JOURNAL OF SOFTWARE MAINTENANCE AND EVOLUTION: RESEARCH AND PRACTICE
J. Softw. Maint. Evol.: Res. Pract. 2004; **16**:255–276 (DOI: 10.1002/smr.296)

Research

An assessment strategy for identifying legacy system evolution requirements in *e*Business context

Lerina Aversano and Maria Tortorella*,†

RCOST—Research Centre on Software Technology, Department of Engineering, University of Sannio, Pal. Ex-poste, Via Traiano, 82100 Benevento, Italy

SUMMARY

The enactment of *e*Business processes requires the effective usage of the existing legacy applications in the *e*Business initiatives. Technical issues are not enough to drive the evolution of the existing legacy applications, but problems concerning the perspectives, strategies, and business of the enterprises have to be considered. In particular, there is a strict relationship between the evolution of the legacy systems and the evolution of the *e*Business processes.

This paper proposes a strategy to extract the requirements for a legacy system evolution from the requirements of the *e*Business evolution. The proposed strategy aims at characterizing the software system within the whole environment in which its evolution will be performed. It provides a useful set of attributes addressing technical, process, and organizational issues. Moreover, a set of assessment activities is proposed affecting the order in which the attributes are assessed. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: evolution strategy; systems evaluation; system assessment; system re-engineering; process model; business rules; *e*Business; business process re-engineering

1. INTRODUCTION

The diffusion of the Internet requires the evolution of traditional business models in order to include *e*Business capabilities, defined as any Internet initiative that transforms business relationships, whatever those relationships might be: business-to-consumer, business-to-business, intrabusiness, or even consumer-to-consumer [1].

In this context, since information systems are determinant in supporting the execution of the business processes, the analysis of the assets of the existing software systems is a significant need.

*Correspondence to: Maria Tortorella, RCOST—Research Centre on Software Technology, Department of Engineering, University of Sannio, Pal. Ex-poste, Via Traiano, 82100 Benevento, Italy.
†E-mail: tortorella@unisannio.it

Received 7 June 2003
Revised 29 October 2003
Accepted 16 February 2004

Therefore, before assessing a business process, it is necessary to address problems concerning the role of the supporting legacy software systems. The aim is to integrate the enterprise's systems with those of the trading partners for facilitating business-to-business exchanges [2] or involving the customers with an active role in the enterprise's business processes.

Conceptually, evolving business processes [3,4] should entail discarding the existing legacy systems to develop new software systems that meet the new business needs. However, legacy systems cannot be simply discarded because they are important for the business they support and encapsulate a great deal of knowledge and expertise regarding the application domain. Thus, in order to help an enterprise to react more effectively to changing business requirements and technological innovations, it is necessary to focus even on the legacy system evolution strategy. Dually, the system evolution also has to be driven by the analysis of the organization hosting the process to be evolved [4,5].

Many approaches consider the legacy system evolution as a stand-alone problem. In reality, a legacy software system is embedded in the process using it and it can only evolve with the changes enacted by the process evolution. In reality, the motivation for requiring the process evolution is similar to that for interventions of *adaptive* and *perfective* maintenance. In fact, the evolution can be due to modifications in business processes and software products to keep them usable in a changed or changing environment and to improve their performance and/or maintainability. This paper proposes a strategy for supporting the assessment of a legacy system and defining its evolution requirements from the *e*Business requirements. The strategy aims at characterizing the legacy system within the whole environment in which the system evolution will be performed and analyses some attributes addressing technical, process, and organizational issues.

The strategy proposed in this paper represents an evolution of the approach proposed by the authors in [6], in which the research context was described and the whole requirements identification process was presented. Moreover, in [6], some of the attributes to be evaluated were identified, relating to organizations that want to achieve some improvement goals, technologies to be adopted and processes and supporting software systems to be evolved. On the basis of the experience acquired in the meantime, this paper refines the attributes previously introduced and adds new ones. In addition, the strategy proposed provides a systematic means describing the order in which the attributes have to be evaluated and how they interact with each other.

The challenge of the research is to identify the business knowledge captured in the legacy systems and to enhance it with the requirements of the *e*Business process evolution. Moreover, the strategy identifies the evolution requirements and opportunities by focusing on the enterprise mission and future vision. In fact, different enterprises consider the market requirements and the opportunities offered by the emerging technologies in different ways. Therefore, an enterprise moving to *e*Business has to identify which market needs are significant for its own mission, which internal processes are involved in the improvement, which emerging technologies will result in significant benefits, and which existing legacy applications are impacted by the process improvement and are, therefore, candidates for evolution toward the identified innovative technologies.

The strategy considers the legacy system evolution process in the wider context of the *e*Business process improvement and aims at addressing issues regarding the definition of a systematic approach that faces problems concerning: the legacy system evolution within the *e*Business process improvement; the planning of the evolution of large legacy systems according to the planning of the business process evolution; the identification of the most critical factors of success for evolving the legacy systems within the *e*Business context; the evaluation of the appropriateness of the chosen

migration strategy to the *e*Business context; and the possibility of evolving the legacy systems without impacting the processes in which they are used.

The strategy constitutes a methodological component within a research project called *Legacy to e-Legacy* (L2-eL), which aims to support the managers to move business toward *e*Business. The following section describes the aims of the project, its context and the reference framework. Section 3 presents some related work. Section 4 introduces the strategy proposed to support the evolution of legacy toward *e*Legacy, while Section 5 describes the assessment activities discussing the attribute measurements. The subsequent section outlines a case study to illustrate the applicability of the proposed strategy. Final remarks and directions for future work are given in the concluding section.

2. THE CONTEXT: THE L2e-L PROJECT

L2e-L is a research project aiming at developing a framework of methodologies, techniques and tools to transform traditional software infrastructures into new ones, successfully exploiting the opportunities introduced by the Information and Communication Technologies (ICT). The acronym L2e-L represents the methodological framework to reach this goal. In particular, the two expressions 'L' and 'e-L' indicate the two views of an Enterprise Information System: one refers to the traditional legacy systems and the other the *e*Legacy systems exploiting the potentiality of the ICT.

The cited framework has to support the:

- analysis and assessment of the legacy information systems by using innovative approaches;
- specific re-engineering processes for the traditional Enterprise Information System in order to move toward systems exploiting the advantages offered by the ICT;
- realization of an advanced technological platform to support the defined re-engineering process.

The framework aims to prove the following:

(a) an analysis methodology that, on the basis of the current legacy system needs and taking into account the impact of *e*Business on systems and processes, represents a support for identifying the evolution strategy;
(b) a software environment aiming to support the development of new software components;
(c) a software tool for estimating the performances of the renovated legacy systems resulting from the re-engineering process.

The L2e-L framework, depicted in Figure 1, will be supported by a software environment enabling the implementation of the re-engineering process from L to e-L, in a cooperative virtual environment distributed on the Internet.

The framework considers the process innovation and the legacy system evolution activities integrated into continuous monitoring of both the external innovation drivers and the status of the internal processes. All the involved activities are overviewed by the *Change Management* component which organizes the changes and improves the performance within and beyond the boundaries of the organizations.

The component in the framework named *Innovation Strategy Planning* aims to plan a strategy to fulfil the business needs. The definition of an innovation strategy depends on the enterprise size and scope. It is also influenced by the explicit role in the business of the knowledge coming from both the

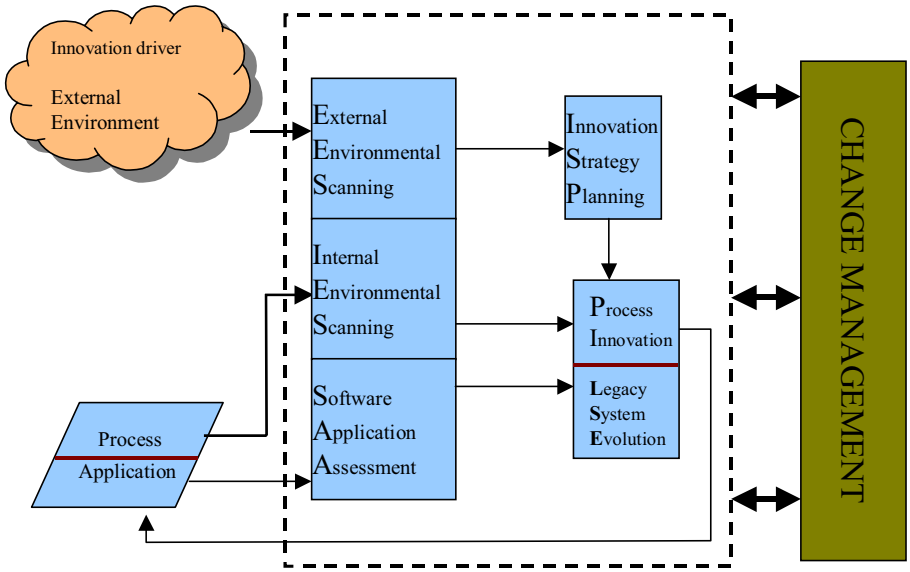


Figure 1. The L2e-L reference framework.

internal measures of the organization and the external environment. For this reason, Innovation Strategy Planning considers the input received from the *External Environment Scanning*, which observes the external innovation drivers and formalizes them in terms of strategy requirements.

The *Internal Environmental Scanning* framework aims to analyse the internal business processes of the enterprise and, with the support of the innovation strategy plans, guides the process innovation. In this context, the software applications embedded in the business processes are assessed by the *Software Application Assessment* component, in order to be renovated and integrated into the new process view.

The measurement and evaluation activities of the Environmental Scanning block represent critical aspects of a successful re-engineering process. They aim at collecting both quantitative and qualitative data needed to define the process design changes. Only when the customers' requirements are understood can an enterprise begin to define process performance and how to measure it. Both external and internal measures are critical in assessing processes and products/services. The external perspective keeps the organization customer focused. The internal measures help to discover the potential and actual sources of problems. The data to be collected are identified and gathered through interviews, observations and questionnaires.

The *Process Innovation* is the framework component entailing the application of the *re-engineering process* in order to guide the evolution of business toward *e*Business on the basis of the improvement goals derived from the Innovation Strategy Planning. It interacts with the *Legacy System Evolution* in order to define the business requirements and conduct the identification of the legacy system

evolution requirements. On the basis of the identified evolution requirements, the *Legacy System Evolution* is the component of the framework that provides methodological support during the evolution of the existing legacy applications by involving activities of *migration*, *re-engineering*, *network integration*, *remodularization*, etc. [7,8]. Figure 1 encapsulates the Process Innovation and Legacy System Evolution components in the same box to highlight that they have to proceed together and cooperate with each other in order to achieve the innovation goals.

3. RELATED WORK

Some research activities have been focused on the relationships between the business processes and embedded software systems when the organization exploiting them is evolving.

In [10], the authors underline the role of information technology in the move towards a process-oriented view of the management. They argue that, besides other factors influencing the outcome of Business Process Re-engineering (BPR) projects, such as top management support and project management, legacy software systems have a critical impact. Moreover, they present a framework for understanding BPR, which is based on a study of 12 European organizations from a cross section of industries dealing with BPR projects. From their study they evidenced that the outcomes of BPR strategies are influenced by the state of legacy software systems.

In [11], the authors suggest an integral view of business process re-engineering, based on both strategic and technological needs at the operational level. The software re-engineering activity aims to use existing corporate resources in a more economic way, i.e. reusing application knowledge to maximum extent, while software engineering approaches concentrate on handling software development according to the end-user requirements within a given amount of time and with limited cost.

Existing systems are the result of large investments and represent a patrimony to be salvaged [12]. In addition, developing a replacement system from scratch often requires excessive resources and entails unaffordable costs and risks. Even though these resources are available, it would take years before the functions provided by a legacy system could be taken up by new reliable components. Thus, legacy system migration strategies are often preferred to replacement [13]. Therefore, to satisfy the goals of the enterprise evolution requirements, it is necessary to work intensely to find a trade-off between the constraints of the legacy systems and the opportune strategy to be adopted.

A relevant problem is how to make a decision about the legacy systems. There are a number of options available. Typical solutions include discarding the legacy system and building a replacement system; freezing the system and using it as a component of a new larger system; modifying the system to give it another lease of life. Modifications may range from a simplification of the system (reduction of size and complexity) to preventive maintenance (redocumentation, restructuring, and re-engineering), or even extraordinary processes of adaptive maintenance (interface modification, wrapping, and migration) [14–16]. These alternatives are not mutually exclusive and the decision on which approach, or combination of approaches, is the most suitable for any particular legacy system should be assumed on the basis of an assessment of its technical and business value.

It is widely documented in the literature [8,17] that migrating a legacy system towards a new networked platform entails costs and risks that depend on the characteristics of both the architecture of the source system and the target platform. In addition, to perform a BPR project, it is

necessary to take into account the possible impact of the process redesign on the existing legacy systems. It is important to evaluate the costs of the different available options and, in particular, replacement with standardized packages, when available, possibly after a reverse engineering stage, or migration.

Several authors have identified possible alternatives for dealing with legacy systems and have proposed decision frameworks to choose between the alternatives. Bennett *et al.* [7] identify six strategies to deal with a legacy system: discard, wrap, outsource, freeze, carry on with maintenance and support, and reverse engineer. They stress the need to model the business strategy of an organization from a top-down perspective, including many stakeholders, to make informed decisions about legacy systems. They introduce a two-phase model, called SABA (Software As a Business Asset) that uses an organizational scenario tool to generate scenarios for the organization's future and a technology scenario tool to produce a prioritized set of solutions for the legacy system. Prioritization of solutions is achieved by comparing the available legacy systems with the systems required by each scenario generated by the organizational scenario tool.

Sneed [8] suggests that five steps should be considered when planning a re-engineering project: project justification, which entails determining the degree to which the business value of the system will be enhanced; portfolio analysis, which aims to prioritize the applications to be re-engineered based on their technical quality and business value; cost estimation, regarding the estimation of the costs of the project; cost–benefit analysis, in which costs and expected returns are compared; and contracting, which entails the identification of tasks and the distribution of effort.

In general, decision frameworks are required to assess a legacy system from two points of view: a business dimension and a technical dimension [7–9].

In [15], a typical decision framework is proposed for the management of legacy systems based on the following four main alternatives: ordinary maintenance, replacement, evolution and re-engineering/migration. The ordinary maintenance decision is generally adopted in the case of a system with a good technical and business value; this alternative entails only ordinary interventions on the system, aiming to add new functionality or resolve specific problems. By contrast, elimination of the system and replacing it with a new one, developed *ad hoc* or acquired from the market, is generally forced by the low technical and business value of the existing system. The evolution alternative aims to elaborate the system by providing it with new functionalities or adapting the existing ones in order to improve the business value of the system, while maintaining the high technical value. The last option is the re-engineering/migration alternative, whose goal is more complex. It entails moving the legacy system toward a new and more flexible environment, while retaining the original system's data and functionalities. Typically reverse engineering, restructuring, and architecture transformation interventions are suggested to improve software technical attributes in a system with high business value.

The relationships among the organizational and process aspects, and the software systems have already been considered with reference to the development of new software systems [18–22]. These studies focus on capturing the organizational requirements for defining how the system fulfils the organization's goals, why it is necessary, what are the possible alternatives, etc. A developed technique, referred to as i* [21,22], represents these aspects. i* offers two models to represent organizational requirements: the Strategic Dependency Model and the Rationale Dependency Model. The former focuses on the intentional relationships among organizational actors, while the latter permits one to model the reasons associated with each actor and their dependencies.

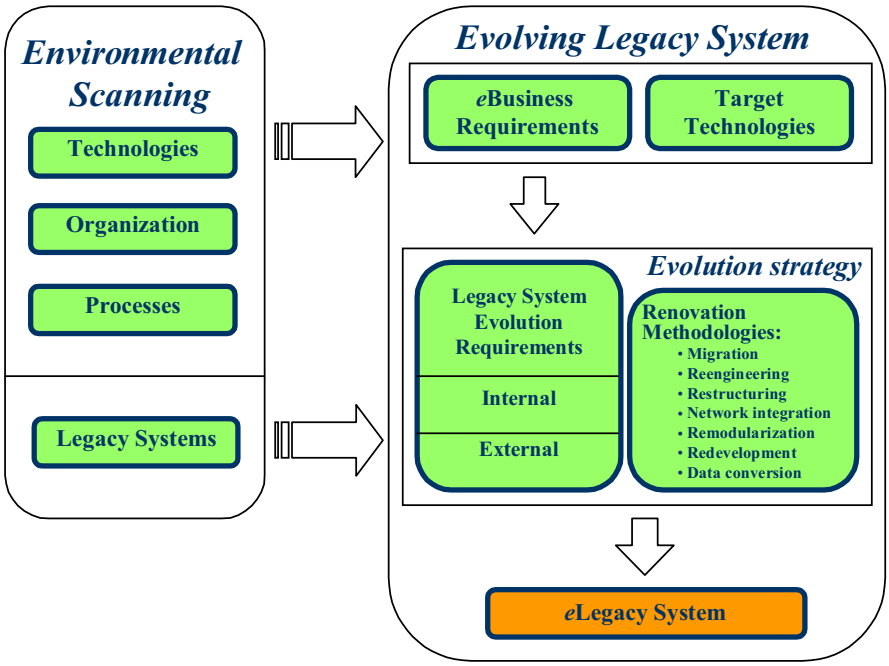


Figure 2. High-level view of the evolution legacy towards *e*Legacy.

Besides i* [20–22], a family of goal-oriented requirements analysis (GORA) methods, such as KAOS [19] and GRL [18], have been proposed as top-down approaches for refining and decomposing the customers' needs into more concrete goals that should be achieved for satisfying the customers' needs. However, all these methods concern the definition of the requirements for the development of software systems.

This paper aims to exploit the experience gained from goal-oriented requirements analysis with reference to the development of new software systems and applying it to the evolution of existing software systems.

4. THE ASSESSMENT ATTRIBUTES FOR EVOLVING LEGACY TOWARD *e*LEGACY

The assessment strategy proposed in this paper focuses on the problems concerning why/which/how legacy systems are impacted and need to be evolved during a project of process re-engineering.

The strategy refers to the environmental scanning block of the framework depicted in Figure 1. It provides guidelines on how to evaluate a set of quality attributes with reference to the evolution of legacy systems. Figure 2 depicts the high-level representation of the strategy and identifies the

basic aspects that managers need to consider before evolving legacy applications. In particular, the Environmental Scanning block of the figure corresponds to the homonymous blocks in Figure 1 and includes the four aspects to be analysed during the definition of the evolution strategy, namely *technologies*, *organization*, *processes* and *legacy systems*. The *technologies* aspect has to be considered as the change requirements often derive from the perspectives offered by the new technologies. In fact, the technologies permit one to satisfy new missions and business process needs, overcome the technological obsolescence, and decrease the maintenance costs. The role of the *organization* and *processes* aspects is very relevant because the current barriers to success are frequently aspects that are not technical, but mainly related to management and innovation issues. The *technologies*, *organization* and *processes* aspects permit one to define the *e*Business requirements of the business toward *e*Business evolution, and to identify the target technologies. This information, together with the characteristics of the *legacy systems* to be evolved, is used in the *Evolving Legacy System* block of Figure 2, which provides the *Legacy System Evolution* block given in Figure 1 in greater detail. In summary, the gained information is translated in terms of *Evolution Strategy*, which includes the definition of the *Legacy System Evolution Requirements*, specifying the functionalities of the legacy system that need to be added and/or evolved and the *Renovation Methodologies* to be adopted in the evolution process. The Legacy System Evolution Requirements are classified as external or internal on the basis of the interaction level with the actors external to the enterprise. In particular:

- *external requirements* are referred to as the actions to be performed on the legacy functionalities in order to improve the business processes from the external actors' point of view;
- *internal requirements* are referred to as the actions to be performed on the legacy functionalities in order to improve the business processes from the internal actors' point of view.

The following sub-sections describe in greater detail the set of attributes considered for each aspect and the relevant technical and management issues for enacting the evolution legacy toward *e*Legacy. They have a general interpretation, and are consequently useful for any kind of business categories. Therefore, full specifications and definitions are required with reference to the workplace and application domain of the improvement intervention. In particular, the specification of the existing attributes and/or the insertion of new ones can customize the approach on the basis of the specific context and gained experience.

4.1. Organization

The term *organization* is used with reference to a quantitative significant set of people, groups, policies, activities, resources and relations unified by a common goal. In this context, the organization has to be process oriented and take into account the employees' needs and enterprise mission. A set of attributes has been identified to characterize the organization, with the aim of tracking the impact of its status on the requirements for the process evolution and, consequently, the legacy system evolution in order to achieve the improvement goals. Some of the identified attributes are as follows:

- *Vision*—to describe the business objectives of the enterprise and identify if criteria are used for verifying and, eventually, describing them.
- *Structure*—to identify a set of details regarding the kind of organizational model adopted; how stable the structure is with reference to the chosen model; the document status of the

model structure; who is responsible for making decisions; which are the roles involved in the processes; which is the skill required; and which is the distribution of roles among the employees.
- *Management strategy*—to define the balancing policy used between processes and organization and describe the distribution of business processes among the enterprise's components.
- *Organization procedures*—to point out the standard procedures used by the enterprise with reference to its organization and processes.
- *Documentation*—to identify the documentation status over the whole organization.
- *Organization to process*—to determine how the processes are distributed over the organization structure.
- *Technological dependencies*—to identify the technological trends of the organization.
- *Technological coverage degree*—to establish the degree of diffusion of the innovation technology over the organization.

4.2. Processes

The *processes* aspect has a central role in the proposed approach, because it establishes the connections between the more abstract *organization* layer and the more practical *legacy system* and *technology* layers.

The following attributes have been identified for evaluating the enterprise's processes.

- *Process management*—to identify the way the processes are managed, the distribution of the resources among them and how closely they are related. The analysed properties are *roles and responsibilities* of each actor participating to the process; *resource and time management*, to identify the resources and time spent for the management activities; and *effort scheduling*, to relieve the existence of mechanisms used to establish the effort required to perform the process activities.
- *Procedures*—to determine the existence and usage of reference standards and guidelines for the process enactment and the related formalization level.
- *Process to activity*—to identify the relationships existing among the activities within a process. This attribute involves the analysis of the dependences among the tasks of the processes in order to define their interaction level and the relative formalization degree; the activity decomposability, representing the granularity existing in the actual decomposition of a process in activities; and the flexibility for changing the process decomposition.
- *Internal and external process interfaces*—to investigate the interactions and related interfaces of the process with the internal and external actors it involves.
- *Automation level*—to determine how the information technologies are used to support the actors of the process while performing the activities of the process.
- *Process efficiency*—to identify an overall view of the performance of the process analysed from the point of view of the manager. It is evaluated in terms of the results the activities produce and the resources they require. The resources are measured in terms of time, costs, and productivity.
- *Quality of the process technological covering*—to verify how many process activities are supported by the existing software systems and how adequate the technological solution is in terms of products that are electronically produced with respect to the required products.

4.3. Technologies

The *technologies* aspect has a significant impact on the methodologies for the management of the legacy systems. The possibility to evolve legacy systems derives directly from the potentiality of the target technologies. However, to provide adequate guidelines, a detailed characterization of the technologies is also required. Indeed, to effectively exploit their potentiality, it is relevant to evaluate them with respect to the contribution and improvement they can introduce in a business process.

Some of the most important attributes concerning the technologies are listed below:

- *Portability*—to identify the effective usability of the technologies in different hardware/software configurations and operating environments. The attribute identifies properties such as being *multi-platform*, when the technology can be used with different operating systems; being *multi-language*, when it can be used with different languages; being *multi-enterprise*, when it can be used by different enterprises with adequate customization; being *multimedia*, when it can manage different types of information (documents, video, sound, etc.).
- *Reliability*—to identify the existence of important security properties and a security layer for the business processes.
- *Maturity*—to analyse the effective usage of methodologies to improve the technology performance on the basis of the experience and the consequent investment in new innovative versions.
- *Diffusion*—to identify the use of a technology in different working environments and its integration into different kinds of business processes.
- *Interoperability*—to evaluate the capability of a technology to be integrated and supported by other technologies using standard protocols.

4.4. Legacy systems

A *legacy system* can be described in terms of the attributes evaluated by analysing technical perspectives in a wider context that considers the business processes. This is particularly relevant when the evolution from business toward *e*Business is considered.

A list of attributes, characterizing the legacy system aspect, is as follows:

- *Adequacy*—to identify how adequate the system is for supporting the current activities of the process to be evolved and, possibly, the evolved processes. Also, how suitable it is for satisfying the involved users' needs.
- *Configuration*—referring to the software/hardware configuration of the environment hosting the legacy system to be evolved.
- *Documentation status*—to determine the existence of the design documentation, its reliability, completeness and rigorousness with reference to the current version of the system. Likewise, the attribute analyses if the available documentation has been updated during previous maintenance interventions.
- *Architecture*—to identify the decomposability of the legacy system and its integration level with the external software components.
- *User interface*—to describe the status and kind of system user interface.

- *Software components interfaces*—to consider the interaction level of the internal legacy components and the type of interfaces existing between each pair and among them and eventual external components, and the availability of the relative formal description.
- *Performance*—to consider the efficiency related to the use of the legacy system within the business process to be evolved.
- *Source code*—to indicate if the source code is available in order to determine the possibility of analysing and evolving the legacy system. In addition, the existence of the source code and the knowledge it provides in terms of programming language used, environments and paradigm, permits to establish the possibility of evolving the software system.
- *Data dependencies*—to evaluate the dependence level of the legacy system from the data it accesses, the data quality and exportability, and the relevance of the data within the business process to be evolved. Information about the data is important as it represents a relevant part of the organization's knowledge, and the decision to evolve a software system is also influenced by the possibility of recovering the managed data.

5. EVOLUTION ASSESSMENT ACTIVITIES

Figure 3 shows the evolution process. Each activity considers the evaluation of some attributes concerning the four aspects described above. Figure 3 depicts in greater detail the information presented in Figures 1 and 2; the order in which the activities are executed is also given. The execution order is worth noting as it affects the order in which the attributes are considered, so that an impact relationship among them is established. In fact, some measures have to be gathered before others. Table I describes how the measures of the attributes flow among the activities and integrates Figure 3. In particular, the table refers to the four aspects described above. The headings of the columns list the activities given in Figure 3, while the headings of the rows refer to the attributes that have to be evaluated. In the table, the attributes are grouped in terms of aspects. Each row highlights an attribute and indicates the activities considering it and those using it.

A description of each activity is given in the following subsections, with emphasis on the measured attributes. The following description does not consider all the attributes, but cites just some examples, with further considerations given in the table.

5.1. Analysis of the organization

It aims to characterize the organization with reference to those attributes that have to be considered to perform the next activities, that regard the identification of the process to be evolved and the formulation of the evolution requirements. In particular, Table I highlights that the attributes the activity considers concern the organization aspect. For example, the activity analyses the *vision* impacting on the choice of the process to be re-engineered and the selection of the technologies to be adopted in the evolving business processes. In addition, the *vision* will also be considered to identify the requirements for the process evolution. Furthermore, if a process view of the organization activities exists, the attribute *organization to process* is considered, together with all the other attributes in the table, to support the identification of the processes. The organization attributes referring to the technology have a relevant role in the identification of what to evolve and how, as the technologies are assumed to be the driving force behind the changes.

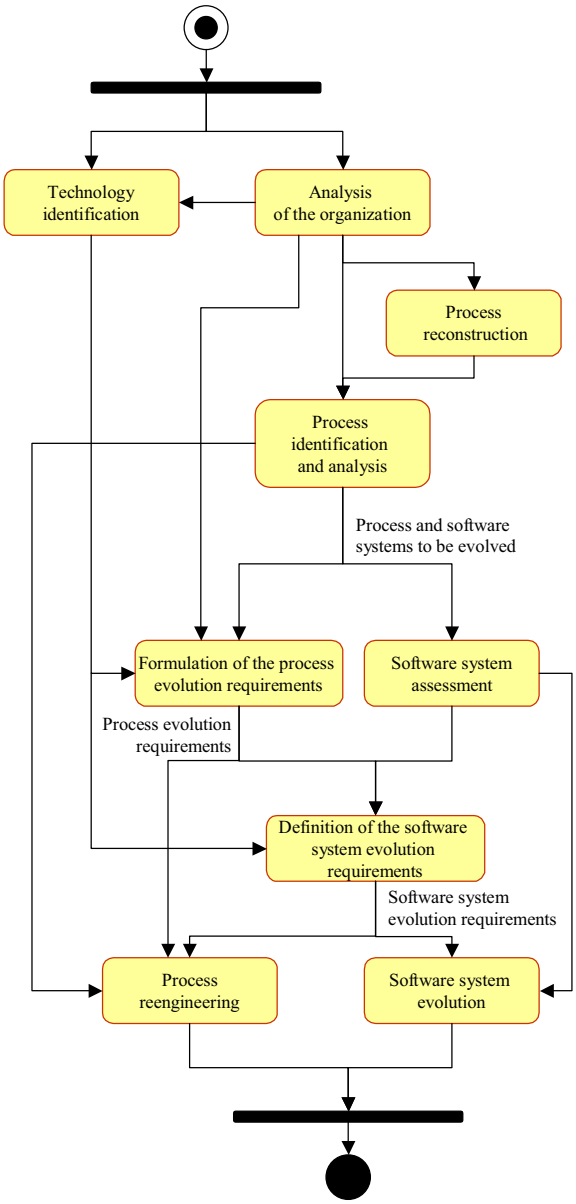
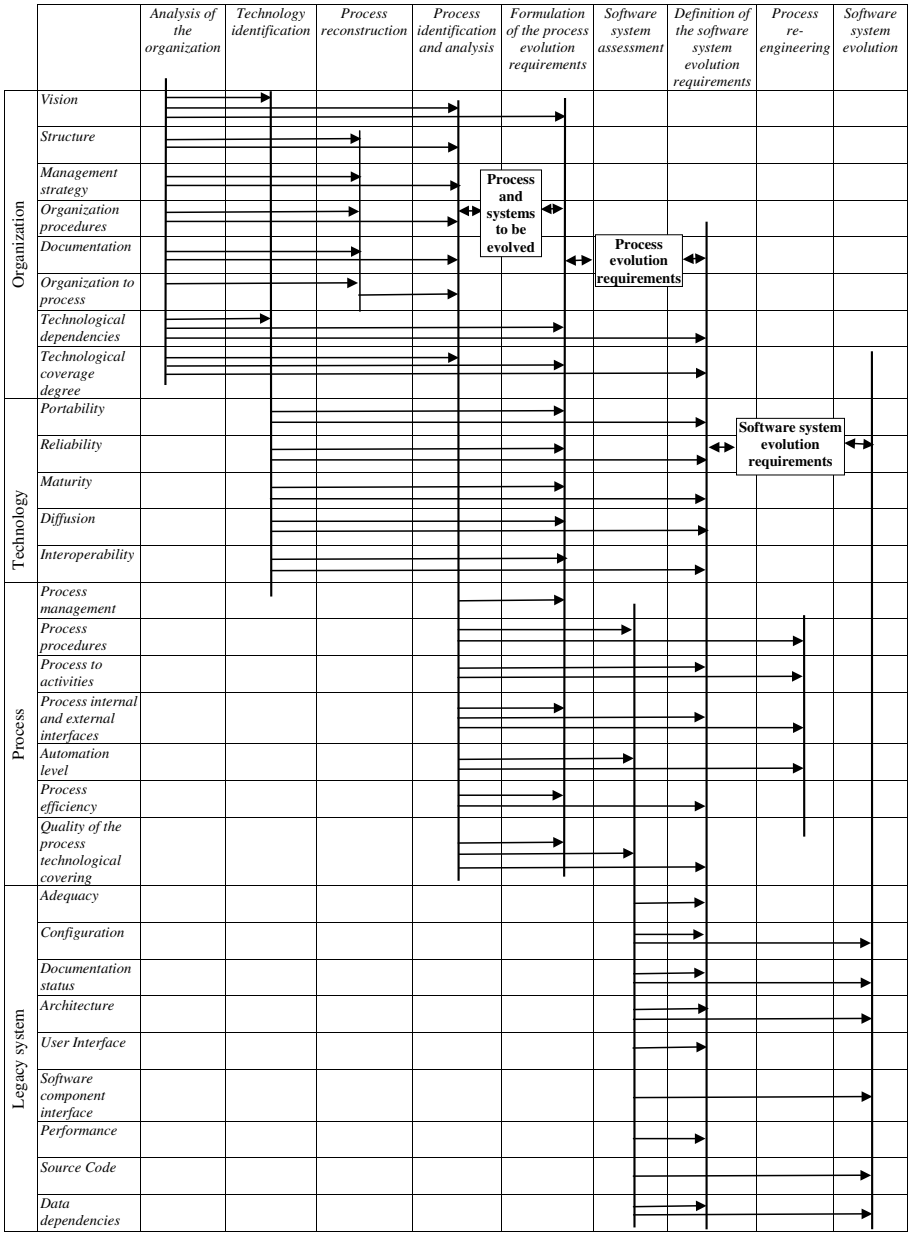


Figure 3. Activity diagram for process and system evolution.

Table I. Activity–attributes relationship.



5.2. Process reconstruction

If a process view of the organization does not exist, it is necessary to construct it by analysing the *organization* attributes and by conducting interviews with managers and employees. An important role is played by the availability of the *documentation* describing the organization. The main output is the defined process view.

5.3. Process identification and analysis

The process to be analysed for re-engineering is selected on the basis of the information received from the *analysis of the organization* activity. This represents the internal organization's components that are potentially impacted by the change that will be introduced by achieving the future *vision* of the organization and by the adoption of the chosen technologies. Table I details the attributes this activity considers. They belong to the *processes* aspect and are used for designing the change and translating it in terms of evolution requirements. In particular, some *process* attributes are considered for identifying the process to be evolved and supporting software systems. For example, all the information about the *process management* and *process efficiency* is necessary to identify the quality of a process and to guide the future process evolution. By analysing the *process internal and external interfaces*, it is possible to identify the level of interaction with external actors; in fact, a process with an intense interaction with external factors is a good candidate for evolution. It is very important to identify the process tasks requiring the actors' interaction and the *quality of the process technological covering* in order to understand if supporting software systems exist and if they need to be evolved or developed from scratch. The listed attributes are also considered for the *definition of the software system evolution requirements* activity, while the other attributes are considered for guiding the evolution process. For example, the *process to activities* attribute is considered in order to establish if the relationship existing among the process activities allows the implementation of the evolution requirements. The *automation level* attribute is necessary for identifying the software systems to be assessed and their evolution requirements. Clearly, the values of these *processes* attributes will evolve after the re-engineering of the process.

5.4. Technology identification

On the basis of the *vision* and *technological dependencies* attributes of the *organization* aspect, the emerging technologies to be considered in the evolution process are identified and evaluated by analysing the attributes of the *technologies* aspect. Figure 1 shows that this choice influences the definition of the requirements for evolving both business processes and supporting software systems. In particular, as already cited, the *interoperability* and *portability* attributes have to be considered when a business process has to be evolved toward an *e*Business process.

5.5. Formulation of the process evolution requirements

On the basis of the *organization vision* attribute and evaluating the technologies to be adopted and the process to be evolved, the evolution requirements can be formulated. In particular, as already described, the *process management*, *process efficiency*, and *process internal and external interfaces*

are attributes impacting on the definition of the evolution requirements. For example, if the process interfaces external actors, the interaction can be made easier and more flexible by introducing Web technologies, which can also facilitate the activity distribution and cooperation among the internal actors. Moreover, before proceeding to formulate the process evolution requirements, it is necessary to analyse the *quality of the process technological covering* and consider which benefits the introduction of the new technologies can bring and how suitable the evolution of the automatic solution supporting the process is to the *Vision* requests. Figure 3 and Table I indicate that the output of this phase is represented by the *process evolution requirements*.

5.6. Software system assessment

The identification of the process to be evolved affects the decision on which software system should be considered for the evolution. In fact, if a software system is embedded in the business process to be evolved, it has to be evolved too in order to satisfy the evolution requirements. To fulfil this purpose, this activity analyses an identified software system for evaluating its adequacy to the evolving process and establishes if it efficiently supports the business requirements, and then takes a decision about its future. Based on this analysis, two alternative decisions can be assumed: substitute the software system with a new one or evolve it congruently with the process evolution. Once decided, the system has to be assessed. With this in mind, the attributes of the *legacy systems* aspects can be considered. Some attributes, such as the availability of the *software system documentation* and *source code*, help one to understand if software system evolution is possible. In fact, these items have to be evolved together with the software system. Likely, the *data dependencies* attribute can influence the adoption of an evolution strategy.

5.7. Definition of the software system evolution requirements

The requirements to be satisfied from the evolution of a software system are defined on the basis of the process evolution requirements and many of the attributes considered in the previous activities. In fact, the above information indicates the activities impacted by the evolution and, if the *process to activities* and *process internal and external interfaces* attributes are well defined, it is possible to identify the software system components that may be affected by the software system evolution. Additional considerations, analogous to those formulated for the *formulation of the process evolution requirements*, can be made. Considering the *quality of the process technological covering* attribute and the characteristics of the *technologies* aspect helps one to recognize whether the software system components, affected by the change, are adequate to satisfy the evolution requests or if they require a re-engineering intervention. At the *legacy systems* aspect level, the *architecture* attribute, indicating the software system decomposability, the way the composing software sub-systems interface with each other and/or with the users, and the availability of good quality documentation and source code, permits one to state the possibility of evolving the software system.

5.8. Process re-engineering

This consists of two main tasks: new business model definition and new business model implementation. The former considers the process redesign from two different points of view: the

process and the social. The process redesign seeks to leverage technology in order to improve process performance by considering the introduction of the chosen innovative technologies. The social redesign seeks to leverage human potential to the same end. The following implementation of the new business model is focused on developing and implementing plans to transform the process from its current state to the future state. The strategies to implement the new process and its associated technological and human resource systems must be executed, and the re-engineered process must be performed company-wide. Once the re-engineering activity has been completed, the obtained results, the assumptions made and the attributes previously considered have to be reviewed, challenged, and changed in order to establish the future state of the process.

5.9. Software system evolution

On the basis of the software system evolution requirements and the results of the software system assessment, it is possible to proceed with implementation of the evolution. Some of the evolution strategies that can be applied are listed in Figure 2. The choice of the strategy depends on the values assumed by the legacy systems attributes and the quality of the available software system items. For example, if the software system documentation and source code are not available or the software system adequacy to the process evolution requirements is very low, the re-development of a new system may be the chosen strategy. Conversely, if the system is not decomposable, it is necessary to re-modularize it [23] before executing migration activities [13]. The new evolved system may require activities concerning the data conversion [24,25] in order to allow continuation of the informative system management. In any case, the aim is to identify the software system components impacted by the evolution requirements. To achieve this, the authors are analysing the possibility of considering the user interface behaviour for identifying those components interacting with the end-users when they are involved in the process activities to be evolved.

Table I shows that the output of some phases does not consist of attribute values but of information about the processes and systems to be evolved and the evolution requirements. It is clear that the processing of this information may require a review of the obtained results and the re-execution of the activities that produced them. For example, the *process evolution requirements* may need to be formulated again if they cannot be achieved as no technological solution can be adopted for supporting them. This requires rework of the *formulation of the process evolution requirements*.

6. A CASE STUDY: A PROCESS OF A LOCAL PUBLIC ADMINISTRATION

In order to evaluate the effectiveness of the strategy proposed in this paper, a case study with pilot users has been conducted. In particular, a peripheral department of the Public Administration, the Town Govern of Maddaloni (a town in the district of Caserta in Italy) agreed to participate in the case study as a customer. Particular attention was paid to the process analysis and re-engineering activities of the *process innovation* and *legacy system evolution* blocks of Figure 1.

All the activities performed involved process analysts, known BPR methodologies and technologies, and the personnel of the Town Govern of Maddaloni, who knew the application domain. The improvement needs of the evolution were identified by interviewing the department managers. The managers highlighted that the mission of a public administration body is to provide services to

the citizen, and the future vision includes increasing the citizen's satisfaction, as well as improving the effectiveness of the employees' tasks. The citizen's satisfaction could decrease with increasing time between putting forward a request and receiving the related answer. However, the length of time necessary to find an answer may be caused by the manual execution of the needed tasks and the time taken to receive information from external organizations.

Then, two goals emerged from the vision analysis concerning the *reduction of the citizen's waiting time* and the *increasing of the tasks' effectiveness*. The next evaluation concerned the choice of the pilot process to be analysed, evaluated and re-engineered. A discussion among all the participants led to the identification of the *tax payment process* as the first candidate to be re-engineered, as it involved the citizen, employees and external organizations. It was considered as a critical process as it required the amount of taxes the citizen had to pay to be checked. In order to analyse the identified process, the administrative manager was interviewed and, at a first level of abstraction, three sub-processes were identified: payment of the land taxation; payment of taxes on cities' solid waste; and payment for occupation of public ground. In particular, the first sub-process was considered. In the following, it is indicated with the acronym ICI—Imposta Comunale sugli Immobili. From the analysis of the information provided, the first model, describing the different activities, involved actors, produced objects, and used technologies, was elaborated.

The Unified Modelling Language (UML) [26,27] and its business modelling extension [27] was chosen as the process modelling language. Different diagrams were used to model different aspects. As Figure 4(a) shows, *activity diagrams* were used to model the flow of the process activities, including decisions and synchronizations. Figure 5(a) highlights that *use-case diagrams* have been adopted to model environmental aspects, namely which actors (roles) participate in which use-cases (activity or group of activities). The ICI process was partly automated by two legacy applications: one supporting the *protocol* activity, and one supporting the *payment form predisposition* and *data comparison with land register* activities. The two legacy applications were single-user applications, not accessible through the Internet, and requiring manual insertion of data in their databases. In addition, the two applications accessed two different databases containing the same information about the citizen, resulting in duplication of the data and repeating the operations to insert the data.

The technologies identified for reaching the improvement goals and evolving the ICI process involved using the Internet. From the analysis of the process model and by considering the attributes previously described, the high-level goals of the organization were refined in requirements impacting the process and supporting legacy systems. In particular, the requirements for the process improvement were defined from two different perspectives:

- An external perspective, in which the improvement goal concerning the citizen was considered and *e*Business requirements were identified from the citizen's point of view. The new *e*Business process would permit the user to interact directly with the system, check the status of his registration, pay the tax on-line and receive on-line communications from the public administration body.
- An internal perspective, in which the improvement goal concerning the employees participating in the process was taken into consideration for identifying other business requirements. The aim was to improve the process by increasing its automation level and decreasing the manual operations.

The first columns of Tables II and III list some of the external and internal requirements of the process evolution to move toward *e*Business. Satisfaction of the identified requirements of the

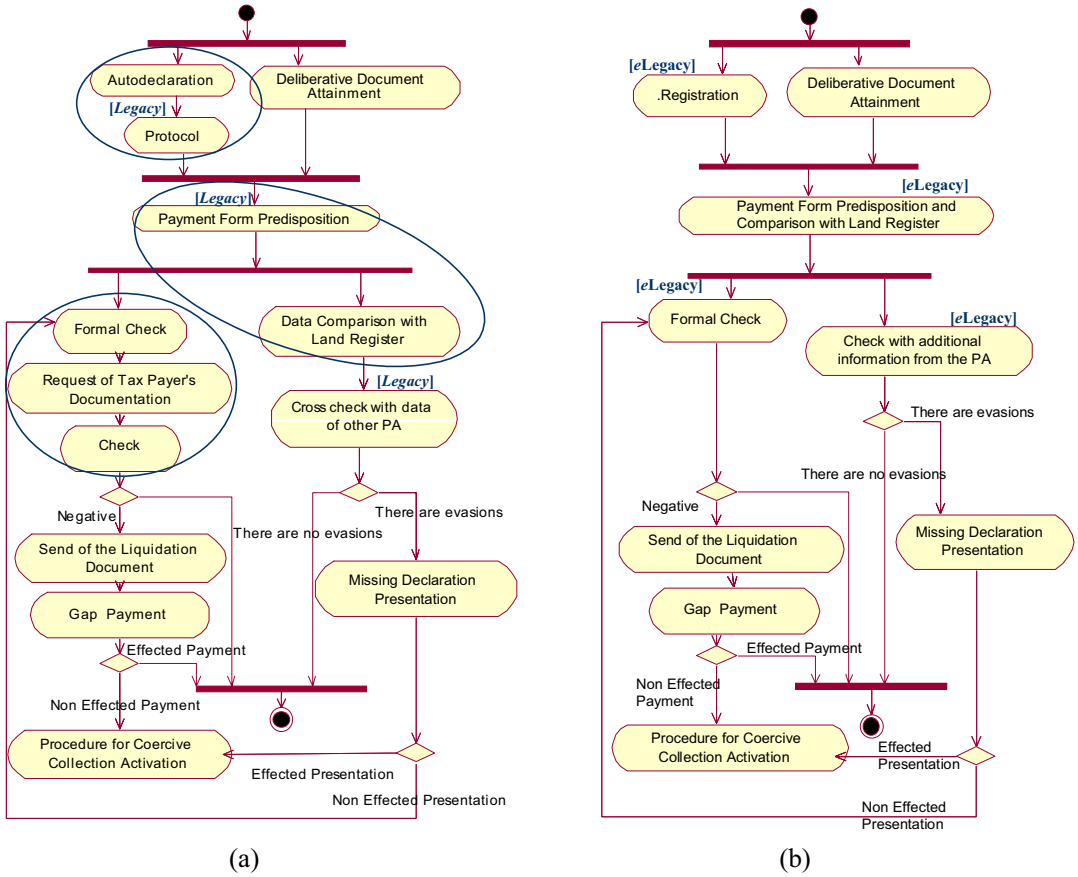
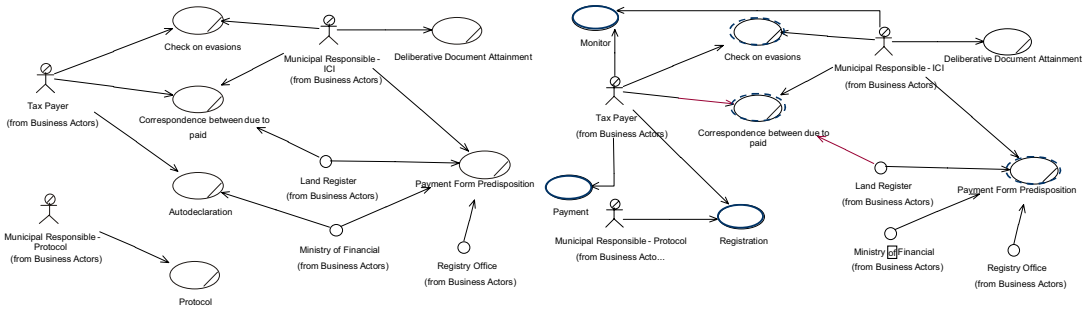


Figure 4. Activity diagrams modelling the process before and after re-engineering.

re-engineering of the ICI process was achieved by considering the impact of the evolution of the legacy applications and the implementation of new functionalities.

For example, on analysing the *process efficiency* attribute of the ICI process it emerged that the most expensive activities were those involving extensive use of human resources, such as the *payment form predisposition*. This allowed the identification of the evolution requirement 'to give the possibility to the citizen to perform the on-line payment', resulting in a decrease in the employees' workload and an increase in the process productivity. Moreover, analysing the *quality of the process technological covering* has highlighted that the software system supporting the *payment form predisposition* activity was not adequate to sustain the improvement requirement and its evolution was required. The related software system evolution requirements are listed in Table II. The chosen evolution strategy was



(a) (b)

Figure 5. Use-case diagrams of the ICI process before and after re-engineering.

Table II. External *e*Business and legacy system requirements.

*e*Business requirement	Legacy system evolution requirement
To allow the citizen to register himself by accessing the Internet	• To migrate the legacy user interface to the Web • To migrate the legacy system from a single-user version to a multi-user version
To allow the citizen to monitor the current status of his registration	• To migrate the data component of the system • To implement new functionalities supporting the management of the user profiles
To give the possibility to the citizen to perform on-line payments	• To evolve the existing user interfaces in order to collect and/or show new data to the user • To integrate the legacy system with external components responsible to manage the payment

migration, made possible by the availability of *documentation* and *source code* and the decomposability of the system.

Figures 4(b) and 5(b) depict the model of the evolved ICI process. Figure 4(a) highlights the more impacted activities that are grouped as new activities in Figure 4(b), with a reduction in the process complexity. The grouping was possible thanks to the evolution of the legacy system and the introduction of new functionalities. The second columns of Tables II and III list for each *e*Business requirement the set of requirements for evolving the legacy system. The tables show that most of the requirements concern the migration of the legacy aspects to Web technologies, or the implementation of new functionalities that are characteristic of Web applications, such as security and user authentication. In particular, by evolving the legacy systems towards *e*Legacy, the introduction of the Web technologies favours the active participation of the citizen (tax payer actor) in the *e*Business process. Figure 5(b) shows the impact of the process evolution on the interaction of both external and internal actors with

Table III. Internal *e*Business legacy system requirements.

Process re-engineering requirements	Legacy system evolution requirements
To check automatically the amount to be paid by the citizen and the performed payment	• To implement new functionalities that automatically check the correctness of the payment
To update the tax percentage	• To implement new functionalities for changing the tax percentage and integrate them into the existing data model
To optimize the tasks' effectiveness in terms of resources and time spent	• To implement new functionalities that automatically cross-check the internal data with those obtained from other public administration departments

the process components. The interactions of external and internal actors are highlighted in the figure by solid and dashed circles, respectively.

Other requirements shown in Tables II and III relate to data migration aimed at facilitating the interaction among the evolved modules of the *e*Legacy system through a unified database.

7. CONCLUSIONS

The research described in this paper addresses the need for evolving legacy systems in the *e*Business context. The current trend towards *e*Business calls for changes in the fundamental concepts of business process innovation and re-engineering. Indeed, traditional views of process innovation and re-engineering have focused on the internal components of an enterprise with the aim of redesigning the business processes in order to for increase efficiency. However, nowadays it is important to react to the central role that the Internet is playing in the definition of new business relationships within and among enterprises. The emergence of virtual and distributed organizational models and the richness of the communication channels between customers and enterprises have resulted in a wider view of the re-engineering methodology, which overcomes the border of the enterprise to involve both customers and business partners.

Therefore, process innovation cannot be performed without a careful evaluation of the impact of process changes on the legacy systems and the introduction of new technologies. The adoption of new software systems to satisfy the new *e*Business requirements is not a practical solution. Economic and technical constraints in most cases force the evolution of the legacy systems as opposed to development of new systems from scratch. This means that a balance must be achieved between the constraints imposed by the legacy systems and the opportunities offered by the re-engineering of the business processes. In other words, the requirements for legacy system evolution must be extracted from the requirements of the evolution of business toward *e*Business.

In this paper, a strategy has been proposed to support the identification of the legacy system evolution requirements in the context of business process innovation. The strategy considers the continuous monitoring of the external and internal changes that drive, respectively, the innovation of the enterprise's organization and processes.

The strategy presented has been developed within a collaborative project with a large enterprise and is now being tested in two different application domains: a bank and public administration.

The first stages of the experimentation, described in the case study section, aimed to investigate the practical applicability of the strategy and to verify the effectiveness of the support provided by the proposed parameters for extracting the requirements of the evolution of the legacy systems. Indeed, the process innovation in the peripheral public administration department is characterized by the fact that there are few critical legacy systems, but the rules governing the processes tend to be very rigid, and the redesign of the processes must be coherent with the current laws.

Because a strategy is applicable, it has to be formalized in a rigorous manner, and it must also be supported by an integrated software environment that helps to apply the proposed assessment strategy and collect quantitative and objective information about the analysed business processes at organizational, process and technological levels [28]. In addition, the software environment should support the identification of the strategies to be applied during the evolution process [29]. The authors are actually involved in the implementation of such an environment [5]. It is based on Web technologies and is used to support the cooperation among the actors involved in the assessment and evolution processes.

REFERENCES

1. Alter S, Scott J, Ein-Dor P, Vessey I, Markus ML. Does the trend toward e-Business call for changes in the fundamental concepts of information systems? A debate. *AIS Communications of Association of the Information Systems* 2001; **5**(10):1–59.
2. Sneed HM. *Business Reengineering in the Age of the Internet*. Case Consult at Wiesbaden: Wiesbaden DE, 2000.
3. Hammer M, Champy J. *Reengineering the Corporation: A Manifesto for Business Revolution*. HarperCollins: New York NY, 1993; 256 pp.
4. Jacobson I, Ericsson M, Jacobson A. *The Object Advantage: Business Process Reengineering with Object Technology*. ACM Press: New York, 1995; 368 pp.
5. Allen P. *Realizing e-Business with Components*. Addison-Wesley Longman: Boston MA, 2001; 254 pp.
6. Aversano L, Tortorella M. Evolving legacy systems towards eLegacy in eBusiness context. *Proceedings IEEE European Conference on Software Maintenance (CSMR 2003)*. IEEE Computer Society Press: Los Alamitos CA, 2003; 201–210.
7. Bennett KH, Ramage M, Munro M. Decision model for legacy systems. *IEE Proceedings Software* 1999; **146**(3):153–159.
8. Sneed HM. Planning the re-engineering of legacy systems. *IEEE Software* 1995; **12**(1):24–34.
9. Verdugo G. Portfolio analysis—managing software as an asset. *Proceedings of the International Conference on Software Maintenance/Management*. Software Maintenance Association: New York NY, 1988: 17–24.
10. Light B, Holland CP. The influence of legacy information systems on business process reengineering strategy. *Proceedings of Business Information Management—Adaptive Futures (BIT '98)*. Manchester Metropolitan University: Manchester, U.K., 1988 (CD Rom).
11. Bernd J, Clifford TYT. Business process reengineering and software systems strategy. *Technical Report n.11*, Institut für Wirtschaftsinformatik, Universität Tübingen, Tübingen DE, 1994; 29 pp. http://www.uni-tuebingen.de/wi/forschung/Arbeitsberichte(3)/ab_wi11.ok/ab_wi11.pdf [21 May 2004].
12. Bennett KH. Legacy systems: Coping with success. *IEEE Software* 1995; **12**(1):19–23.
13. Brodie ML, Stonebaker M. *Migrating Legacy Systems—Gateways, Interfaces and Incremental Approach*. Morgan Kaufmann: San Francisco CA, 1995; 210 pp.
14. Canfora G, Cimitile A. A reference life-cycle for legacy systems. *Proceedings of the Workshop Migration Strategies for Legacy Systems (Technical Report TUV-1841-97-06)*, Information Systems Institute, University of Vienna, Vienna AT, 1997.
15. De Lucia A, Fasolino AR, Pompella E. A decisional framework for legacy system management. *Proceedings IEEE International Conference on Software Maintenance (ICSM 2001)*. IEEE Computer Society Press: Los Alamitos CA, 2001; 642–651.
16. Pigoski TM. *Practical Software Maintenance—Best Practices for Managing your Software Investment*. Wiley: New York NY, 1997; 384 pp.

17. Sneed HM. Risks involved in re-engineering projects. *Proceedings IEEE Working Conference on Reverse Engineering (WCRE '99)*. IEEE Computer Society Press: Los Alamitos CA, 1999; 204–211.
18. Antón AI. Goal-based requirements analysis. *Proceedings IEEE International Conference on Requirements Engineering* (ICRE '96). IEEE Computer Society Press: Los Alamitos CA, 1996; 136–145.
19. Dardenne A, van Lamsweerde A, Fickas S. Goal-directed requirements acquisition. *Journal of Science of Computer Programming* 1993; **20**(1):3–50.
20. Mylopoulos J, Chung L, Yu E. From object-oriented to goal-oriented requirements analysis. *Communications of the ACM* 1999; **42**(1):31–37.
21. Yu ESK. Modeling organizations for information systems requirements engineering. *Proceedings IEEE International Symposium on Requirements Engineering (RE '93)*. IEEE Computer Society Press: Los Alamitos CA, 1993; 34–41.
22. Yu ESK. Modelling strategic relationships for process reengineering. *Doctoral Dissertation*, Department of Computer Science, University of Toronto, Ontario, 1995.
23. Canfora G, Cimitile A, Visaggio G. Assessing modularization and code scavenging techniques. *Journal of Software Maintenance: Research and Practice* 1995; **20**(1):317–331.
24. Alhajj R, Polat F. Reengineering relational databases to object-oriented: Constructing the class hierarchy and migrating the data. *Proceedings IEEE Working Conference on Reverse Engineering* (WCRE 2001). IEEE Computer Society Press: Los Alamitos CA, 2001; 335–344.
25. Blaha M, Benson I. Teaching database reverse engineering. *Proceedings IEEE Working Conference on Reverse Engineering* (WCRE 2000). IEEE Computer Society Press: Los Alamitos CA, 2000; 79–85.
26. Penker M, Eriksson HE. *Business Modeling with UML: Business Patterns at Work*. Wiley: New York NY, 2000; 400 pp.
27. OMG. OMG Unified modeling language specification version 1.5. *Technical Report formal/03-03-01*, Object Management Group (OMG), 2003. http://www.omg.org/technology/documents/formal/uml_2.htm [21 May 2004].
28. Aversano L, Bodhuin T, Canfora G, Tortorella M. WebEv—a collaborative environment for supporting GQM measurement frameworks. *Proceedings IEEE Hawaii International Conference on System Sciences (HICSS-'04)*. IEEE Computer Society Press: Los Alamitos CA, 2004; 10012a.
29. Aversano L, Esposito R, Mallardo T, Tortorella M. Evolving legacy system towards e-business. *Managing Corporate Information Systems Evolution and Maintenance*. Idea Group Publishing: Hershey PA, 2005; in press.
30. Aversano L, Bodhuin T, Canfora G, Esposito R, Tortorella M. Critiquing business processes for their evolution towards e-Business. *Proceedings of ACM Symposium on Applied Computing*, SAC 2004. ACM Press: New York NY, 2004; 1351–1358.

AUTHORS' BIOGRAPHIES

Lerina Aversano received her Laurea degree in Computer Engineering from the University of Sannio, Italy, in 2000 and her PhD degree in Computer Engineering from the University of Sannio, Italy, in 2003. She is currently an assistant researcher at the RCOST—Research Centre on Software Technology at the University of Sannio. Her research interests include process and information system modelling, workflow management, document management, business process re-engineering, software re-engineering and migration.

Maria Tortorella is an assistant professor at the Engineering Faculty of the University of Sannio. She is research leader at the RCOST—Research Centre on Software Technology at the University of Sannio. Her research interests include reverse engineering, maintenance, software comprehension, business process re-engineering, business process and software system evaluation. She received a BS degree in Computer Science at the University of Salerno, a MSc in Computer Science at the University of Durham, and a PhD in Electronic Engineering and Computer Science at the University of Naples.

