Domain Visualization
for Dealing with Complex Information Systems

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

Abstract

Identifying, and successfully applying the right information system requires an organisation to deal with a large amount of heterogeneous issues, at a technical, organizational, and social level. Domain visualization is a key capability to allow the analysts and the stakeholders to navigate through the complexity of the solution space. In such a perspective, this paper introduces a goal-driven requirements engineering framework, and illustrates how it has been applied throughout an eGovernment project.

1. Introduction

Information and communication technologies (ICT) are increasingly becoming more powerful and pervasive. Powerful, as they open huge improvement opportunities to both private and public organizations, by providing new ways of organizing working procedures and business processes, dealing with partners, and reaching out for customers. Pervasive, as they affect a large number of stakeholders, within an organization, and across different and interacting organizations and their customers, by affecting organizational roles and procedures traditionally left untouched by the ICT trend.

Such a combination of power and pervasiveness results into complex information systems, characterized by a high organizational impact. These systems do not simply aim at providing more efficient ways of performing old tasks (e.g., by replacing one or more existing systems, or automating well-established procedures and routines), but especially at introducing new ways of operating that could not exist otherwise.

For theirs own nature (i.e., high impact), however, such systems usually face a high risk of rejection, and are subjected to a slow (and expensive) acceptance process. There may be many specific reasons for this to happen, but, basically, because of an environment not yet ready to fully accept and/or exploit their potential. For example: the organization structure may be unfitted for the new system (e.g., the business process may be outdated if compared with the capabilities of the system); there may be a lack of suitable personnel because of inappropriate hiring and training policies; the new working procedures could be too innovative and/or automated, creating in the personnel feelings of vulnerability, i.e., the fear of becoming superfluous for the organization from which a tacit resistance could result against the new system.

At the same time, while all these factors would demand for a carefully designed and extensively assessed plan of actions, in a dynamically changing business and organizational environment, the time factor plays a strategic role [14]: faster and more dependable system identification, validation, implementation and deployment procedures are strongly advocated to face the competitive business challenges.

The successful identification and acceptance of the new system, its integration with the legacies (both technological and organizational), and the introduction of the necessary organizational changes are therefore all factors that rely on the ability of the analysts and the stakeholders of quickly gaining a firm understanding of the application context. In particular, on their ability of swiftly: (1) capturing organizational needs and transform them into system functions and properties; (2) envisioning the impact of the system upon the application context, and (3) changing (adapting) the application context accordingly, to better exploit the system capabilities and avoid possible negative reactions.

The quest for such capabilities opens a series of relevant questions. For example, how to discuss with a domain expert new ways of organizing the application context around a system that is not yet available? Most of the knowledge needed to assert such scenarios can be classified as tacit: how can the analysts, in cooperation with the stakeholders, first make it explicit, and then usefully apply it to shape a “to-be” environment? How the direct users can be led to “think” in terms of the new system and therefore identify the relevant system requirements before seeing it? In order to answer all these challenges, the analysts and the stakeholders need to be
able (i.e., empowered with the rights tools) to jointly understand and navigate a complex solution space. A puzzle-like space where, to identify and successfully deploy the right system, they need to combine different and heterogeneous pieces of “knowledge”: personnel training and hiring policies, organizational changes, function and properties of the new system, architectural choices, components selection, and corresponding procurement and/or development options.

In line with this, the paper presents the evolution of a goal-driven requirements engineering framework (REF). Initially developed for supporting the requirements engineering (RE) process for complex simulation systems [9], REF has been then extended and enriched to deal with organizational information systems [4,10].

REF makes of domain visualization (and abstraction) the key tool for efficiently dealing with complex systems [12]. The tradeoffs between expressiveness and simplicity have always been at the basis of REF design and evolution, as discussed in [3]. The availability of a rigorous, yet simple approach is in fact crucial. Adopted formalisms and notations need to be quickly grasped and efficiently applied by organizational roles that do not have (and are not intended to have) technical ICT skills (top managers, administrative roles, etc.). However, these are the roles that need to understand (visualize) the application context’s characteristics, weaknesses and opportunities, in order to be able to quickly assess possible solutions and make the decisions whether to proceed or not, and in which direction.

The paper is organized as follows. Section 2 describes the main characteristics of REF. Section 3 introduces the case study, an eGovernment project. Sections 4, 5 and 6 develop the case study, highlighting REF characteristics that supported the decision-making process leading to the final system. Finally, conclusions are given in Section 7.

2. REF

REF builds upon what achieved and recently suggested in the RE field (i.e., agents [8,15,16] and goals [1,7,15,16]), while exploiting ideas and approaches from the software quality modelling area [2]. The basic idea behind REF is to exploit the advantages provided by the application domain modelling notions of agent, goal and intentional dependency, which have been recognized as suitable to support a smooth transition from high-level organizational needs to system requirements [5], and the advantages provided by quality modelling techniques, which allow to capture the stakeholders’ perception of quality since the beginning of a new project.

In REF, the application context is modelled as a network of interacting agents (both humans and machines), 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 behalf of some other agents, may decide to collaborate with other agents for a specific goal, or clash on some other ones.

The notion of goal allows the analysts and the stakeholders to explicitly model, and reason about, the rationale behind the organizational (actual or desired) behaviour. By visualizing the goals, REF allows the analysts and the stakeholders to constantly base their strategies upon the real organizational needs, and, therefore, navigate the solution space more efficiently.

In REF, two types of goals are adopted, hard and soft goals [5,6,15]. A goal is hard when its achievement criterion is sharply defined (e.g., the goal make the document available is hard, being easy to check whether or not it has been achieved: is the document available, or not?). A goal, instead, 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 document easily and promptly available is a soft goal: as soon as we introduce concepts such as “easy” and “prompt”, 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. These, in fact, are human activity systems where the designation of the objectives (goals) is often itself problematic, consisting of soft “ill-structured” issues [13]. In other terms, REF allows the analysts to early recognize, isolate and formalize soft issues (i.e., all those aspects of the solution space that cannot, or are not easy to be specified in precise sharp terms), and enables them, by strictly collaborating with the involved stakeholders, to reason about their “softness”, and, eventually, reduce it into more manageable and “implementable” functions and properties for both the system and the organization.

The notion of agent allows bringing into the picture the stakeholder. Being modelled as agents in the application domain, the stakeholders can visualize their involvement in the new context, and so be in the position of better understanding what could be happening, how they could be affected, and what could be demanded from 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, in order 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.
As shown in Figure 1, REF adopts a cyclic development flow. Through continuous interaction with the stakeholders, the analysts will deal first with the high level organization structure, then will descend step by step into the details of the application context of the new IT system, until the focus will be placed upon the single agents and their role within the organization (among which the system itself). In particular, once an initial model of the organization is built, detailed enough to identify the scope of the problem, including the initial high-level goals, the main agents involved and their interactions (start-up phase), the REF process evolves in a cyclic way through two main steps:

a) **Goal modelling phase**, during which the soft and hard goals discovered during organization modelling are refined in terms of more elementary goals, tasks and constraints.

b) **Organization modelling phase**, during which the analysts use the information gained during the previous step (see the “mapping to the organization” arrow in Figure 1) to enrich and extend the initial organization 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.

Such a cycle is continued until the desired (and needed) level of detail is reached.

### 3. The case Study

EGovernment [11] aims at exploiting ICT to provide better quality services to the government customers (citizens and businesses), mainly through electronic delivery channels (Internet, digital TV, mobile phone, etc.). A critical step in this direction is the enhancement of the internal capabilities (i.e., back-office) of the public sector organizations [4].

The case study reports on a project [4,10] concerned with introducing an **Electronic Record Management System** (ERMS) into the administrative processes of the Italian Cabinet Office, in order to transform a huge repository of documents (from decrees, to tender-related documentation) into a readily available source of knowledge, suitable to be shared organization-wide. The impact of such a system on the common practices of knowledge workers is quite relevant. ERMS is expected to reach a regime situation of about 2000 users and 2 million documents per 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 standard. It 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.).

REF has been adopted throughout the project decision-making process leading to the final system. In particular, REF models have been developed to:

- understand and analyse the original application context;
- capture the high-level organizational goals and refine them into organizational and system requirements;
- support the definition of the main aspects of the IT system architecture and of the corresponding implementation/acquisition strategy.

4. Understand and analyse the original application context

The ERMS target organisation, as many others in the public sector, is structured into internal main units. The REF organisation model in Figure 2 illustrates the generic structure of such a unit, before the introduction of the ERMS.

The basic elements upon which an organization model is built are: agents, soft and hard goals, tasks, resources, and dependency links (Figure 2). They are depicted in organization models as nodes of directed graphs. An agent may represent any kind of active entity within the organization, a human operator, a team, a process, or a machine (e.g., the target system). A goal (hard or soft) defines a desired status within the organization, without specifying how to achieve it. A task is a well-specified activity. A resource represents any entity in the organization, both physical and informational, that can be used by an agent. Goals, tasks and resources can be connected to agents by means of directed edges: the dependency links. Dependency links are used to describe dependencies, in the form of depender Æ dependum and dependum Æ dependee, where the depender and the dependee are agents, and the dependum could be a hard goal, a soft goal, a task or a resource. In particular: an agent is linked to a dependum when it depends on it, that is, when it desires that goal to be achieved, or it needs that task to be performed or it requires that resource to be provided; a dependum is linked to an agent when it depends on that agent to be achieved/performed/provided. Thus, in the pattern depender Æ dependum Æ dependee, the depender depends upon the dependee in order for the dependum to be achieved/performed/provided.

![Figure 2. The organisation context before the ERMS](image-url)

Thus, in Figure 2, the agent Secretary receives from the enclosing context (out of the scope of our analysis) the incoming documents, which then s/he passes to the Head of Unit. The Head of Unit depends on the Secretary for receiving the documents and for the soft goal show the most important first. Depending on the corresponding subject, the Head of Unit will decide to which employee the document has to be assigned. For this, and for the safe conservation of the document, he or she will depend upon the Archivist. On its turn, the agent Employee (modeling...
any employee) will depend upon the Archivist for the soft goal *provide the document as soon as available*, in order to have the maximum amount of time to work on it. Finally, we can see how the Archivist depends upon the Employee to receive the document back once the work has finished, and upon the Personal Computer (hosting a stand-alone documents managing system), to generate/seek the document reference (and then place/fetch it to/from the Physical Archive).

In REF, dotted lines are used to bound the internal structure of complex agents; that is, agents containing other agents. The Organizational Unit is a complex agent.

5. From high-level goals to system and organizational requirements

**Start-up phase (Figure 1)**

According to REF, the first step of the approach consists in building an initial model of the organization, suitable to identify the scope of the problem, the initial high-level goals, the main agents involved and their interactions.

The project analysed in this work (ERMS) is one of the various results emerging from a broader effort aiming at improving the organization [4]. In particular, at a higher organizational level, the decision was made of adopting new technologies to transform the internal structure into a more creative, knowledgeable and citizen-centric environment, to have more services and information online, increase employee participation, and improve collaboration with other partner organizations.

Such a situation is modelled in Figure 3. Here, two soft goals have been placed upon the Head of Unit, as the agent in charge of the Organizational Unit: *exploit ICT to increase performance while avoiding risks*, and *cost/effective and quick solution*.

**Goal modelling phase (Figure 1 - step a)**

After building the initial organization model, the next step is to model the emerging goals. For a soft goal, the modelling process aims at making explicit the embedded softness: a soft goal model allows the analysts and the stakeholders to produce a set of tasks, hard goals, and constraints that precisely defines the way to achieve the initial soft goal. For a hard goal, the modelling process seeks to determine how to achieve it: a hard goal model allows the analysts and the stakeholders to express a combination of subordinate hard goals, tasks and resources needed to accomplish the initial hard goal (by definition, no soft goals can emerge from a hard goal refinement).

![Figure 3. Start-up phase: the initial organization model](image)

![Figure 4. The "exploit ICT ....." Soft Goal Model](image)
In the following, we focus on the soft goal *exploit ICT to increase performance while avoiding risks*. For the sake of clarity, the modelling process has been divided into two steps. In Figure 4 we can see a possible strategy that the *Head of Unit* (as result of a personal choice or of a negotiation with the upper organizational level) could apply to achieve the assigned goal: s/he may decide to *increase personal performance*, to *increase productivity* of the whole unit, and also to *avoid risks due to new technology*. To achieve the first sub-goal, *increase personal performance*, the *Head of Unit* will ask to have an *easy document access*, and to *increase process visibility*, in order to be able to take better-informed decisions. On its turn, this last goal will spawn two other goals, *provide employee’s performance* and *provide process performance*.

Also in goal models the arrowhead lines indicate dependency links. Goals decompositions may be conjunctive (all the sub-components must be satisfied, to satisfy the original goal - label “A” on the links) or disjunctive (it is sufficient that only one of the components is satisfied - label “O”, as in Figure 7).

Figure 5 shows how, for example, in order to have an *easy document access*, the *Head of Unit* will require to have no filter from the secretary (compare with Figure 2), and to have a multi channel access to the documents; this is further refined into a set of hard goals, specifying the different access channels to the documents that s/he would like to have.

Figure 5 shows also how the analysts and stakeholders can deal at the same time with different aspects of the solution space, i.e., technical, organisational, and social ones. Aside the technical aspects, such as, the technologies to employ to provide a multi channel access to the documents, the stakeholder can also identify the need of intervening at organizational level (in terms of structures or policies), or at personnel level. For what concerns the organisation, for example, the *Head of Unit* recognises that to *increase productivity* of the Organisational Unit, it will be necessary to *reduce administrative process constraints*, starting from recognising the legal value and allowing the use of electronic signature on the documents. At the same time, to *avoid risks due to the new technology*, the *Head of Unit* sees as crucial to guarantee a smooth transaction towards the new environment, which spawns the constraint *maintain the process structure*. The social and human factors, instead, come into the picture when the *Head of Unit* requires the employees to be more productive (*employee more productive* soft goal).

![Soft Goal Model](image_url)
recognising the legal value of electronic signature), others will be instead passed out, having the Head of Unit decided to depend on other agents for their achievement.

In Figure 5, these items are in bold outline: for example, the soft goal employee more productive will be passed out to the Employee agent, whereas the hard goal PDA for reading documents will be passed out to the ERMS. For such a reason, they are not further analysed here, but will be refined as further agreement between the Head of Unit and the agent that will be appointed of their achievements. This refinement usually results in the identification of some reciprocal dependencies, underling the “contractual” nature of the agreement between the two social agents. Of course, in the case the appointed agent is the IT system, prescriptive requirements are simply imposed on it.

**Organization modelling phase (Figure 1 - step b)**

The results of previous goal analysis allow us to enrich Figure 3, leading to Figure 6. Here, some new agents have been introduced: the Employee and the Archivist which have to be more productive, the IT (Information Technology) unit, which has to guarantee security and the agent representing the final system, the ERMS. From Figure 6, we can also see that the Head of Unit has decided to delegate the soft goal cost/effective and quick solution to the IT 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, and it will generate its owns goals, as for example, the easy to integrate and the prefer market solutions soft goals.

Fig. 6. The evolving organization model

**Completing the process**

At this point, iteration of steps (a) and (b) may be carried on several times, until the desired level of details has been reached. For example, let’s see how the Employee will try to be more productive (Figure 7). Again, it is worth noticing that the resulting operationalisation of the soft goals may uncover both technical (e.g., documents access trough the Internet, adopt known technologies, etc.), and organizational (e.g., allow working from remote, activate call centre) issues. The Archivist, instead, will require the ERMS to exactly replicate the functionalities and the interface provided by the already available system, by formulating the constraint adopt functionalities and interface already in use.

Figure 7. The “be more productive” Soft Goal Model
The results of the previous goal analysis allow extending the organisational model of Figure 6. The resulting organization model, partially illustrated in Figure 8, includes all the goals, tasks, and constraints emerged during goal analysis activities, similar to those presented above.

Figure 8. The final organization model

6. Supporting architecture and implementation strategy definition

REF allows the analysts to model the application context at a socio-technical level [3]. Starting from the high-level organizational goals, in fact, REF allows the analysts to obtain the main requirements for the target software system (e.g., ERMS), expressed as a collection of hard goals, tasks, and constraints placed upon it by the various stakeholders (i.e., agents) operating within or outside the organization. Collectively, these hard goals, tasks and constraints form the initial set of system requirements, a set of “macro requirements” that identifies the main system functionalities and properties, and establishes the basis for further requirements refinement and the subsequent development activities.

At this point, the analysts will have to deal with a purely technical system, and an analysis technique more suitable for such systems could be adopted. In [9], in particular, it is illustrated how REF can successfully act as forerunner for UML-based approaches. The REF output, in fact, as set of hard goals, tasks, and constraints placed upon the target software system by well-defined agents, not only results to be entirely consistent with “use-case modelling” (i.e., the front-end activity of any UML-based approach), but also enhances it. Both the research community and the business world, in fact, clearly recognize that use-case analysis can substantially benefit from supporting methods, which help to identify both the actors, which interact with the system, and their motivating goals, crucial for a successful use-case analysis [17].

Before proceeding in this detailed analysis, however, the graphical nature of REF can provide a useful support in identifying the main logical and/or physical architectural components of the new system and the corresponding (possibly different) implementation strategies. Complex IT systems, in fact, usually consist of a synergic combination of different components, some to be developed ad-hoc, some to be acquired on the market, but that require customisation, some others already available (legacy components) and that could (or have to) be reused. By adopting REF, the analysts and the stakeholders can easily visualize such a situation, and identify the combination of components and development strategies that best matches the set of goals, tasks and constraints placed upon the system.

For the ERMS, this analysis step is illustrated in Figure 9. Although only partially reproduced for the sake of clarity, Figure 9 shows how the ERMS has been modelled as a complex agent encompassing four different agents (highlighted in grey), each of them modelling a ERMS
component, and depending on each others for different goals. The resulting components, and their implementation strategies, are the following:

- **Document Management System**: to provide advanced and sophisticated documents management mechanisms, it represents the kernel of ERMS, has to be acquired from the market (i.e., to satisfy the soft goal *prefer market solutions*), and has to be easy to integrate with the legacy components (i.e., to satisfy the soft goal *easy to integrate*);

- **Data Base system**: to provide the basic storage capability, it is a legacy component, obtained by simply augmenting the capabilities of the already available data base system;

- **Archivist access system**: to provide functions specific for the archivists (e.g., documents classification, digitalisation, etc), it has to be developed *ad-hoc*, by replicating the already available stand-alone document managing system used by the archivist (i.e., to satisfy the constraint *adopt functionalities and interface already in use*);

- **Web-based access system**: to provide access via web, it has to be acquired from the market (i.e., to satisfy the soft goal *prefer market solutions*), also if some customisation is necessary in order to introduce an electronic-signature capability and the compatibility with the browser already in use.

In performing the refinement, the main indications into which components split the ERMS have been obtained by clustering the goals, tasks and constraints placed upon the ERMS (i.e., the macro requirements) on the basis of their similarities: e.g., based on similar technologies (web-based access); required by the same agent (the Archivist); aiming at achieving similar goals (mobility-related goals, such as *multi-channel access* and *enable remote collaboration*).

Similarly, the macro requirements, and in particular the constraints, have provided useful indications about the implementation strategies. For example, where feasible, although some customisation activities were necessary, solutions available on the market have been preferred (as required by the corresponding soft goal); for one component, instead (the Archivist access), it has been necessary to developed it, although on the basis of the already available tool, to satisfy the constraint *adopt functionalities and interface already in use*.

![Fig. 9. ERMS main architectural components](image-url)
7. Conclusions

REF has been adopted throughout the project decision-making process leading to the final ERMS. Domain visualization has provided a valuable support across the most critical stages of the project, by supporting the analysts and the stakeholders in capturing, understanding, and dealing with a large amount of different issues that populate the solution space, at a technical, organizational, and social level. REF models have allowed the analysts to discuss with both domain experts and direct users about how to reorganise the application context around a new system, or which functionalities should have been provided. In particular, by providing clear models of the domain, REF has allowed to:

a) quickly understand and analyse the original application context;
b) identify organizational goals and capture the corresponding achievement strategies;
c) formalise the initial set of the system macro requirements;
d) support the identification of the main system logical architectural components and of their corresponding development/acquisition strategies.

Once the project is over and the system deployed, the knowledge acquired by applying REF does not extinguish its role. On the contrary, made easily accessible and reusable through the different graphical models into which has been captured, such a knowledge offers potential benefits also in the post-deployment phases, supporting system maintenance and evolution.

The clear links established (through the different models) between organisational goals and system requirements, in fact, allow the analysts and the stakeholders to quickly identify the effects that changes in the organizational goals or new technology trends (e.g., web services, open source, etc.) may have upon the system. The possibilities offered to the application context by a new access technology (e.g., an integrated mobilephone and PDA device) could be easily evaluated by observing the corresponding goal models (i.e. Figures 5 and 7): 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 stakeholders to better achieve their goals (e.g., to provide the employees with only one rather than two different devices). In the same way, a change in an organizational goal may easily be translated into requirements changes.

8. References


