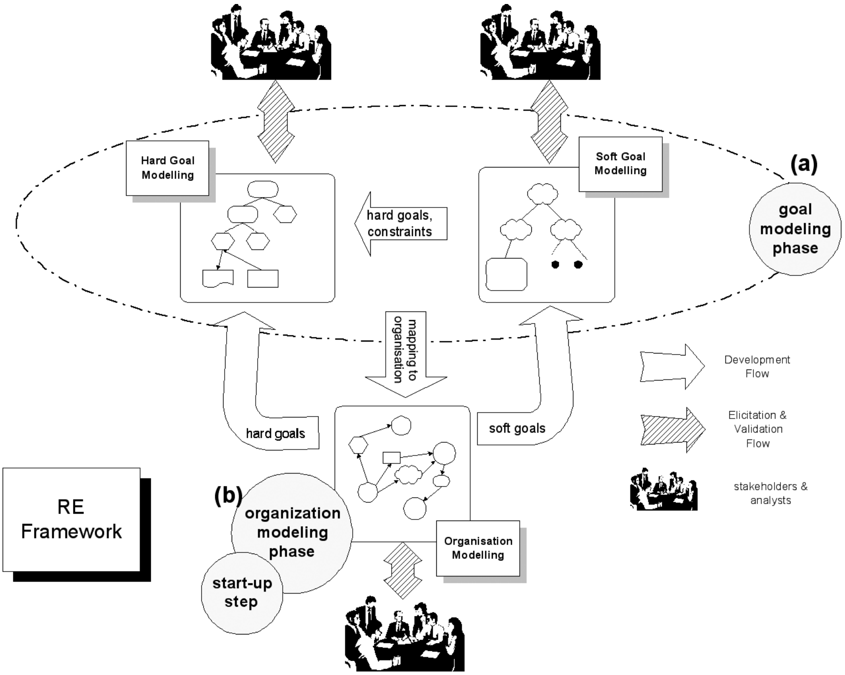


Figure 4: The REF architecture

order to consider the goal as achieved. Beside goals, tasks and resources are two other modelling concepts provided by REF: a *task* represents a well-specified prescriptive activity, while a *resource* represents any physical or logical entity that may be necessary to achieve a goal or to perform a task. While refining the goal, the agents may also identify by whom (or by what) the emerging goals, tasks, and resources should be achieved, performed or delivered, leading in this way to the identification of new agents that should be involved. In particular, hard goals are decomposed into subordinate hard goals, tasks and resources. Soft goals are decomposed into more elementary subordinate soft goals, hard goals, tasks, resources, and constraints. For a soft goal, in fact, the refinement has to be reiterated until its softness has been completely operationalized. In other terms, refinement continues until the soft goal is separated from its functional part (emerging hard goals, tasks, and resources), and its purely soft parts are refined into well-specified constraints, as seen in the previous Section. Constraints are the modelling mechanisms provided by REF to “harden softness”.

Organization modelling phase (Figure 4.b), during which the information gained during the previous phase (see the “mapping to the organization” arrow in Figure 4) is used to enrich and extend the initial organizational model: i.e., to replace the goals with their models, and to introduce the new agents identified as relevant to achieve those goals. New agents usually lead to new goals, triggering the goal-modelling phase again.

The REF cycle is continued until the desired level of detail about the application context is reached. Through continuous interaction with stakeholders, the analyst deals first with the high-level organizational structure, and then descends step-by-step into the details of the application context. Organization Modelling and Goal Modelling, as shown in Figure 4, do not exist in isolation; rather, they are different views of the same modelling effort, linked by a continuous flow of information: the Development and the Elicitation and Validation flows. The Elicitation and Validation flow shows where interaction with stakeholders occurs, whereas the Development flow shows how information discovered in one model may feed the others, in a continuous loop.

5. THE CASE STUDY

The case study reports on a project (Antonelli *et al*, 2001; Donzelli and Bresciani, 2004) concerned with the design, development, and deployment of Electronic Record Management System (ERMS) for a complex public sector administration.

A ERMS is a complex Information and Communication Technology (ICT) system which allows efficient storage and retrieval of document-based unstructured information, by combining classical filing strategies (e.g., classification of documents on a multi-level directory, cross-reference between documents, etc.) with modern information retrieval techniques. It provides mechanisms for facilitating routing and notification of information/document among the users, and supporting interoperability with similar (typically remote) systems, through e-mail and various electronic data interchange standards. An ERMS represents the basic element for a knowledge workplace, i.e., a working environment where a knowledge worker can easily access and gather information, produce knowledge and deliver results through a multitude of channels (from personal computers, to laptops, PDAs, mobile phones, etc.).

Due to the complexity of the system, the decision was made to outsource its development through a public tender, while maintaining full strategic control over the project, in particular, the control over the entire requirements and architectural definition process (Antonelli *et al*, 2001). On the organization side, a small group of highly skilled people (about eight) was committed to the requirements definition task. The activity lasted for about 6 months, before the tender procedures were initiated.

While the requirements were being written in a more traditional textual-based format (required also by the bid regulations), REF was applied as a parallel, but closely related activity, with the two-fold aim of clarifying the requirements and better understanding the impact of the system upon the organization (and changing it accordingly). In particular, a small group of selected stakeholders was involved in drawing, reviewing and refining the REF models (both organization and goal models), prepared by the analysts through a commercial graphical tool.

In the following, we illustrate some extracts from this application of the framework, while introducing in detail the REF notation and discussing some of the already described capabilities.

6. APPLYING REF

As described in Section 4, the REF application process (Figure 4) begins with the start-up step:

6.1 Start-up Step

The aim of the start-up step is to build a model of the organization with enough detail to define the scope of the problem, the initial high-level goals, and the main agents involved. This model is shown in Figure 5. Circles represent agents, and dotted lines are used to bound the internal structure of complex agents, that is, agents containing other agents. In Figure 5, the complex agent *Organization Unit* corresponds to the organization portion into which the ERMS has to be introduced. The *Head of Unit* is the agent, within the *Organizational Unit*, responsible for achieving the required organizational improvement (modelled by the soft goals *exploit ICT to increase performance while avoiding risks*, and *cost/effective and quick solution*).

As mentioned, goals, tasks, resources and agents (see also next Figures) are connected by dependency-links, represented by arrowhead lines. An agent is linked to a goal when it needs or wants that goal to be achieved; a goal is linked to an agent when it depends on that agent to be achieved. Similarly, an agent is linked to a task when it wants the task to be performed; a task is linked to an agent when the agent is committed at performing the task. Again, an agent is linked to a resource when it needs that resource; a resource is linked to an agent when the agent has to provide it. By combining dependency-links, we can establish dependencies among agents. So, for example, a path $A_1 \rightarrow G \rightarrow A_2$ (where A_1 and A_2 are agents and G is a goal) means that the agent A_1 depends on the agent A_2 to achieve the goal G (see also Figures 8 and 10).

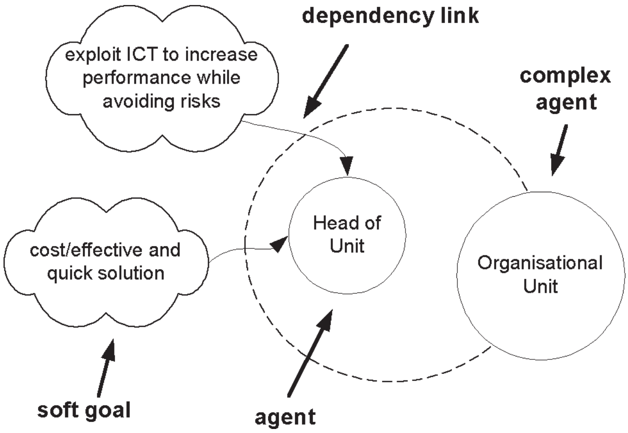

dependency link
exploit ICT to increase performance while avoiding risks
complex agent
Head of Unit
cost/effective and quick solution
Organisational Unit
soft goal
agent

Figure 5: Start-up step: the initial organization model

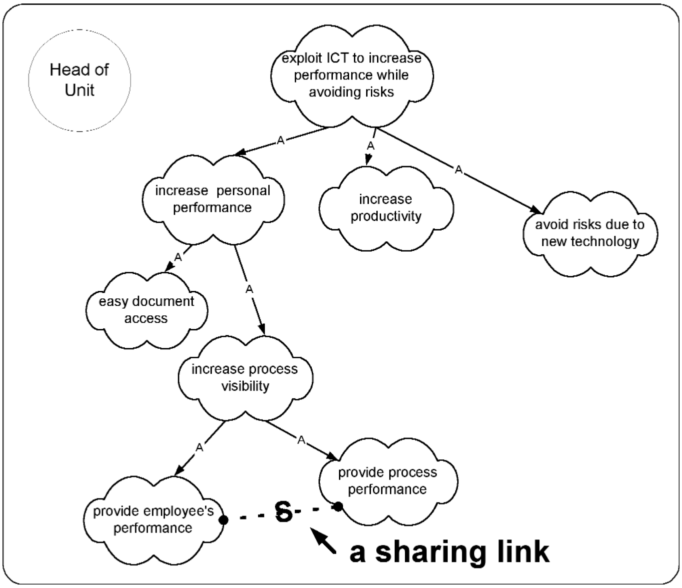

Head of Unit
exploit ICT to increase performance while avoiding risks
A
A
A
increase personal performance
increase productivity
avoid risks due to new technology
A
A
easy document access
increase process visibility
A
A
provide process performance
provide employee's performance
S
a sharing link

Figure 6: The partial "*exploit ICT …*" Soft Goal Model

6.2 Goal Modelling Phase

Once the initial organization model is built, the next step is to model the emerging goals, in this case, the emerging soft goals. In the following, for the sake of the example, we focus on the *exploit ICT to increase performance while avoiding risks* soft goal. As mentioned, the soft goals modelling process allows the analysts and the stakeholders to operationalize all the soft aspects implicitly included in the meaning of the soft goal: to produce a set of tasks, hard goals, and constraints that precisely defines its meaning, i.e., the way to achieve it.

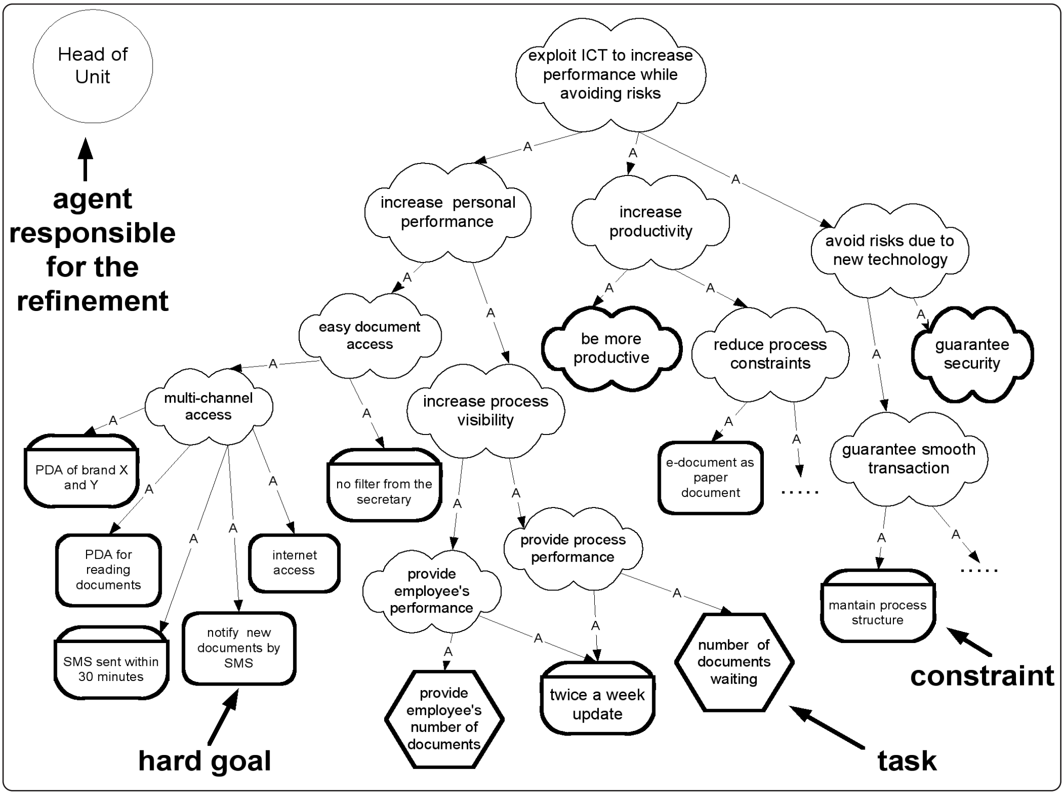
For the sake of clarity, this soft goal refinement has been divided into two figures.

Let us start from Figure 6. Here we can see the strategy that the *Head of Unit* (as result of a personal choice or of a negotiation with the upper organizational level) will apply to achieve the assigned goal. In particular, in order to achieve the soft goal *exploit ICT to increase performance while avoiding risks*, the *Head of Unit* recognizes the need of achieving three more elementary subordinate soft goals: *increase personal performance, increase productivity* (of the whole unit), and *avoid risks due to new technology*. Again, also in a goal model, the arrowhead lines indicate dependency links. A goal depends on a subordinate goal, task or constraint, when it requires that goal, task or constraint to be achieved, performed, or implemented in order to be achieved itself. Goal decomposition may be conjunctive (all the sub-items must be satisfied, to satisfy the original goal), indicated by the label "A" on the dependency link, or disjunctive (it is sufficient that only one of the sub-items is satisfied), indicated by the label "O" on the dependency link (as, for example, in Figure 9).

Each one of the emerging subordinate soft goals can be then further refined. Figure 6 focuses on the soft goal *increase personal performance,* and shows how the *Head of Unit* recognizes the need to further refine it into two more elementary soft goals: to have an *easy document access* and to *increase process visibility*, in order to make better-informed decisions. On its turn, the latter spawns two other soft goals, *provide employee's performance and provide process performance*.

At this point, these two subordinate goals could be refined separately. However, since both are related to performance, it may be that some commonalities are hidden, i.e., that some shared-goals (or tasks, or constraints) may be found during analysis. For this reason, a special link is used in Figure 6: the *Sharing* (*S*) *– connection* (Bresciani and Donzelli, 2004). It is a reasoning support tool that enables the analysts to record their intuitions while building the goals models, for example, to highlight where a top-down breath-first diagram expansion may be preferable to a top-down depth-first strategy. The result of this choice can be seen in Figure 7, where the final and complete model of the soft goal *exploit ICT to increase performance while avoiding risks* is illustrated. In Figure 7, in fact, we see how the simultaneous analysis of the soft goals *provide employee's performance* and *provide process performance*, suggested by the S-connection, has led to the identification of shared constraints (i.e., both data have to be provided twice a week rather than with different frequencies), thus simplifying system requirements. In Figure 7, we also see how the refinement has to be reiterated until the soft goal is operationalized. In other terms, until the hard core of the soft goal is extracted (emerging hard goals, tasks, and resources), and its purely soft components transformed into well-specified constraints. For example, the subordinate soft goal *easy document access* has been transformed into: a) the soft goal *multi-channel access*, then further refined into a set of hard goals and constraints specifying the *Head of Unit* need of having different access channels—with specific properties—to the documents; b) the constraint of having *no filter from the secretary*, for what concerns document access.

Finally, let us observe that some of the items (goals, tasks, and constraints) in Figure 7 are emphasized with a bold outline. These are the items that the *Head of Unit* will pass out, having decided to depend on other agents for their achievement (for example, the soft goal *be more productive* will be passed out to the *Employee* agent, whereas the hard goal *PDA for reading documents* will be passed



Head of Unit
agent responsible for the refinement
exploit ICT to increase performance while avoiding risks
A
increase personal performance
increase productivity
avoid risks due to new technology
easy document access
be more productive
reduce process constraints
guarantee security
multi-channel access
increase process visibility
PDA of brand X and Y
no filter from the secretary
e-document as paper document
.....
guarantee smooth transaction
PDA for reading documents
internet access
provide process performance
provide employee's performance
mantain process structure
SMS sent within 30 minutes
notify new documents by SMS
number of documents waiting
provide employee's number of documents
twice a week update
constraint
hard goal
task

Figure 7: The final "*exploit ICT…*" Soft Goal Model

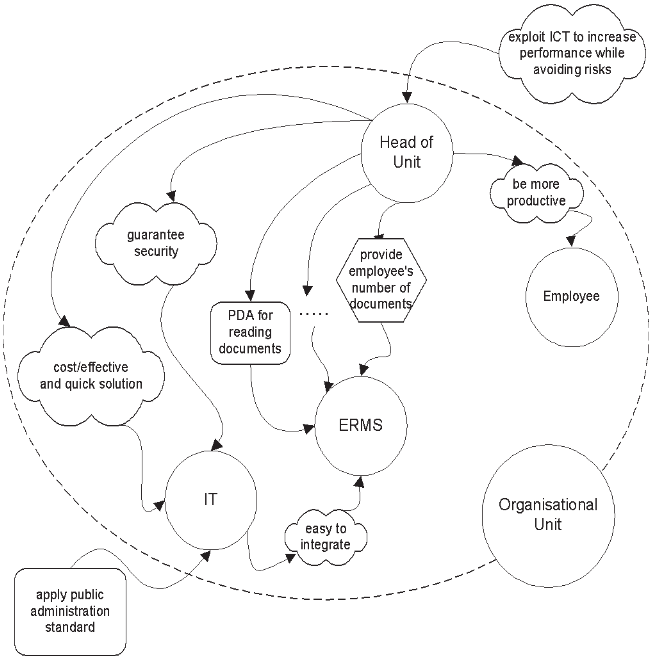

exploit ICT to increase performance while avoiding risks
Head of Unit
be more productive
guarantee security
provide employee's number of documents
Employee
PDA for reading documents
.....
cost/effective and quick solution
ERMS
IT
Organisational Unit
easy to integrate
apply public administration standard

Figure 8: The evolving organization model

out to the *ERMS*, the system to be developed). For this reason, they are not further analyzed; instead they will be refined as further agreement between the *Head of Unit* and the agent that will be appointed of their achievements (as, e.g., shown in Section 6.4). This refinement usually results in the identification of some reciprocal dependencies, making the “contractual” nature of the agreement between social agents explicit. Of course, when the appointed agent is the system, instead of an agreement between peer-level social agents, prescriptive requirements are imposed on the system.

6.3 Organization Modelling Phase

The results of this goal analysis, according to the REF approach, allow us to enrich the initial organization model in Figure 5, leading to the model in Figure 8. Here, some new agents have been introduced: the *Employee*, which has to be more productive, the *Information Technology* unit (*IT* in the figure), which has to guarantee security and the agent representing the final system, the *ERMS*, upon which the identified goals, tasks and constraints will be placed. From Figure 8, we can also see that the *Head of Unit* has decided to delegate the soft goal *cost/effective and quick solution* to the *Information Technology* agent, which, on its turn, will have to achieve other goals coming from the external environment, such as, for example, the hard goal *apply public administration standards*.

6.4 Goal Modelling Phase (again)

To continue our analysis, we have to focus on the emerging goals. Here, for the sake of the example, we will focus only on how the *Employee* will try to achieve the soft goal *be more productive*.

To *be more productive*, the *Employee* defines his own strategy, eventually reaching an agreement with the *Head of Unit*. The soft goal model in Figure 9 shows such a strategy. Here we can see how, in order to *be more productive*, the *Employee* asks that the system be *easy to learn* and *make collaboration easier* with the other employees who deal with the same documents. The *easy to learn*

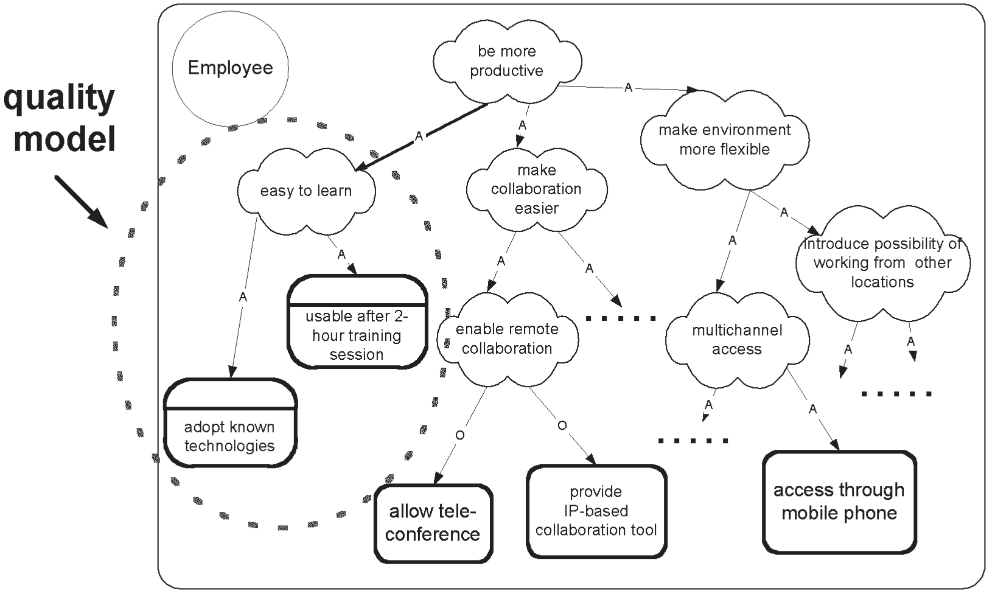

quality model
Employee
be more productive
make environment more flexible
easy to learn
make collaboration easier
introduce possibility of working from other locations
usable after 2-hour training session
enable remote collaboration
multichannel access
adopt known technologies
allow tele-conference
provide IP-based collaboration tool
access through mobile phone
A
O

Figure 9: The "be more productive" Soft Goal Model

soft goal is then refined into a set of implementable constraints, stating, for example, that the system has to *adopt known technologies*, and that has to be *usable after a 2-hour training session*. This subordinate soft goal model can be seen as a "quality model" (see Figure 9) along the lines described in Section 3, representing, in this case, the organizational soft concept of a system easy to learn. The soft goal *make collaboration easier* leads, instead, to the subordinate soft goal *enable remote collaboration*, further refined into a series of hard goals implying specific capabilities (e.g., either a teleconference or an IP-based collaboration tool). Finally, the soft goal *make environment more flexible* will spawn: a) the soft goal *multi-channel access*, modelling the employee need of being able to access the documents with different means, and eventually leading to a number of capabilities that the ERMS will have to provide; but also: b) the soft goal *introduce possibility of working from other locations*. In this latter case, the soft goal does not lead to identifying other technical capabilities of the new system, but it requires the organization to adjust its internal regulations to allow the employees to work from home, or from special sites located in different areas (e.g., easier to reach for commuters). These are a clear example of social requirements affecting organization structure.

6.5 Organization Modelling Phase (again)

The results of Figure 9 may now be used to enrich the organization model of Figure 8, leading to the organization model of Figure 10. This model, for the purposes of our example, can be considered as the final one. Although simplified for the sake of clarity, in fact, the organizational model in Figure 10 describes the operational context within which the final system (the ERMS) and the other agents act (and interact) in order to achieve the initial organizational goals.

Figure 10 also shows how the *Employee*, as any other agent required achieving a goal, may generate his/her own goals. In particular, he/she requires his/her privacy to be protected, by placing the soft goal *protect my privacy* upon the ERMS. This goal provides the occasion to show how REF

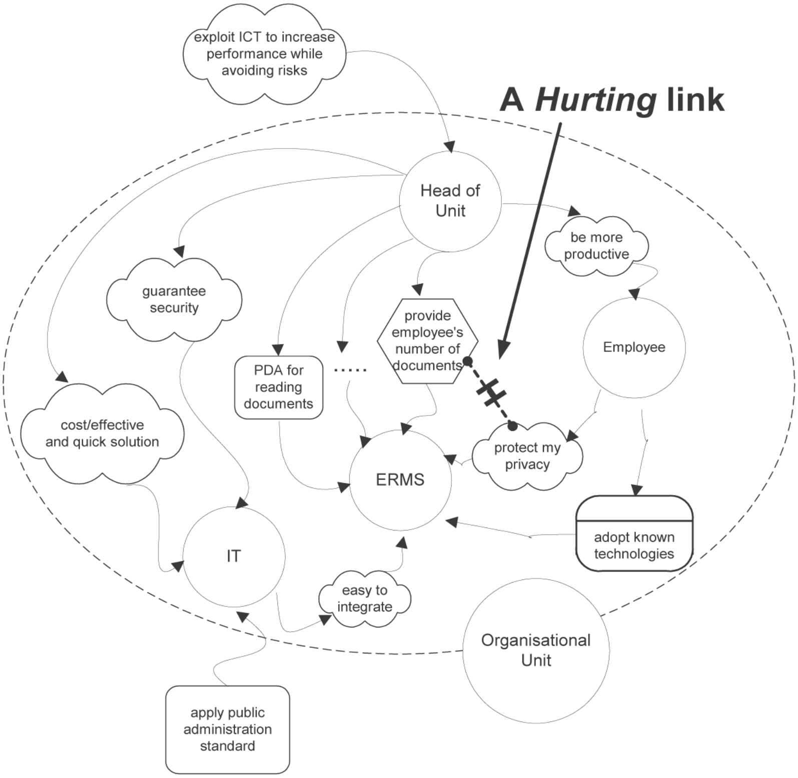

exploit ICT to increase performance while avoiding risks
A *Hurting* link
Head of Unit
be more productive
guarantee security
provide employee's number of documents
Employee
PDA for reading documents
.....
cost/effective and quick solution
protect my privacy
ERMS
adopt known technologies
IT
easy to integrate
Organisational Unit
apply public administration standard

Figure 10: The final organization model

can support detection and resolution of clashing needs. Indeed, through goal modelling, REF allows to recognize such situations. In fact, when fully refined, according to different agents, the resulting goal models allow detecting clashing situations and may support the reconciliation process, providing a basis upon which to identify an agreeable solution (Donzelli, 2004; Donzelli and Moulding, 1999). However to be able to recognize such situations as early as possible, and avoid further goal refinement in the wrong direction, REF provides analysts with the *Hurting (H)-connection* link, to mark possible conflicting situations (Bresciani and Donzelli, 2004). In Figure 10, the H-connection is used to highlight the possibility of a conflict between the *Employee's* soft-goal *protect my privacy* and the task *provides employee's number of document*, required by the *Head of Unit* (see Figures 7 and 8). On the basis of this annotation, the analyst carries on the refinement of the soft-goal *protect my privacy* by bearing in mind possible conflicts with the task, for example by explicitly submitting the issue to *Employee* attention, to evaluate whether providing the number of documents he/she is dealing with represents or not a breach of privacy. If turns out to be the case, the *Employee* and the analyst may agree on a different solution, for example for the ERMS to provide only the average number of documents; and then suggest such a solution to the *Head of Unit*.

Finally, Figure 10 highlights the hard goals, the tasks, and the constraints placed upon the *ERMS* by other agents. Collectively, these hard goals, tasks and constraints form the initial set of requirements for the ERMS. At this point, the analysts will have to decide whether carrying on requirements refinement by using REF or by adopting a different (and maybe more suitable) technique, for example a UML-based approach, as discussed for example in Donzelli (2004).

7. CONCLUSIONS AND RESULTS DISCUSSION

The case study showed how REF provides valuable support during the requirements engineering process. The framework allowed for active stakeholder participation, and each type of REF model (soft goal, hard goal and organization models) provided a specific knowledge representation vehicle for supporting analysts in dealing with stakeholders. In particular, hard goal models have allowed the stakeholders to make explicit the tasks they have to perform and the resources they need. Soft goal models brought quality issues into the picture, and supported the stakeholders while reasoning about their own concepts of quality, by highlighting possible conflicts, and supporting negotiation towards a feasible solution. Thus, while soft goal and hard goal models, as a whole, lead toward system and organizational requirements, soft goal models also allowed (as by-product) analysts to freeze acquired knowledge and produce quality models of the attributes of interest (for example, the easy to learn attribute in Figure 9). The organization model, finally, provided management with a clear view of how the business process would change, or be affected by the introduction of the new system.

The results are consistent with those provided by other case studies, where REF was applied to identify the requirements of complex simulation systems (Donzelli, 2004; Donzelli and Moulding, 1999), an organizational smart-card system (Donzelli and Setola, 2001), or a workflow-based system (Antonelli *et al*, 2000). This greatly increases our confidence in the capability of the framework and in the benefits it could provide to the requirements engineering process.

However, we are also aware of some limits to the practical application carried out up until now. In particular, REF has been applied by a small group of highly skilled analysts and the stakeholders were members of the same organization, and probably more willing to cooperate.

For these reasons, to be able to collect more general results, we are conducting further and different experiments. Initial, satisfactory results have already been obtained, for example, by applying REF as a knowledge-management tool for electronic government, in particular for the extraction and analysis of the knowledge embedded in a set of local laws (Bresciani *et al*, 2004).

REFERENCES

ALEXANDER, I. and ROBERTSON, S. (2004): Understanding project sociology by modeling stakeholders, *IEEE Software*, IEEE Computer Society press, 21(1): 23–27.

ANTON, A. and POTTS, C. (1998): The use of goals to surface requirements for evolving systems, *Proceedings of the 20th International Conference on Software Engineering* (ICSE), Kyoto, Japan, 19–25 April.

ANTONELLI, C., DONZELLI, P., MASTROIANNI, C., SETOLA, R., VINTI, S. and TUCCI, S. (2000): A web-based workflow solution to support the Italian government agenda definition process, *Proceedings of the Italian Automatic Computation Association* (AICA) 2000 Conference, Taormina, Italy, 27–30 September.

ANTONELLI, C., DONZELLI, P., GANDINI, E., TUCCI, S. and VINTI, S. (2001): An enterprise information system for the Italian cabinet office: a scenario, *Proceedings of the Italian Automatic Computation Association (AICA) 2001 Conference*, Como, Italy, 19–22 September.

BASILI, V.R. and WEISS, D.M. (1984): A methodology for collecting valid software engineering data, *IEEE Transactions on Software Engineering*, 10(6): 728–738.

BRESCIANI, P. and DONZELLI, P. (2004): The agent at the center of the requirements engineering process—a framework for complex socio-technical systems, in *Software Engineering for Multi-Agent Systems II*, LUCENA, C., GARCIA, A., ROMANOVSKY, A., CASTRO, J. and ALENCAR, P.S.C. (Eds.), Springer Verlag, LNCS 2940: 1–18.

BRESCIANI, P., PERINI, A., GIUNCHIGLIA, F., GIORGINI, P. and MYLOPOULOS, J. (2001): A knowledge level software engineering methodology for agent oriented programming. In *Proceedings of the Fifth International Conference on Autonomous Agents*, Montreal, Canada, 28 May–1 June.

BRESCIANI, P., DONZELLI, P., FERRARI S. and SANNICOLO F. (2004): Knowledge elicitation and modelling for Egovernment – a practical approach. 5th IFIP International Working Conference on Knowledge Management in Electronic Government (KMGov 2004), Krems, Austria, May 17–19, LNAI, 3035: Springer-Verlag.

BRESCIANI, P., GIORGINI, P., GIUNCHIGLIA, F., MYLOPOULOS, J. and PERINI, A. (2004): TROPOS: An agent-oriented software development methodology. Autonomous Agents and Multi-Agent Systems, 8(3): 203–236. Kluwer Academic Publishers.

BUBENKO J.A. (1993): Extending the scope of information modelling, *Proceeding of the 4th International Workshop on the Deductive Approach to Information Systems and Databases*, Lloret-Costa Brava, Spain, 20–22 September.

CASTRO, J., KOLP, M. and MYLOPOULOS J. (2000): Developing agent-oriented information systems for the enterprise. In *Proceedings Third International Conference on Enterprise Information Systems*, Stafford UK, 4–7 July.

CHECKLAND, P. (1993): System thinking, system practice, John Wiley and Sons, England.

CHUNG, L.K., NIXON, B. A., YU, E. and MYLOPOULOS, J. (2000): Non-functional requirements in software engineering. Kluwer Publishing.

DARDENNE, A., VAN LAMSWEERDE, A. and FICKAS S. (1993): Goal directed requirements acquisition. *Science of Computer Programming*, 20(1–2): 3–50.

D'INVERNO, M. and LUCK, M. (1997): Development and application of an agent based framework. *Proceedings of the First IEEE International Conference on Formal Engineering Methods*, Hiroshima, Japan, 12–14 November.

DONZELLI, P. (2004): A goal-driven and agent-based requirements engineering framework. *Requirements Engineering*, Springer-Verlag London Ltd, February 9(1): 16–39.

DONZELLI, P. and BRESCIANI, P. (2004): Domain visualization for dealing with complex IT systems, *Proceedings of the 9th IEEE International Conference on Engineering of Complex Computer Systems* (ICECCS 2004), 14–16 April, Florence, Italy.

DONZELLI, P. and MOULDING M. (1999): Developments in application domain modelling for the verification and validation of synthetic environments: A formal requirements engineering framework. In *Proceedings of the Spring 99 Simulation Interoperability Workshop*, Orlando, FL,14 March.

DONZELLI, P. and SETOLA, R. (2001): Putting the customer at the center of the IT system – a case study, *Proceedings of the EuroWeb 2001 Conference – The Web in the Public Administration*, Venice, Italy, 18–20 December.

DORR, J., KERKOW, D., VON KNETHEN, A. and PAECH, B. (2003): Eliciting efficiency requirements with use cases, In *Proceedings of the International Workshop on Requirements Engineering: Foundations of Software Quality* (REFSQ'2003), Klagenfurt, Austria, 16–17 June.

FENTON, N. E. and PLEEGER, S. H. (1997): Software metrics: A rigorous and practical approach – second edition, International Thomson Computer Press, UK.

FICKAS, S. and HELM, B.R. (1992): Knowledge representation and reasoning in the design of composite systems, *IEEE Transactions on Software Engineering*, June 18(6): 470–482.

ISO/IEC 9126–1 (2001): Software engineering – Product quality – Part 1; Quality model.

McCALL, J., RICHARDS, P. and WALTERS, G. (1977): Factors in software quality, *Technical Report 77CIS02*, Rome Air Development Center.

MYLOPOULOS, J. and CASTRO, J. (2000): Tropos: A framework for requirements-driven software development. In BRINKKEMPER, J. and SOLVBERG, A. (eds), Information System Engineering: State of the Art and Research Themes, Lecture Notes in Computer Science. Springer-Verlag.

MYLOPOULOS, J., CHUNG, L., LIAO, S., WANG, H. and YU. E. (2001): Exploring alternatives during requirements analysis. *IEEE Software*, 18(1): 92–96.

OFFEN, R. (2002): Domain understanding is the key to success system development, Requirements Engineering, Springer Verlag London, 7(3): 172–175

VAN LAMSWEERDE, A. (2000): Requirements engineering in the year 00: A research perspective. In *Proceedings of the 22nd International Conference on Software Engineering*, invited Paper, ACM Press, Limerick, Ireland, 4–11 June.

VAN SOLINGEN, R. and BERGHOUT, E. (1999): The goal/question/metric method, McGraw-Hill, UK.

YU, E. (1997): Why agent-oriented requirements engineering. In *Proceedings of 3rd Workshop on Requirements Engineering For Software Quality*, Barcelona, Spain, 16–17 June.

YU, E. and MYLOPOULOS, J. (1996): Using goals, rules, and methods to support reasoning in business process reengineering. *International Journal of Intelligent Systems in Accounting, Finance and Management*, 5(1):1–13.

BIOGRAPHICAL NOTES

Paolo Donzelli

Paolo Donzelli is a visiting senior research scientist with the Computer Science Department at the University of Maryland. Dr Donzelli is a director with the Department of Innovation and Technology of the Office of the Prime Minister in Italy, and is currently working at UMD on sabbatical leave. Formerly, he served as an engineering officer with the Italian Air Force, and then he was senior research fellow with Cranfield University, UK. He received an electronic engineering Laurea degree cum laude from the University of Naples Federico II, Italy, a MSc from Cranfield University, and a PhD from the University of Rome TorVergata, Italy. His research interests include software process improvement, requirements engineering, and dependability modelling and validation.

Paolo Bresciani

Paolo Bresciani received his Laurea degree cum laude from University of Milan, Italy, in 1985. Currently, he is a research scientist at the ITC-irst (Centre for Scientific and Technological Research) of Trento, Italy. As well, he had and has several teaching contracts with the University of Verona and the University of Trento. He participated in several national and European Community projects. His research interests lie in the areas of knowledge representation and reasoning, conceptual modelling, semantics-based interfaces to information systems, knowledge management, and agent- and goal-based software engineering and requirements engineering. He is a member of several program committees of international conferences and workshops. He is one of the organizers of the Third International Workshop on Agent-Oriented Methodologies at OOPSLA 2003, and is co-chair of the International Workshop on Agent Oriented Information Systems at CAiSE 2004.