

Migrating Legacy Systems to the Web

Lerina Aversano

University of Sannio, Italy

Gerardo Canfora

University of Sannio, Italy

Andrea De Lucia

University of Salerno, Italy

INTRODUCTION

Business Process Reengineering (BPR) is defined as “the fundamental rethinking and radical redesign of business processes to achieve significant improvements of the performances, such as cost, quality, service, and speed” (Hammer & Champy, 1993). Most BPR projects aim at converting business organisations from hierarchical centralised structures to networked decentralised business units cooperating with one another. This conversion is assuming a strategic relevance as the Internet is changing radically business processes, not only because they are purposely reengineered, but also because the Internet and the information and communication technology, offer more convenient means of fulfilling their requirement.

Current business processes have been profoundly fitted to the available software. The technologies involved in process execution impact the way businesses are conceived and conducted. Abstractly, BPR should entail discarding the existing and legacy systems to develop new software systems that meet the new business needs. This is superficially attractive and human appealing. However, legacy systems cannot be simply discarded because they are crucial systems to the business they support and encapsulate a great deal of knowledge and expertise of the application domain. This entails that even the development of a new replacement system may have to rely on knowledge which is encapsulated in the old system. In summary, existing systems are the result of large investments and represent a patrimony to be salvaged. Therefore, to satisfy the goals of a BPR project, it is necessary to work intensively to search a trade-off between the constraints of existing legacy systems and the opportune BPR strategy.

In this article, we discuss a strategy for migrating business processes and the supporting legacy systems to Web-centric architecture. The overall strategy comprises modelling the existing business processes and assessing the business and quality value of the supporting software systems. This is followed by the migration of the legacy systems, which can be in turn enacted with

different strategies. The initial step consists of understanding and modelling the business processes together with the involved documents and software systems. The analysis of the existing processes is required to get an inventory of the activity performed, compare them with best practices, and redesign and/or reengineer them. Our overall approach is discussed in details in references (Aversano, Canfora, De Lucia, & Stefanucci, 2002a; Canfora, De Lucia, & Gallucci, 2002b), together with experience concerned with their applications. The final phase related to legacy system analysis and assessment is discussed in details in Aversano, Canfora, and De Lucia (2003) and briefly presented here.

BACKGROUND

Legacy systems are “large software systems that we don’t know how to cope with but that are vital to our organization” (Bennett, 1995). There are a number of options available in managing legacy systems. 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 reengineering) or even to extraordinary processes of adaptive maintenance (interface modification, wrapping, and migration) (Pigoski, 1997; De Lucia, Fasolino, & Pompella, 2001).

Several authors have identified possible alternatives for dealing with legacy systems and have proposed decision frameworks to select among the alternatives. In general, decision frameworks require that a legacy system be assessed from two points of views: a business dimension and a technical dimension (Bennett, Ramage, & Munro, 1999; De Lucia et al., 2001; Sneed, 1995). This information measures the complexity of the business processes and administrative rules that a system, or system’s component, implements and their relevance to

Figure 1: A decisional framework

Decomposability value	High	Component based Incremental Migration	Short term Migration
	Low	Elimination / Replacement	Reengineering / Redevelopment
		Low	High
		Business value	

achieve business competitiveness. The technical value of a legacy system can be assessed through different quality attributes, such as the obsolescence of the hardware/software platforms, the level of decomposability, the maintainability, and the deterioration (De Lucia et al., 2001).

We assess the technical quality of a legacy system by considering the obsolescence and the decomposability level. In particular, we focus on making decisions on the actions to perform as a consequence of a BPR project aimed at taking advantage of the Internet. We assume that the obsolescence of the system is high, and therefore, extraordinary maintenance is required. Accordingly, the decision about the particular type of intervention to take will be made based on the decomposability and business value of the system.

Two different kinds of decomposability can be considered:

- *vertical decomposability*, which refers to the possibility of decomposing a system into major architectural layers;
- *horizontal decomposability*, which accounts for the possibility of decomposing a legacy system into independent and cooperating business components.

In particular, concerning the vertical decomposability, Brodie and Stonebaker (1995) refer that a software system can be considered as having three types of components: *interface components*, *application logic components*, and *database components*. Depending on how separated and well identified are these components, the architecture of a legacy system can be *decomposable*, *semidecomposable*, or *nondecomposable*. In a decomposable system, the application logic components are independent of each other and interact with the database components and system interfaces. In a semidecomposable system, only interfaces are separate

modules, while application logic components and database services are not separated. A nondecomposable system is a black box with no separated components.

Figure 1 shows a decisional framework that takes into account the considerations described previously. The decision about the intervention to take on the legacy system with a high business value is mainly driven by the vertical decomposability of the system. If the vertical decomposability value is sufficiently high, that is, the system is decomposable or semidecomposable in the Brodie and Stonebaker terminology, the best strategy is a short term migration of the system, achieved through wrapping the application logic and/or data management functions (that define the server tier) and reengineering/redeveloping the user interface to a Web-centric style. This strategy represents a good alternative also in the case of a nondecomposable system, provided that the costs and risks of its decomposition are affordable (Canfora, De Lucia, & Di Lucca, 1999; Sneed, 1995). If the decomposability level of a system with high business value is too low, the complete reengineering/redevelopment alternative has to be preferred, as the legacy system can still be used.

For legacy systems with a low business value, the decision about the intervention to take is mainly driven by the horizontal decomposability. In particular, if the horizontal decomposability value is high, it is possible to use again the framework in Figure 1 to make different decisions for the different business components. This provides the basis for a component based incremental migration of the legacy system. Whenever both the business value and the decomposability of a legacy system are low, the only possible option is elimination/replacement. Indeed, in this case, there are no adequate business preconditions to evolve the existing system.

The migration of a legacy system entails the reuse of the system components while moving the system toward newer and more modern technology infrastructure. Brodie and Stonebraker (1995) propose an incremental approach, named *Chicken Little*, to migrate a legacy information system using gateways. A different approach proposed by Wu et al. (1997) is the *Butterfly* methodology, which eliminates the needs of accessing both the legacy and new database during the migration process, thus avoiding the complexity induced by the introduction of gateways to maintain the consistency of the data. Both the methodologies aim at migrating a legacy system mainly based on its vertical decomposability.

Migration strategies have also been proposed that take into account the horizontal decomposability. Canfora et al. (1999) and Serrano, Montes de Oca, and Carter (1999) present two strategies for incrementally migrating legacy systems to object-oriented platforms. The main difference is in the method adopted for identifying objects in proce-

4 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/migrating-legacy-systems-web/14543

Related Content

Managing the Impact of Rapid IT Change

John Benamati and Albert L. Lederer (2010). *Information Resources Management Journal* (pp. 1-16).

www.irma-international.org/article/managing-impact-rapid-change/38907

From Information Model to Controllable Implementation

Hilary J. Kahn and Nick P. Filer (2001). *Information Modeling in the New Millennium* (pp. 324-344).

www.irma-international.org/chapter/information-model-controllable-implementation/22995

The Ontological Stance for a Manufacturing Scenario

Michael Gruninger (2009). *Journal of Cases on Information Technology* (pp. 1-25).

www.irma-international.org/article/ontological-stance-manufacturing-scenario/37391

A Review of User Involvement in Information System Projects

Bradford R. Eichhorn and Oya I. Tükel (2015). *International Journal of Information Technology Project Management* (pp. 26-53).

www.irma-international.org/article/a-review-of-user-involvement-in-information-system-projects/123464

Flexible Negotiation Modeling by Using Colored Petri Nets

Quan Bai, Minjie Zhang and Kwang Mong Sim (2009). *Journal of Information Technology Research* (pp. 1-16).

www.irma-international.org/article/flexible-negotiation-modeling-using-colored/4139