Requirements Engineering for Knowledge Management in eGovernment

Paolo Bresciani[1], Paolo Donzelli[2], Angela Forte[2]

[1]ITC-irst, Istituto per la Ricerca Scientifica e Tecnologica,
Via Sommarive 14, 38050 Trento-Povo (Italy)
brescian@itc.it

[2]Presidenza del Consiglio dei Ministri, Dipartimento per l'innovazione e le tecnologie
Via Barberini 38, 00187 Roma
{p.donzelli, an.forte}@governo.it

Abstract. EGovernment aims at exploiting *Information and Communication Technologies* (ICT) to provide better quality services to citizens and businesses, mainly through electronic delivery channels. Different strategies have been suggested to implement eGovernment; all recognize as fundamental, to deal with, and exploit, continuous ICT evolution, to transform the public administration into a learning organization, characterized by a high sharing, reuse, and strategic application of the acquired knowledge and lessons learned. An interesting role can be played by techniques, methods and approaches recently suggested by requirements engineering (RE), being most of the reusable knowledge located at requirements level: from ICT components requirements, to business and organizational models.
The paper presents an advanced agent- and goal-based RE framework, designed to support capturing and formalizing the knowledge embedded in the organization. An on-going project concerned with the introduction into a complex administrative organization of a Electronic Record Management System is described.

1. Introduction

EGovernment [1,2] aims at exploiting *Information and Communication Technologies* (ICT) to provide better quality services to the government customers (citizens and businesses), mainly through electronic delivery channels (internet, digital TV, easy web, mobile phone, etc.).

Although there are differences among strategies adopted by different governments, it is possible to identify a common roadmap towards government implementation, characterized by four main milestones: (1) Establish a government–wide communication infrastructure, to enable cooperation among the different public sector components, both at central and local level, and as a necessary step to (2) create a virtual corporate IT infrastructure, upon which (3) activate channels for service delivery. Fundamental for the success of the first three steps, and recognized as fundamental to efficiently manage eGovernment evolution (e.g. to deal with and exploit continuous ICT evolution), it is to (4) transform the public administration into a

learning organization, characterized by a high sharing, reuse, and strategic application of the acquired knowledge and lessons learned.

A learning organization [3] is an organization skilled at "modifying its behavior to reflect new knowledge and insights". The basic idea behind it consists in creating a *knowledge chain* (for knowledge collection, production, customization and delivery), suitable to support and improve the whole organization functioning. The critical point for applying such a concept to eGovernment is to find the materials suitable to feed the knowledge chain: a) to identify the fragments of knowledge that could be efficiently reused but, above all, accepted; b) to represent and formalize such fragments so to be tractable (stored, analyzed, understood, customized, and eventually transferred).

A modern public administration can be considered as an ecosystem where different entities, from central government units to local authorities, and public sector agencies, interact, cooperate and sometimes clash to achieve both general and private goals. While, as a whole, such entities act as the owners of the eGovernment process, each entity is granted a considerable level of political and economical independence, being entitled, for example, to make its own choices in technological, organizational and strategic terms. Transform such a structure into a learning organization is a difficult task: the kind of knowledge suitable to flow among the different entities has to be carefully identified and planned in order to be not only practically and economically feasible, but also acceptable by the various actors, so becoming a support to the evolution of the whole system.

In such a context, an interesting role can be played by techniques, methods and approaches recently suggested by *Requirements Engineering* (RE) [4,5,6,7].

RE is concerned with managing desired properties and constraints of software-intensive systems and with goals to be achieved in the environment. Traditionally, the focus has been placed upon the final system and its, more or less, direct users. Recently, the scope of the attention has been enlarged until encompassing the application context. The software system and its application context form in fact a larger social-technical system that has to be treated and analyzed as a whole: the overall needs of such social-technical system are the ones that RE has to fulfill. Consequently, appropriate organization modeling techniques are typically advocated to capture high-level organizational needs and to transform them into system requirements, while redesigning the organizational structure that better exploit the new system. While doing so, RE can also provide the means to capture, formalize and package the resulting knowledge, turning it into materials suitable to feed the knowledge chain, at the basis of the public administration learning organization. Most of the reusable knowledge, in fact, can only be located at requirements level, where the term “requirements” does not refer only to requirements for ICT components (e.g. from the classical system requirements, to user manuals, to procurements guidelines etc.), but also to organization structure and business models (to better employ the new technology), to technology transfer approaches, to human resources management (e.g. training and updating programmes), to final users needs.

The remainder of this paper is organized as follows. Section 2 introduces the concept of public administration as an ecosystem and discuss specific needs knowledge management and transfer may have for such systems. Section 3 discusses the role that RE, and specifically the recently proposed RE techniques based on concepts such as those of Agent, Goals and Dependency [4,5,6,7], may have in this context. Section 4

introduces a specific goal and agent-based requirements engineering framework (REF), by briefly describing its main characteristics. Section 5 elaborates some of the aspects introduced in Section 3, by showing some extracts from an on-going project aiming at introducing an Electronic Record Management System within the main administrative and decision-making processes of a main government unit: a first step towards a paperless knowledge workplace. Finally, Section 6 concludes by discussing some of the observed benefits, mainly in terms of knowledge management.

2. The Public Administration as an Ecosystem

Recently, different questions have been raised regarding the strict correlation between politics, public administration and ICT [1], among these, (1) the possibility of modernizing the public administration through the ICT; (2) the identification of strategies suitable to control the ICT application; (3) the impact of ICT on the political institutions; (4) the relationships between the innovation potential offered by the ICT and the changes that are transforming the public sector not directly related to the ICT, as, for example, the transfer of responsibilities towards local authorities. In an attempt to establish a government characterized by a strong interaction between political and social components, such a devolution process aims at establishing an auto-regulative and co-evolutive relationship among all the involved entities.

Borrowing from biology, we can assimilate the public administration, the citizens, the businesses, and, in general, all the social components to an ecosystem: an entity that encompasses all the subjects acting in the same area and interacting with the external environment, in such a way that the flow of energy (i.e., knowledge in our context) leads to a well-defined structure. The term ecosystem allows to stress the strong interdependency among components, but also to highlight the existence of an "universal law" that tends to optimize entities' freedom and wealth distribution. In particular, for the public domain, the presence of a decisional center enables to identify the "right", also if for a limited extension of time, for the whole society [8].

These are complex organizational issues, and the current shape of the public sector structure plays a controversial role, not facilitating such a transfer of powers. Various socio-political and administrative thesis recognize the ICT as an important accelerating factor in the process of government decentralization and see in the eGovernment a powerful implementation tool to support this new concept of public administration [1,2]. Although, at an initial stage, the application of ICT does not change the traditional government ways of operating, the eGovernment sets the basis for more advanced integration and exchange models: from a more active involvement of the citizen, not only user of services but generator of proposals, to the electronic democracy, where the citizen becomes a critical element of the political decision making process. The current debate about the public sectors options and the available technologies is the symptom of an innovation process that is clearly generating uncomfortable feelings, by which, however, the research towards models more suitable to satisfy the new socio-organizational needs through the eGovernment could be activated. As will be discussed in the next Section, knowledge representation and reasoning support tools recently suggested within the context of RE could be beneficial in dealing with such complex issues.

3. The Role of RE to support eGovernment

RE deals with the definition, formalization, and analysis of the requirements that a potential system must have. Recent approaches [4,5,6,7] also suggest that RE must face, as well, the "problem of introducing the system"—an already existing or yet to be built system—into the organization. New developments in ICT have made new systems highly pervasive and strictly intertwined with their application context, so that, introducing a new system may have different and strong impacts (positive, but also negative—thus, to be analyzed as potential sources of problems) on the organization itself; these impacts may have a level of importance similar to that of the introduction of new human actors or positions and related responsibilities. Thus, defining and analyzing the requirements of a new system cannot anymore be considered as a stand-alone activity, rather has to be strongly interrelated with the deep comprehension of the target organization and of its evolutionary process, posing on the requirements engineers a completely new set of issues.

The impact of the new system on the organization is much more relevant when the system is tighter incorporated in the workflow of data and knowledge interchange that characterize the complex lattice of relationships inside the organization, and has to be dealt with by means of an approach in which the system and the other (social) components (i.e., individuals and teams) of the organizations can be tackled in the same way. EGovernment, in particular, specifically requires the adoption of such a perspective. Public administrations, and their organizational environment, are characterized by the presence of very diverse kinds of actors (e.g., citizens and businesses, employees and administrators, politicians and decision-makers —both at central and local level), each of them with its own objectives and goals. Thus, in general, eGovernment 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 system has to be introduced.

This level of knowledge, situated between the high-level needs and goals an organization wants to achieve and the technological solutions, not only results fundamental in identifying the right system, but represents also the kind of information more suitable to be spread within the public administration, likely to be accepted and eventually reused by the different components of such an ecosystem. Being able to cope with such a level of knowledge, i.e. to capture, formalize and make it easily available, is therefore crucial to support and accelerate the eGovernment process.

Among different RE methodologies, only few center their attention on notions like those of Agent (or Actor), Goal, and Intentional Dependency [4,5,6,7] that can support the analysts in dealing with, and reasoning about, the kind of knowledge described above [9,10,11], making it available to foster the eGovernment process. In such a perspective, we present our RE methodology, called REF [7,9,12,13], that, by

exploiting the characteristics just mentioned, we believe is particularly suited to be applied to eGovernment applications in general, while creating reusable fragments of knowledge.

4. REF

REF [7,9,12] is designed to deal with, and reason about, socio-technical systems. The basic idea is that REF has to provide the analyst with a powerful tool to capture high-level organizational needs and to transform them into system requirements, while redesigning the organizational structure that better exploit the new system. Moreover, while doing so, REF can also provide the means to capture, formalize and package the resulting knowledge, turning it into materials suitable to feed the knowledge chain that could be at the basis of the public administration learning organization.

The framework tackles the modeling effort by breaking the activity down into more intellectually manageable components, and by adopting a combination of different approaches, on the basis of a common conceptual notation.

Agents (elsewhere the term Actor is used) are used to model the organization [4,6,7,14]. The organizational context is modeled as a network of interacting agents (any kind of active entity, e.g. teams, humans and machines, one of which is the target system), collaborating or conflicting in order to achieve both individual and organizational goals. Goals [4,6,7,15] are used to model agents' relationships, and, eventually, to link organizational needs to system requirements. According to the nature of a goal, a distinction is made between hard goals and soft goals. A goal is classified as hard when its achievement criterion is sharply defined. For example the goal *document be available* is a hard goal, being easy to check whether or not it has been achieved (i.e., is the document available, or not?). For a soft goal, instead, it is up to the goal originator, or to an agreement between the involved agents, to decide when the goal is considered to have been achieved. For example, the goal *document easily and promptly available* is a soft goal, given that as soon as we introduce concepts such as "easy" and "prompt", different persons usually have different opinions. Another characteristics of soft goals is that they can often be seen as a kind of modifiers or quality attributes associated to a hard goal; thus, in the previous example, the soft notion of easily and promptly modifies the precise objectives of having the document available. Distinguishing goal modeling from organizational modeling, and then, further distinguishing between hard goal modeling and soft goal modeling, is a key aspect of REF, and helps to reduce the complexity of the modeling effort. The proposed framework, therefore, supports three inter-related modeling efforts: the organizational modeling, the hard goal modeling and the soft goal modeling.

During Organization Modeling, the organizational context is analyzed and the agents and their hard and soft goals identified. Any agent may generate its own goals, may operate to achieve goals on the behalf of some other agents, may decide to collaborate with or delegate to other agents for a specific goal, and might clash on some other ones. The resulting goals will then be refined, through interaction with the involved stakeholders, by hard and soft goal modeling. The Hard Goal Modeling seeks to determine how the agent can achieve a hard goal placed upon it, by decomposing them into more elementary subordinate hard goals and tasks (where a task is a well-

specified prescriptive activity). 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 agents [16,17,18] 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. Soft goals refinement has to be reiterated until only hard goals, tasks and constraints are obtained (that is, until all the "soft" aspects are dealt with). Constraints are associated with hard goals and tasks to specify the corresponding quality attributes. So, for example, the soft goal to make a document easily and promptly available, beside spawning the hard goal to 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. In other words, for each soft goal, the resulting set of constraints represents the final and operationalised views of the involved quality attributes, i.e. the quality measurement models that formalize the attributes for the specific context [16,17].

During Soft Goal Modeling the analysts and, above all, the stakeholders tend to (and somehow are forced to) clarify very early in the project concepts that are usually left blurred until implementation imposes to make some choice. Thus, soft goals become a knowledge representation vehicle that: 1) encourages the interaction between the analysts and the stakeholders, and among the stakeholders themselves; 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, allow the analysts and the stakeholders to early detect clashing requirements, which usually are hidden behind generic and left-implicit assumptions, providing, at the same time, an operational and cooperative way to resolve them, by reconciling the different stakeholders' points of view.

In a nutshell, we can say that REF provides a significant series of tools to identify, define, verify and transfer relevant pieces of context knowledge, taking into account the needs and points of views of the various involved actors, both as knowledge providers and knowledge users.

5. The case Study

In order to illustrate REF potentials, we adopt as a case study some extracts from an on-going project aiming at introducing an Electronic Record Management System (ERMS) within the Italian Cabinet Office. 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.

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. Moreover, it usually 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 XML [19]. 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.). It is, in fact, a necessary step for introducing more sophisticated document management tools, such as workflow technology and digital signature, both fundamental mechanisms for a paperless and ubiquitous working environment. 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. The initial organization model expressing such a situation is shown in Figure 1. Circles represent agents, and dotted lines are used to bound the internal structure of complex agents; that is, agents containing other agents. In Figure 1, the complex agent *Organization Unit* corresponds to the organizational fragment into which it is planned to introduce the new ERMS, whereas the *Head of Unit* is the agent, acting within the *Organization Unit*, responsible for achieving the required organizational improvement (modeled by the soft goals *exploit ICT to increase performance while avoiding risks*, and *cost/effective and quick solution*).

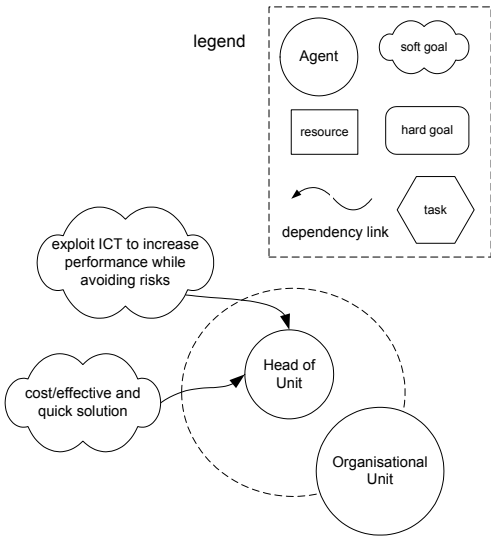

Fig. 1. Introducing the ERMS: the initial model

Goals, tasks 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 (i.e. two or more) agents.

As mentioned, the soft goals modeling process allow the analysts and the stakeholders to operationalise all the soft aspects implicitly included in the meaning of the soft goal. Thus, for example, Figure 2 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 2, in other terms, represents 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. Again, the arrowhead lines indicate dependency links. A soft goal depends on a sub-ordinate soft goal, hard goal, task or constraint, when it requires that goal, task or constraint to be achieved, performed, or implemented in order to be achieved itself. These dependency links may be seen as a kind of top-down decomposition of the soft goal. Soft goals decompositions may be conjunctive (all the sub-components must be satisfied, to satisfy the original soft goal), 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" on the dependency link (see Figure 4). According to Figure 2, the *Head of Unit* has to increase personal performance, to increase productivity of the whole unit, and also to avoid risks due to new technology.

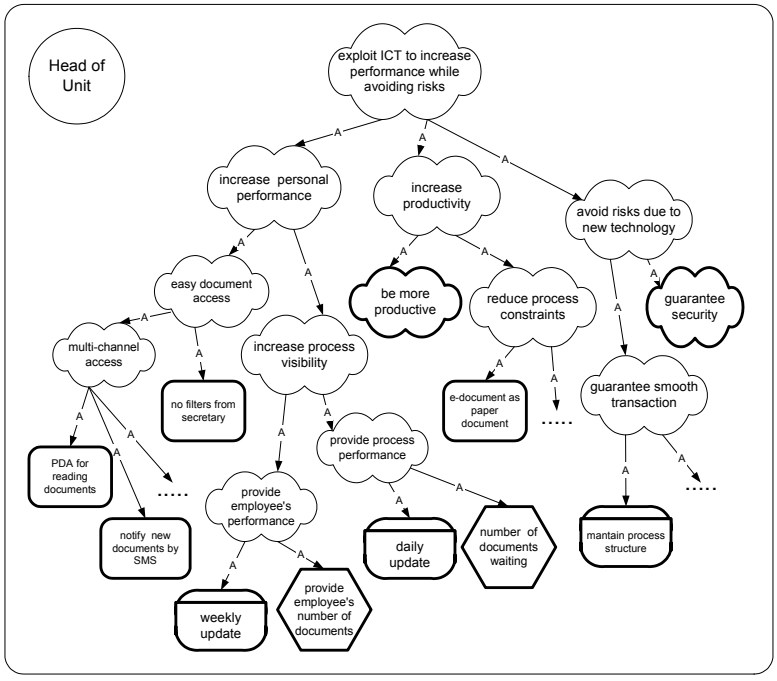

Fig. 2. The "*exploit ICT* ….." Soft Goal Model

In Figure 2, the items in bold outline are those that the *Head of Unit* will pass out, having decided to depend on other agents for their achievement. For such a 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. The results of

this analysis allow us to enrich the initial organization model in Figure 1, leading to the model in Figure 3, where some details have been omitted for the sake of clarity. In Figure 3 some new agents 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 3, 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, *apply public administration standards*. At this point, the analysis can be carried on by focusing, for example, on how the *Employee* will try to achieve the soft goal *be more productive*. On the other side, to be more productive, the *Employee* will define its own strategy, eventually reaching an agreement with the *Head of Unit*. Such a strategy is shown by the soft goal model in Figure 4, where we can see how in order to be more productive the *Employee* will ask that the system will be *easy to learn* and will *make collaboration easier* with the other employees, which are dealing with the same document.

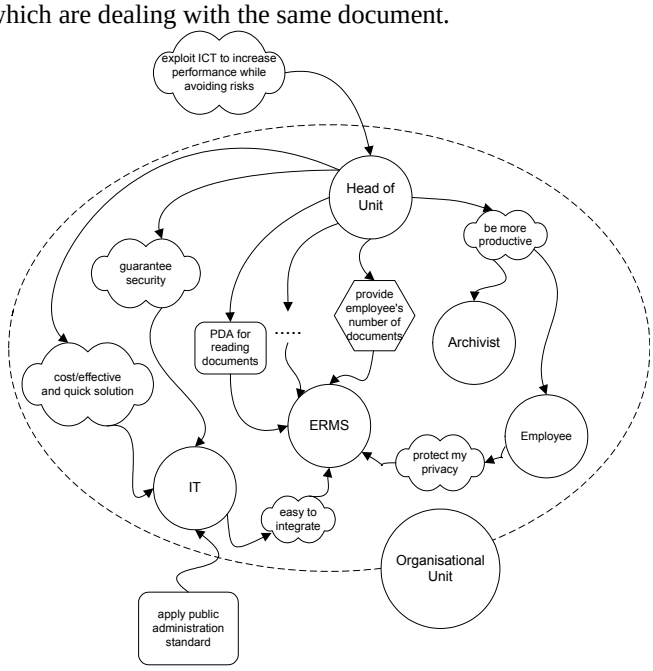

Fig. 3. The evolving organization model

For sake of brevity, only a partial refinement of these soft goals is shown. The soft goal *easy to learn* will spawn, among other items here omitted, the constraint *adopt known technologies* (i.e. technologies which the employee is already used to), whereas the soft goal *make collaboration easier* will lead, through further refinement, to a series of hard goals implying specific capabilities (e.g. either a teleconference or an IP-based collaboration tool) and access channels (e.g. mobile phone, laptop, etc.).

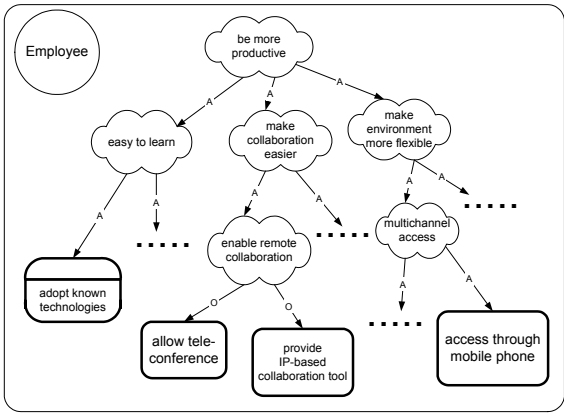

Fig. 4. The "*be more productive*" Soft Goal Model

6. Discussion and conclusions

The described application example, as other similar ones [9], demonstrates the feasibility of the suggested approach, and the benefits it offers during the early phases of requirements engineering process, when the analysts 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.

A large amount of information is discovered in these early phases and have to be properly organized in order to be fully beneficial to development activities. In such a perspective, each type of model produced within the framework (soft goal, hard goal and organizational model) plays a particular role as a knowledge handling tool, by providing a specific knowledge representation vehicle that the analyst can use to interact with the stakeholders. For example, hard goal models allow the stakeholders to make explicit the tasks they will have to perform and the resources they will need. In addition to this, soft goal models bring quality issues into the picture and support the stakeholders while reasoning about their own concepts of quality, highlight possible conflicts, and support negotiation towards a feasible solution, in terms of tasks and constraints. So, while soft goal and hard goal models as a whole lead to system requirements, soft goal models, as by product, allow the analysts to freeze the knowledge acquired and produce models of the quality attributes of interest (for example, the model of employee's performance). The organization model, finally, represents the organization requirements, providing the management with a clear view of how the business process will be changed, or affected, by the introduction of the new system, and allowing the stakeholders to understand and validate their role within the organization, by making explicit the interactions with the system or with other agents in which they will be involved.

Once the project is over and the system deployed, the knowledge acquired during the requirement engineering process does not extinguish its role. On the contrary, made easily accessible and reusable through the different models into which has been

captured, such a knowledge offers potential benefits also in the post-deployment phases, supporting system maintenance and evolution and use.

The clear links established (through the different models) between organizational goals and system requirements, in fact, allow the analysts and ICT managers to quickly identify the effects that changes in the organizational goals or new technology trends may have upon the system requirements. The possibilities offered to our context by a new access technology, for example, could be easily evaluated by observing the corresponding goal models (i.e. Figures 4 and 5): the new technology will be judged valuable for the system (and the organization) when capable to overcome some of the limits found during the previous analysis, or, in other terms, to enable the stakeholder to better achieve his goals. At the same way, a change in an organizational goal may easily be translated into requirements changes.

The framework models improve and support also knowledge transfer and sharing across different projects. In particular, fragments of knowledge, pieces or information, or system components may be reused in different application contexts. For example, the outcome of the framework, i.e. the requirements expressed as refinements of high-level goals, can be easily generalized and exported to different government units, contributing to improve the sharing and reuse of the knowledge captured, produced, and formalized during the very early stages of an ICT system development. In other terms, they represent the fragments of knowledge that may used to create the knowledge chain necessary to transform the public administration into a learning organization.

The last advantage we foresee that could be exploited by adopting REF is during the very use of a knowledge management (KM) system in the organization. Of course, the KM system so developed is in principle compliant to the knowledge workers needs; nevertheless, still it may be not easy for the users to totally understand the structure of the system, and the role it plays, as an agent—among others in the organization—with the capability of collecting and being the “communication channel” for the knowledge flow. As well, it may not be easy to appraise the value of the retrieved knowledge, if no context on the sources is provided. It is clear that both the roles of the KM system—as a repository and as a communication channel—and the value of the retrieved knowledge tightly depend on the social network, which is the source of the knowledge and in which the system is embedded. Having a clearly understandable representation of such social network, of its dependencies, of the motivations that stimulate the knowledge providers, and of their relationships with the system and among them, may represent a useful support to understand the real meaning and the value of the provided pieces of knowledge (and, thus, also to locate the most relevant ones). All these is already available in the REF diagrams developed during the RE process. Providing them, in appropriate formats, to the final system users, may represent an opportunity to improve the socio-technical system—as a whole—understandability, as well as its usability, providing, at the same time, the occasion for an evolution of the organization performances.

References

1. Frissen P.H.A. “Politics, Governance and Technology”, Edward Elgar, Northampton, MA, USA, 1999.

2. Margetts H. “Information Technology in Government”, Routledge, London, UK, 1999
3. Garvin, D. A., “Building a learning organization”. Harvard Business Review 71(4): 78–91, 1993.
4. Yu, E. “Why Agent-Oriented Requirements Engineering” Proceedings of 3rd Workshop on Requirements Engineering For Software Quality, Barcelona, Catalonia, June 1997.
5. Yu E., Mylopoulos, J. “Using Goals, Rules, and Methods to Support Reasoning in Business Process Reengineering”. International Journal of Intelligent Systems in Accounting, Finance and Management. (5) (1),1996.
6. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J. “A knowledge level software engineering methodology for agent oriented programming”. Proceedings of the 5th International Conference on Autonomous Agents, Montreal, Canada, May 2001.
7. Donzelli, P., Moulding M.R. “Developments in Application Domain Modelling for the Verification and Validation of Synthetic Environments: A Formal Requirements Engineering Framework”, Proceedings of the Spring 99 Simulation Interoperability Workshop, Orlando, FL, March 1999.
8. Sen A, “Development as Freedom”, Mondadori Ed., Milano, Italy, 1999.
9. Donzelli P., Setola R., “Handling the knowledge acquired during the requirements engineering process”, Proceedings of the Fourteenth International Conference on Knowledge Engineering and Software Engineering (SEKE), 2002.
10. Mylopolous, J., Castro, J. “Tropos:A Framework for Requirements-Driven Software Development”, Information System Engineering: State of the Art and Research themes, Brinkkemper, J., Solvberg, A. (eds), Lecture Notes in Computer Science, Springer Verlag, 2000.
11. Giorgini, P., Perini, A., Mylopoulos, J., Giunchiglia, F., Bresciani, P. “Agent-oriented software development: A case study”. Proceedings of the Thirteenth International Conference on Software Engineering and Knowledge Engineering (SEKE), 2001.
12. Donzelli, P., Moulding M.R. “Application Domain Modelling for the Verification and Validation of Synthetic Environments: from Requirements Engineering to Conceptual Modelling”, Proceedings of the Spring 2000 Simulation Interoperability Workshop, Orlando, FL, March 2000.
13. Donzelli, P., Bresciani, P., “Goal Oriented requirements Engineering: a case Study in eGovernment”, proceedings of the 15th Conference on Advanced Information Systems Engineering (CAISE’03), Klagenfurt, Austria, 16-20 June, 2003 (to appear)
14. D'Inverno M., Luck, M. “Development and Application of an Agent Based Framework” Proceedings of the First IEEE International Conference on Formal Engineering Methods, Hiroshima, Japan, 1997.
15. Dardenne, A., Van Lamsweerde, A., Fickas, S. “Goal-Directed Requirements Acquisition” Science of Computer Programming, Vol. 20, North Holland, 1993.
16. Basili, V. R., Caldiera, G., and Rombach, H. D., “The Goal Question Metric Approach”, Encyclopedia of Software Engineering, Wiley&Sons Inc., 1994.
17. Cantone, G., Donzelli P. “Goal-oriented Software Measurement Models”, European Software Control and Metrics Conference, Herstmonceux Castle, East Sussex, UK, April 1999.
18. Chung, L., Nixon, B., Yu, E., Mylopoulos, J. Non-Functional Requirements in Software Engineering, Kluwer Academic Publishers 2000.
19. http://www.w3.org/XML/