Goal-Oriented Requirements Engineering: a Case Study in E-Government

Paolo Donzelli[1] and Paolo Bresciani[2]

[1] Department of Innovation and Technology
Italian Cabinet Office
Via Barberini 38, I-00187 Roma (Italy)
p.donzelli@governo.it
[2] ITC-irst
Via Sommarive 18, I-38050 Trento-Povo (Italy)
bresciani@itc.it

Abstract. This paper presents a requirements engineering framework based on the notions of *Actor*, *Goal*, and *Intentional Dependency*, and applies it to a case study in the field of Information Systems for e-Government.
The framework provides the analyst with a powerful tool for capturing high-level organizational needs and transforming them into system requirements in a smooth and controlled manner, and for redesigning, at the same time, the organizational structure that better exploit the new system. Organizations are modeled using *Actors*. Thus, a network of interacting actors form the organization model. These actors collaborate or conflict in their efforts to achieve individual or organizational goals. *Goals* represent relationships between actors and link organizational needs to system requirements.
In comparison with other more articulated actor- and goal-centered approaches to requirements and software engineering, the proposed framework adopts only a basic set of notational elements, and introduces a simplified, top-down, decomposition-based, analysis process. This approach greatly enhances stakeholders' acceptance and understanding, that are crucial factors for the success of real projects, especially when very diverse stakeholders, with very different skills and backgrounds, are involved, as in the case of e-Government projects.
The definition of the requirements for an Electronic Record Management System for e-Government is used to illustrate the framework.

1 Introduction

In Requirements Engineering (RE), Goal and Actor orientation has been recognized as an approach more promising than other system- and functionality-based techniques used in most of the traditional Software Engineering methodologies [7, 13, 1, 2, 17, 18, 6, 15]. By adopting the notions of *Actor*, *Goal*, and *Intentional Dependency*, it is in fact possible to refine high-level requirements originating from the organizational setting (i.e., stakeholders' needs and desires) into detailed descriptions of the system to be implemented (in terms of architecture, components, and functions), in a smooth and controlled manner, particularly if the target programming paradigm is an agent-oriented one [14, 4, 5], but not only [16].

This paper presents a requirements engineering framework (called *REF*), by means of its sample application to a case study in e-Government.

REF is designed to deal with, and reason about, socio-technical systems. It is a powerful tool that allows the analyst to model high-level organizational needs and to transform them into system requirements, while redesigning the organizational structure to better exploit the new system. The modeling effort breaks activities down into more intellectually manageable components on the basis of common conceptual notations. To this end, REF introduces the notions of *Actor*, *Goal*, and *Intentional Dependency* as modeling elements used throughout requirements acquisition, formalization and analysis, to capture and describe both the new system and the organizational context in which the system will operate. The framework uses *Actors* (any kind of active entity, e.g., teams, humans and machines, including the target system) to model the organization [9, 20]. As well, it uses *Intentional Dependencies* among actors to describe the organizational context in terms of a network of interacting actors collaborating or conflicting in their efforts to achieve both individual and organizational goals. Finally, REF uses *Goals* [9, 20, 7] to model actors' relationships (in terms of intentional dependencies), and, eventually, to link organizational needs to system requirements.

Describing both the requirements of the system-to-be and the encompassing organizational context by means of the same notions, not only allows us to verify the fulfillment of the organizational needs very directly, but also allows for a high stakeholders' involvement.

The paper is organized as follows. Section 2 describe the reasons for using REF in Requirements Engineering for e-Government applications. Section 3 introduces the case study. The applications of REF to the case study is then described (Section 4), including the use of interesting and novel high-level features (Section 5). Conclusions are drawn in Section 6.

2 REF and E-Government Applications

E-Government means to exploit Information and Communication Technologies (ICT) to provide in a more efficient way higher quality services to the government customers (citizens and businesses), mainly through electronic delivery channels (from Internet-based channels, to digital TV, mobile phone, etc.).

E-Government is a particularly interesting and challenging sector for RE, in which it is relevant to adopt a methodology capable of taking into account different actors, their points of view, and the resulting strategic dependencies. In this context, in fact, very diverse kinds of actors (e.g., citizens, employees, administrators, politicians and decision-makers in general —both at central and local level) are involved, each of them with its objectives and goals. Some goals may be quite straightforward (as, e.g., for a citizen, to be able to require on-line a new passport), others rather complex and articulated (as, e.g., for an administrator, to be compliant to laws and norms), and sometimes they may diverge quite considerably. For example, to have a passport so easily delivered, the administration has to be able to perform cross-checks of personal data usually managed by different (possibly decentralized) units, and to enable mechanisms for secure identification (e.g., digital signature) and transmission. Thus, in general, e-Government applications have to operate in a social environment characterized by a rich tissue of actors with strong inter-dependent intents. Due to this complex network of interrelated objectives, synergies and conflicts may be present. Being able to clearly identify the set of involved actors, their objectives (i.e., goals), and the way they depends on each other in order to achieve such goals, most likely by exploiting possible synergies or trying to avoid potential conflicts, is of utmost importance to obtain a clear and complete comprehension of the organizational setting into which the new technology should be introduced. And only from such a deep comprehension of the application context the correct system requirements can be derived, and, consequently, the correct system can be designed and implemented. Thus, a RE methodology satisfying this perspective must be capable of describing both the requirements of the system and its social context. These descriptions must share a common notation, in order to be able to strictly connect the system requirements to the real organizational needs the organization has, and to easily evaluate the possible impact the system may have on the way of acting of the organization and its actors.

REF tries to provide such a capability by adopting a diagrammatic notation for describing the organizational settings: this notation immediately conveys the intentional dependencies among the different actors, and allows for a detailed analysis of the goals upon which the actors depend, through a goal decomposition process. Moreover, the notation is used to describe not only the organizational setting but also the system itself, as an (artificial) actor placed in the context of the organization. Thus, *Actors*, in REF diagrams, may represents social entities (both individuals and organizations, e.g., enterprises, departments, offices) as well as artificial elements of the organization (e.g., pre-existing systems and, of course, the target system).

Another very important notion, in the REF notation, is that of *Goal*. *Goals* represent states of the world that are desired by one (or more) actor(s), and for the achievement of which an actor may depend on other actors. REF adopts a semi-formal approach for describing actors and goals, based on precisely defined typed graphs, in which nodes are labeled with natural language sentences that intuitively describe their semantics. Of course, some level of common understanding, shared by the analyst and the stakeholders, is assumed.

The REF notation is widely inspired by the i* framework [19] for RE [20] and business analysis and re-engineering [21]. Moreover, REF introduces also a clear methodology to drive the process of requirements discovery, definition, refinement and reconciliation [9]. An important difference between REF and i* is that only the most basic and essential notational ingredients of i* are adopted in REF [9]. Another important simplification is introduced by imposing a strict top-down approach to the process of goal analysis. These choices, although apparently constraining, results to be quite successful in practical terms. Several case studies [10, 11] demonstrate, in fact, that the simplified notation and process facilitates the acceptance of REF by the stakeholders, and contributes to a quicker introduction of the methodology in the RE process.

As in i* [19, 21] and in other derived methodologies, like, e.g., Tropos [14], a distinction is made in REF between soft and hard goals [11]. *Soft-goals* are used to specify, at a qualitative level, not sharply-cut objectives, the precise definition of which require to develop further details, while *hard-goals* clearly define a state/target, an actor desires to reach. For example, “having a passport delivered” is clearly an hard-goal, while “having it delivered quickly” is a soft-goal, being the notion of “quickly” highly subjective.

3 The case study

The case study reports on an on-going project aiming at introducing an Electronic Record Management System (ERMS) into the administrative processes of the Italian Cabinet Office, to transform a huge repository of documents (from decrees, to tenders related documentation) into a ready available source of knowledge to be shared among all the actors acting within the organization. The impact of such a system on the common practices of the communities and the sub-communities of knowledge workers who will adopt it is quite relevant. Indeed, ERMS is at the moment used by more than 300 employees and handles a flow of about 200.000 document/year, but it is expected to reach about 2000 users and 2 million documents/year.

ERMS is based on the adoption of complex ICT solutions which allow 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. Moreover, it encompasses mechanisms for facilitating routing and notification of information and documents among the users, and supporting interoperability with similar (typically remote) systems, through e-mail and XML. It represents the basic elements 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.).

Accordingly to the REF process, a first organization diagram describing the original organizational setting before the introduction of the ERMS was produced (see Figure 1). In REF diagrams, circles represent *actors*, and dashed ovals are used to bound the internal structure of complex actors, i.e., actors containing other actors. As well, rounded boxes represent *hard-goals* and clouds represents *soft-goals*. Goals are always connected with arrows to (one or more) actors: an incoming arrow means that that goal is desired, wanted, or needed by the connected actor. An out-going arrow means that there is a dependency, on the connected actor, for the fulfillment of the goal. Thus, in the most general case in which an actor A is connected to a goal G that is connected to another actor B, we have that A (who wants G) depends on B for G to be fulfilled. In a very similar way, *resources* (rectangles), and *tasks*[3] (hexagons), may be represented. An actor is connected to a task when it wants the task to be performed; a task is linked to an actor when the actor is committed at performing the task. An actor is linked to a resource when it needs that resource; a resource is linked to an actor when the actor has to provide it. In Figure 1, the complex actor Organizational Unit corresponds to the organizational fragment into which it is planned to introduce the new ERMS, whereas the Head of Unit, the Secretary, the Employee, the Archivist, the Personal Computer and the Physical Archive are simple actors, acting within the Organizational Unit.

The Secretary receives from the enclosing context (out of the diagram scope) the input documents, which then she passes to the Head of Unit. That is, the Head of Unit

[3] A task is a well specified prescriptive activity.

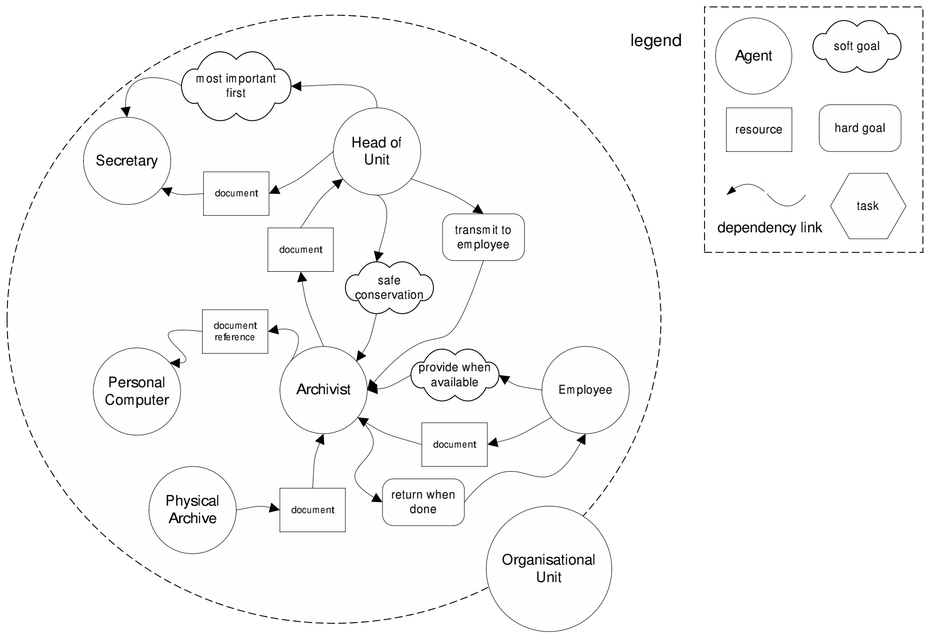

Fig. 1. The organizational context before the ERMS

depends on the Secretary for receiving the document (a resource) and for the qualitative constraint: most important first (a soft-goal). It is up to the Head of Unit to decide to which employee the document has to be assigned (transmit to employee). For this, and for the safe conservation of the document, she depends upon the Archivist. On its turn, the Employee depends upon the Archivist to "provide the document as soon as available" (the soft-goal provide when available). The complete description of the diagram can be found in [11].

The organizational unit before the introduction of the ERMS works in a simple way, with a flow of documents between the actors and a set of responsibilities very well defined, as quite intuitively captured in Figure 1; nevertheless, several factors (international benchmarking studies, citizens demand, shrink budgets, etc.) called for the decision of leveraging new technologies to transform the organization's bureaucratic structure into a more creative and knowledgeable environment.

This led to the definition of ERMS requirements, as discussed next.

4 Applying REF

The first step towards the identification of the requirements for the ERMS, together with the new organizational setting suitable to exploit ERMS capabilities, is to produce a new organization diagram, capturing the motivating pushes underlying the project. Figure 2 represents again the complex actor Organizational Unit and the actor Head of Unit. As

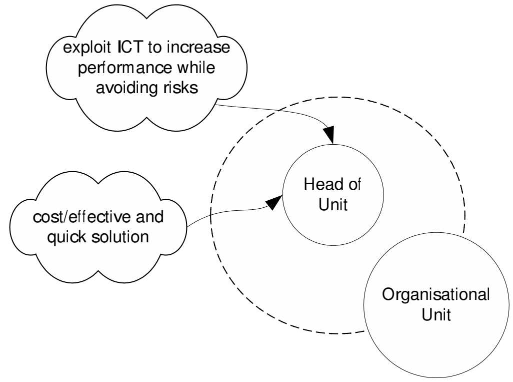

Fig. 2. Introducing the ERMS: the initial organization model

well, new elements are introduced: the soft-goals exploit ICT to increase performance while avoiding risks, and cost/effective and quick solution, that represent the required organizational improvement that the Head of Unit is responsible to achieve.

REF models are introduced and refined along an incremental and iterative analysis process, including three distinct phases: *Organization Modeling*, *Hard-Goal Modeling*, and *Soft-Goal Modeling*. This helps to reduce the complexity of the modeling effort.

During *Organization Modeling*, the organizational context is analyzed and the actors and their hard and soft goals identified. Any actor may generate its own goals, may operate to achieve goals on the behalf of some other actors, may decide to collaborate with other actors for a specific goal, or to delegate it, and might clash on some other ones. The resulting goals will then be refined, through interaction with the involved actors, by hard and soft goal modeling. The diagrams presented so far are example of output of the organization modeling phase.

The *Hard-Goal Modeling* phase seeks to determine how the actor can achieve an hard-goal placed upon it, by decomposing it into more elementary subordinate hard-goals and tasks.

The *Soft-Goal Modeling* aims at producing the operational definitions of the soft-goals that emerged during the organizational modeling, sufficient to capture and make explicit the semantics that are usually assigned implicitly by the involved actors [3, 6] and to highlight the system quality issues from the start. A soft-goal is refined in terms of subordinate soft-goals, hard-goals, tasks and constraints, where *constraints* (represented by a rounded box with a line) are associated with soft-goals to specify the corresponding quality attributes. So, for example, the soft-goal "make a document easily and promptly available", beside spawning the hard-goal "make a document available", will lead also to a set of constraints (e.g., types of access channels, number of hours after which a document is available, etc.) specifying the concepts of "easy" and "prompt". Thus, soft-goals become a knowledge representation vehicle that: 1) encourages the interaction between the analysts and the stakeholders; 2) leads towards a common terminology; 3) supports reasoning about trade-offs; 4) allows freezing temporary solutions, and formalizing final decisions. Soft-goal models, in addition, allows the analysts and the stakeholders to early detect synergic as well as clashing requirements, which usually are hidden behind generic and left implicit assumptions, providing, at the same time, an operational and cooperative way to better exploit or resolve them, by coordinating or reconciling the different stakeholders' points of view.

According to the REF process, the analysts, through continuous interaction with the stakeholders, and supported by the different models, will deal first with the high level organizational structure, then will descend step by step into the details of the application context of the new system, until the focus will be placed upon the single actors and their role within the organization. In particular, once an initial model of the organization is built (as in Figure 2), the REF process evolves in a cyclic way through the phases of hard and soft goal modeling, and of organization modeling. Hard and soft goal modeling refines the goals (both hard and soft) discovered during organization modeling; as well, output from goal modeling is used to enrich and extend the initial organizational model: the new actors so introduced may lead to new hard and soft goals, triggering goal modeling again. The two activities of hard and soft modeling are, of course, tightly connected and, at least to some extent, may be developed in parallel.

Thus, on the basis of the organization model in Figure 2, the analysis may proceed by modeling the emerging soft-goals. An example of the result of a soft-goal modeling activity is presented in Figure 3. The figure describes how the soft-goal exploit ICT to increase performance while avoiding risks is iteratively top-down decomposed to finally produce a set of tasks, hard-goals, and constraints that precisely defines the meaning of the soft-goal, i.e., the way to achieve it.

Figure 3, in other terms, gives the operational elements that the Head of Unit will apply to achieve the assigned goal. Again, the arrowhead lines indicate dependency links. A soft-goal depends on a subordinate soft-goal, hard-goal, task, resource or constraint, when it requires that goal, task, resource or constraint to be achieved, performed, or implemented in order to be achieved itself. These dependency links may be seen as top-down decompositions of the soft-goal, in a similar way as introduced by i* [19, 21] and NFR [6]; yet they are different in the direction of the arrows, due to their reading as "dependencies" given above. This choice provides some advantages: first, it results to be, for the stakeholders, an acceptable and well understood simplification [11], because the same kind of link is used to describe both actor dependencies and goal and task decompositions; as well, this allows for a more intuitive reading of the diagrams and a more natural flow of the dependencies and decompositions among the different models; finally, the top-down direction of the decomposition process is evidenced. Soft-goals decompositions may be conjunctive (to satisfy the original soft-goal, all the sub-components must be satisfied), indicated by the label "A" on the dependency link, or disjunctive (it is sufficient that only one of the components is satisfied), indicated by the label "O".

5 Special links supporting the analysis process

Soft-goal modeling is usually a long and fatiguing process, during which stakeholders and analyst interact more times, according to the strategy devised by the analyst. In

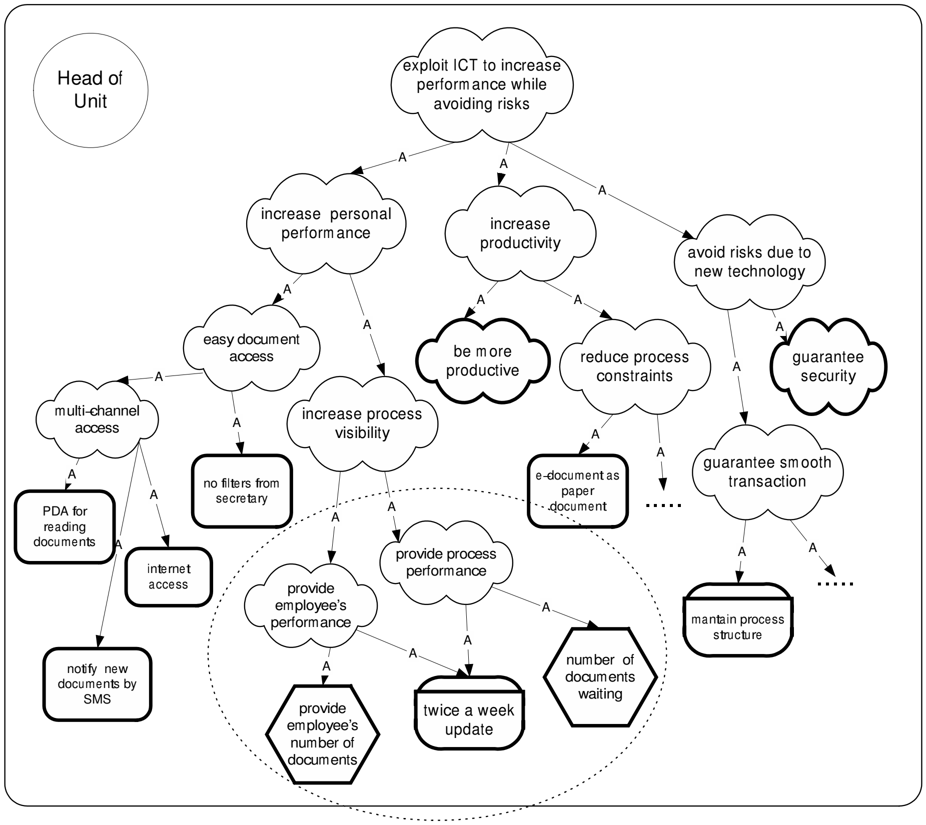

Fig. 3. The exploit ICT to increase performance while avoiding risks Soft-Goal Model

order to better understand this process, let us take a step back, and analyze Figure 4, where the soft-goal model of Figure 3 is shown at an initial stage of the process.

According to Figure 4, the Head of Unit has: to increase personal Performance, to increase productivity of the whole unit, and to avoid risks due to new technology. Let us consider in details only the sub-soft-goal: increase personal performance. It spawns two subordinate soft-goals, easy document access, and increase process visibility, to take better informed decisions. In particular, the soft-goal increase process visibility will eventually lead to the identification of some information about the process that the system will have to provide (i.e., tasks, that is functionalities, that the system will have to implement), as shown in Figure 3.

As mentioned, obtaining such details is a costly process. Before venturing in it, the analyst may prefer, in order to minimize the future analysis effort, to highlight situations where she believes that some commonalities among different goals could be hidden, i.e., situations in which possible shared goals (or tasks, resources, or constraints) could be found during some of the possible alternative developments of the analysis.

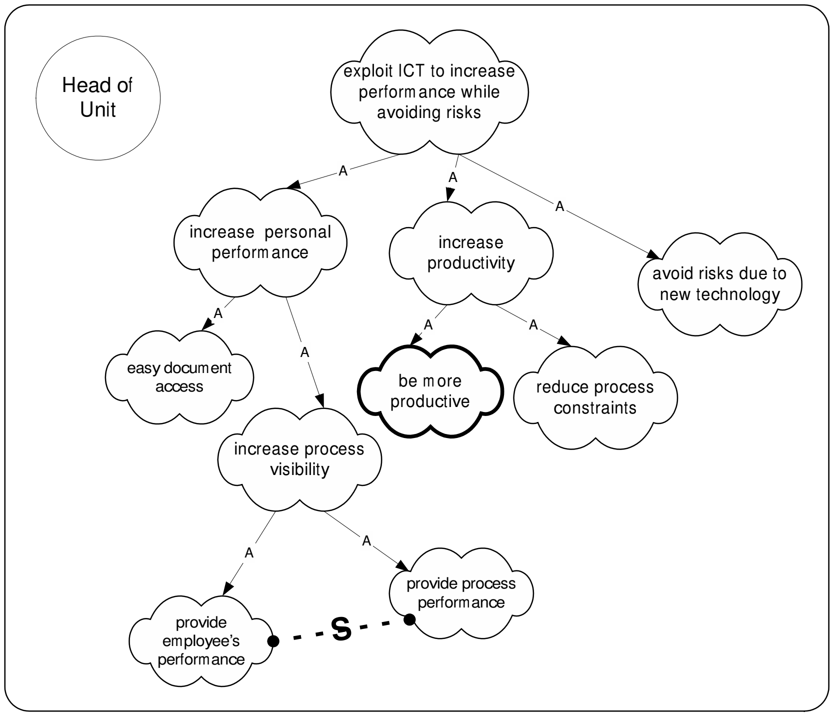

Fig. 4. The exploit ICT to increase performance while avoiding risks Soft-Goal Model under construction

Such a possibility has to be granted as soon as possible. In fact, once the sub-trees have been expanded, not only possible common aspects could easily be cluttered by the details, but, also, the two different sub-trees developments may have led to so diverging solutions that reconciliation or coordination or sharing parts become almost impossible.

In such a perspective, REF provides to the analyst a specific notation to connect two or more goals between which a possible commonality is presumed: it is a dotted line, labeled with "S" (for "Sharing") and referred to as a S-connection . It is relevant noticing that the S-connections are used only by the analysts and have to be removed once the diagram is ready to be presented to the stakeholders.

In Figure 4, the S-connection is used to connect the two soft-goals provide employee's performance and provide process performance. The S-connection may act as a high-level reasoning support tool to enable the analysts to record their intuitions while building the goals models, i.e., making notes to drive their strategies. For example, to highlight where a top-down breath-first diagram expansion may be preferable to a top-down depth-first strategy. As well, the S-Connection reminds the analyst to pay special attention in developing the two sub-trees, and to continuously verify, during the process, the presence of possible common items in the two sub-trees, by organizing more interviews with the stakeholders, and possibly joint elicitation sessions. The final soft-

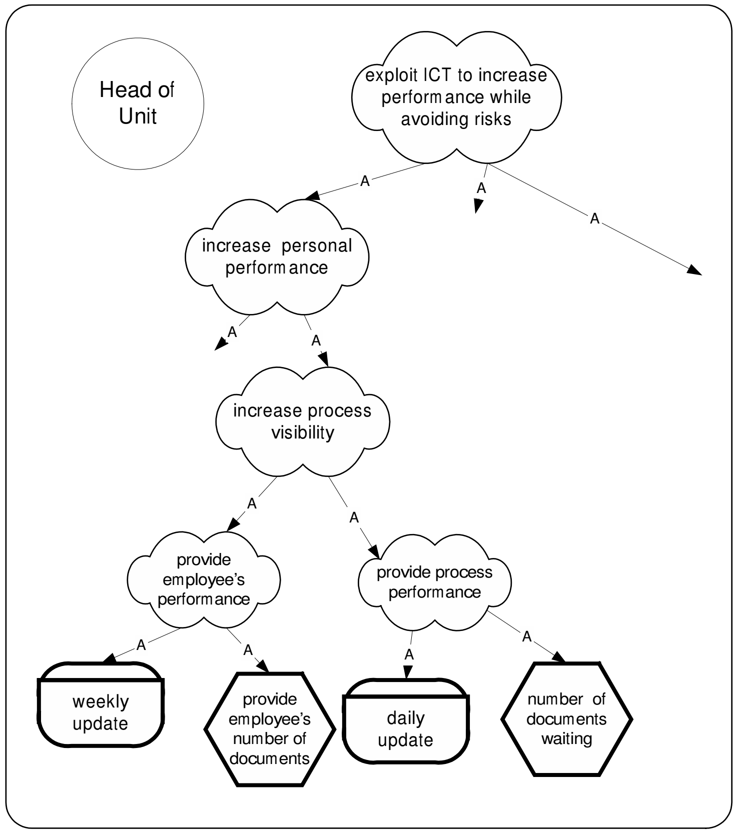

Fig. 5. The exploit ICT to increase performance while avoiding risks alternative (without sharing) Soft-Goal Model. Only the modified diagram fragment is depicted

goal diagram in Figure 3, where the constraint twice a week update has been identified as a common item for the two soft-goals, confirms the validity of the foreseen commonality. Clearly, this is a trivial example: the S-Connection can be better appreciated when deeper sub-trees or different actors are involved. Nevertheless, even in this simple case, the value of the S-Connection can be appreciated, by considering what would have happened without it. The two soft-goals provide employee's performance and provide process performance could have been taken into account in two very different moments and by means of independent analyses and interviews with the stakeholders, leading to two different frequency of update: daily for the process data and weekly for the employee's ones, as shown in Figure 5.

Of course, the diagram in Figure 5 is the one that more finely represent the requirements spontaneously expressed by the stakeholders (it is assumed that weekly update and daily update corresponds to the best options —from the local points of view— in the two cases, respectively). Nevertheless, the solution discovered with Figure 3 (twice a

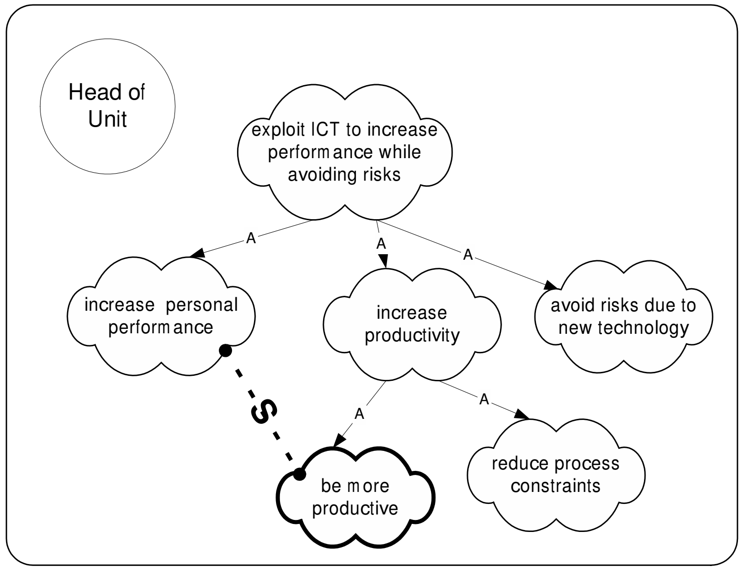

Fig. 6. A possible sharing between goals of different agents

week update) may be considered a reasonable compromise. Moreover, this acceptable (from the local points of view) trade-off corresponds to a much higher global advantage: any possibly necessary further analysis will be carried on in fact only once, rather than twice. In our example the final constraints (leafs with thick borders) correspond to well (quantitatively) characterized requirements for the system-to-be, in other terms, to functional requirements. This means that the solution of Figure 5 will lead to a faster, cheaper, and lighter systems architecture, design, and implementation.[4]

Another, and less trivial, use of the S-Connection is represented in the diagram fragment of Figure 6.

Here, the possible sharing is between the soft-goal increase personal Performance, that the Head of Unit wants to achieve, and the soft-goal be more productive, that the Head of Unit imposes, transfers, to the Employee. In this case, one part of the analysis continues inside the Head of Unit diagram, but the other part is dealt with in the employee's soft-goals analysis diagram. Thus, a possible sharing may happen among different diagrams, as in Figure 7.

Again, as in the previous case, the analysis of such a shared soft-goal immediately assume a higher relevance and priority over the analysis of other goals. Its satisfaction is desired by two actors! The analysis can be carried out only once for both the actors, exploiting the approach to better combine their needs in a synergic way, and avoiding

[4] It is relevant here to note that small advantages at requirements time are then considerably amplified during later phases of Software Engineering: saving a single constraint in REF may finally correspond to saving several lines of implemented code.

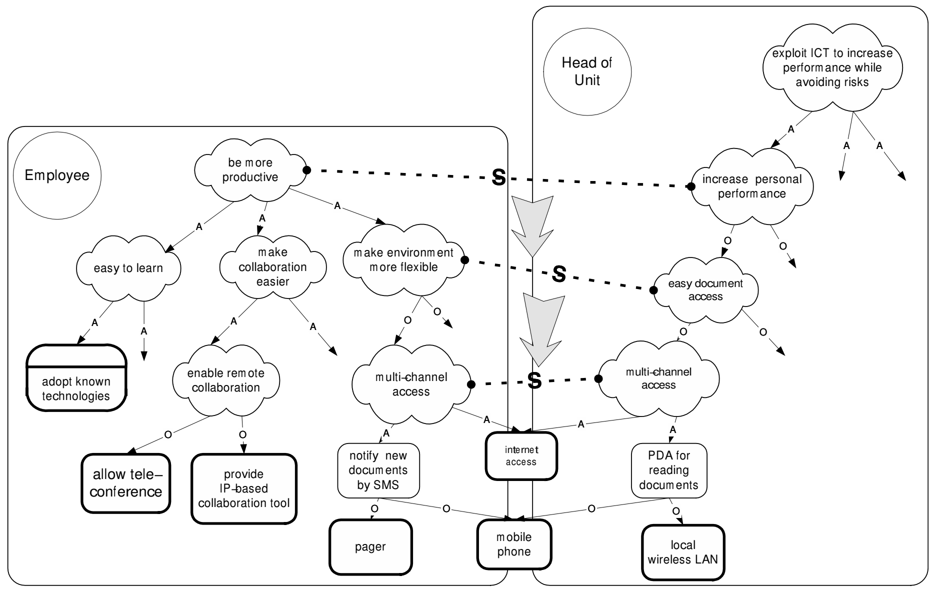

Fig. 7. S-Connection among two different actors

duplications that, although most of the times are due only to slightly different ideas, may generate over complexity in terms of system's functionalities.

Figure 7 summarizes several steps of the analysis process. First, the S-connections is placed at a very top level. The S-connection is then moved by the analyst during the goal refinement process, and better positioned to indicate more specific areas where commonalities could be hidden. At each refinement step, in both the diagrams, several options (evidenced by "O" dependencies) are possible, but the priority is given to those that allow to move the S-connection down, favoring the search of commonalities, accordingly with the opinions of both the stakeholders.

In Figure 7, such movements are evidenced with large gray arrows. Initially, the S-connection links the soft-goal increase personal performance and the soft-goal be more productive. After an initial and combined refinement of the two soft-goals, the analyst moves the S-connection down, between the soft-goals make environment more flexible and easy document access. Finally, after further refinements, the analyst finds two very similar —even with the same label— soft-goals (the multi-channel access), between which the S-connection is placed. Although very similar, these two soft-goals may in reality hide different ideas about what multi-channel access means, i.e., the Head of Unit and the Employee can have different concepts of this capability. For example, while they may agree on a classical Internet based access, the Head of Unit may also want to use a wireless communication based PDA to remotely access the ERMS, while for the Employee may be enough to be informed by a SMS when new documents have been assigned. Thus, different options of refinement of the two commonly labelled soft-goals are possible, but a preference is given to those that eventually operationalize both

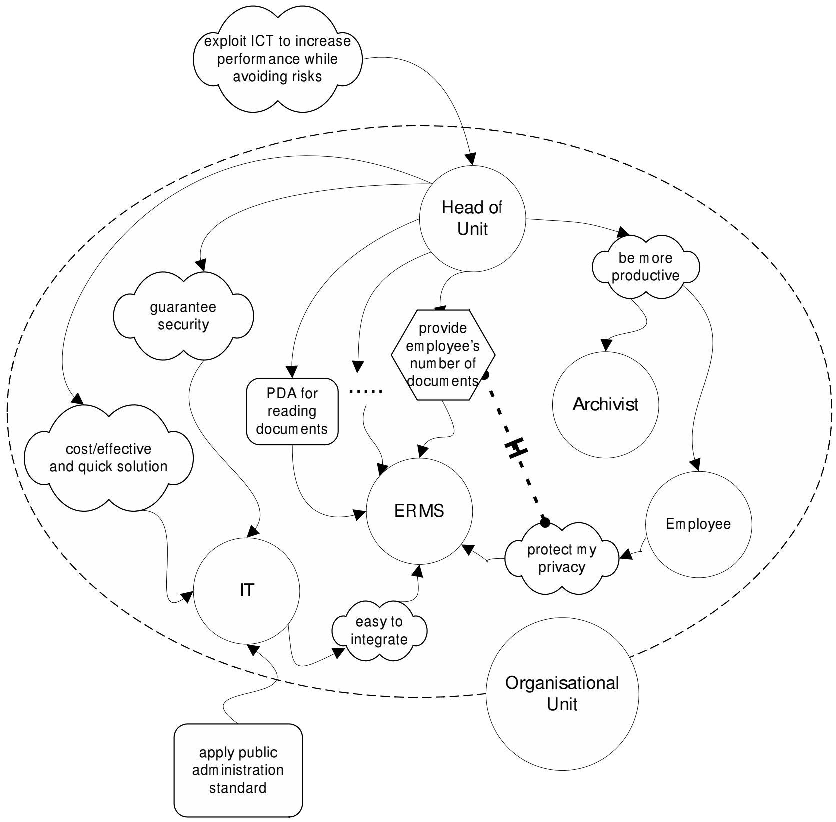

Fig. 8. The evolving organization model

the soft-goals by means of common goals, shared by both the actors. This solution is shown in Figure 7, where the common communication channel (mobile phone) has been accepted by both the actors, as viable to enable both SMS messaging and PDA remote access. Thus, the leaf nodes drawn in thick lines (meaning that they have not to be further analyzed inside the considered context, but transfered to another actor, namely, the ERMS systems) correspond to the final (possibly common) system requirements.

As a final step, let's consider how the initial organization model of Figure 2 can now be enriched, leading to the model in Figure 8 (where some details have been omitted for the sake of clarity). Some new actors have been introduced: the Archivist and the Employee, which have to be more productive, the Information Technology, which has to guarantee security and 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 actor Information Technology, which, on its turn, will have to achieve other goals coming from the external environment, such as, for example, apply public administration standards.

From Figure 8, we can also see how the Employee, as any other actor, may generate its own goals. In particular, it requires the ERMS to protect her privacy (soft-goal protect my privacy).

This goal provides the occasion to show how REF can provide support in detecting and resolving clashing needs (goals, or constraints, tasks, and resources).

Indeed, through goal modeling, REF provides the tool for the detailed recognition of such situations. In fact, when fully operationalized in terms of tasks and constraints, goals models can be adopted to detect clashing situations and to resolve them. Nevertheless, it is always better to be able to recognize such situations as early as possible, even if only at a very qualitative level, before pushing the analysis down to the final constraints and details. For such a reason, to enable the analysts to mark possible conflicting situations (and build their refinement strategy to deal with them), the H-connection link ("H" for "Hurting") is available in REF. This is a powerful tool to detect possible conflicts and try to reconcile different stakeholders points of view, allowing to evolve the analyses only along the most promising alternatives. An example application is shown in Figure 8, where the H-connection has been used to highlight the possibility of a conflict between the Employee soft-goal protect my privacy and the task provides employee's number of documents, required by the Head of Unit. On the basis of such an 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 the Employee's attention to evaluate whether or not providing the number of documents she is dealing with represents a breach of her privacy.

If such is the case, the Employee and the analysts may agree for a different solution, for example that the ERMS will provide only the average number of documents, and then propose such a solution to the Head of Unit.

Concluding, the H-connection, similarly to the S-connection, provides the analyst with a marking mechanism very useful to better drive and control the analysis process. As it happens with S-connections, also the H-connections are dropped once the final diagrams are ready to be submitted to the stakeholders In fact, S-connections, as well as H-connections, are used only to support the analyst during the development phases. Once a stable model has been produced, there is no more the need to maintain this kind of annotations in the diagrams, specially if different unexploited S-connections and H-connections are present, since one of the main aims of REF is to keep the diagrams as simpler as possible for the stakeholders.

6 Conclusions

The paper introduced REF, a requirements engineering framework explicitly designed to support the analysts in reasoning about socio-technical systems, and transform high-level organizational needs into system requirements. By adopting concepts as those of *Actors*, *Goals*, and *Intentional Dependency*, and introducing an essential graphical notation, REF results to be a very effective and easy to deal with (usable) tool, able to tackle complex real situations, while remaining simple enough to allow a concrete and effective stakeholders involvement. In addition, REF supports the analysts in dealing with complex and system/organizational design related issues, such as shared and clashing stakeholders' needs, by introducing some specific analysis-oriented notations to allow an early marking and detection of such situations.

The described application example demonstrates the feasibility of the suggested approach, and the benefits it offers during the early phases of requirements engineering, when the analyst and the stakeholders have to cooperate to understand and reason about the organizational context within which the new system has to function, in order to identify and formalize not only the system requirements, but also the organizational setting that better exploits the new system's capabilities.

Finally, although REF addresses only the early stages of the requirements engineering process, the possibility of combining its outcome with techniques more suitable for dealing with further system development phases has been investigated. For example, practical results suggest that it can be usefully applied as a forerunner to object-oriented approaches, such as those based upon UML [8], as well as to Agent-Oriented Software Engineering approaches for the development of Multi-Agent-Systems [4, 14, 12].

References

1. A. I. Antón. Goal-based requirements analysis. In *Proceedings of the IEEE International Conference on Requirements Engineering (ICRE '96)*, Colorado Springs, USA, Apr. 1996.
2. A. I. Antón and C. Potts. Requirements for evolving systems. In *Proceedings of the International Conference on Software Engineering (ICSE '98)*, Kyoto, Japan, Apr. 1998.
3. V. Basili, G. Caldiera, and H. Rombach. *The Goal Question Metric Approach*. Wiley&Sons Inc, 1994.
4. P. Bresciani, A. Perini, , F. Giunchiglia, P. Giorgini, and J. Mylopoulos. A Knowledge Level Software Engineering Methodology for Agent Oriented Programming. In *Proceedings of the Fifth International Conference on Autonomous Agents*, Montreal, Canada, May 2001.
5. J. Castro, M. Kolp, and J. Mylopoulos. Developing agent-oriented information systems for the enterprise. In *Proceedings Third International Conference on Enterprise Information Systems*, Stafford UK, July 2000.
6. L. K. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos. *Non-Functional Requirements in Software Engineering*. Kluwer Publishing, 2000.
7. A. Dardenne, A. van Lamsweerde, and S. Fickas. Goal-directed requirements acquisition. *Science of Computer Programming*, 20(1-2):3–50, 1993.
8. P. Donzelli and M. Moulding. Application domain modelling for the verification and validation of synthetic environments: from requirements engineering to conceptual modelling. In *Proceedings of the Spring 2000 Simulation Interoperability Workshop*, Orlando, FL, Mar. 2000.
9. P. Donzelli and M. Moulding. 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*, LNCS, Orlando, FL, 2000. Springer-Verlag.
10. P. Donzelli and R. Setola. Putting the customer at the center of the IT system — a case study. In *Proceedings of the Euro-Web 2001 Conference — The Web in the Public Administration*, Pisa, Italy, Dec. 2001.
11. P. Donzelli and R. Setola. Handling the knowledge acquired during the requirements engineering process. In *Proceedings of the Fourteenth International Conference on Knowledge Engineering and Software Engineering (SEKE)*, 2002.

12. P. Giorgini, A. Perini, J. Mylopoulos, F. Giunchiglia, and P. Bresciani. Agent-oriented software development: A case study. In *Proceedings of the Thirteenth International Conference on Software Engineering - Knowledge Engineering (SEKE01)*, Buenos Aires, June 2001.
13. S. Jacobs and R. Holten. Goal driven business modelling: supporting decision making within information systems development. In *Proceedings of conference on Organizational computing systems*. ACM Press, 1995.
14. J. Mylopoulos and J. Castro. Tropos: A framework for requirements-driven software development. In J. Brinkkemper and A. Solvberg, editors, *Information System Engineering: State of the Art and Research Themes*, Lecture Notes in Computer Science. Springer-Verlag, 2000.
15. J. Mylopoulos, L. Chung, S. Liao, H. Wang, and E. Yu. Exploring alternatives during requirements analysis. *IEEE Software*, 18(1):92–96, 2001.
16. A. Perini, P. Bresciani, P. Giorgini, F. Giunchiglia, and J. Mylopoulos. Towards an Agent Oriented approach to Software Engineering. In *Proceedings of WOA 2001 – Dagli oggetti agli agenti: tendenze evolutive dei sistemi software*, Modena, Sept. 2001. Pitagora Editrice Bologna.
17. C. Rolland and N. Prakash. From conceptual modelling to requirements engineering. *Annals of Software Engineering*, 10:151–176, 2000.
18. A. van Lamsweerde. Requirements Engineering in the Year 00: A Research Perspective. In *Proceedings of the 22nd International Conference on Software Engineering, Invited Paper*, ACM Press, June 2000.
19. E. Yu. *Modeling Strategic Relationships for Process Reengineering*. PhD thesis, University of Toronto, Department of Computer Science, University of Toronto, 1995.
20. E. Yu. Why agent-oriented requirements engineering. In *Proceedings of 3rd Workshop on Requirements Engineering For Software Quality*, Barcelona, Catalonia, June 1997.
21. E. Yu and J. Mylopoulos. Using goals, rules, and methods to support reasoning in business process reengineering. *International Journal of Intelligent Systems in Accounting, Finance and Management*, 1(5):1–13, Jan. 1996.