Chapter 27

ERP System Implementation from the Ground Up
The ERP5 Development Process and Tools

Rogério Atem de Carvalho
Fluminense Federal Institute, Brazil

Renato de Campos
State University of São Paulo, Brazil

Rafael Manhães Monnerat
Nexedi SA, France

ABSTRACT

The design and implementation of an ERP system involves capturing the information necessary for implementing a system that supports enterprise management. This process should go down through different abstraction layers, starting on enterprise modeling and finishing at coding. For the case of Free/Open Source ERP, the lack of proper modeling methods and tools jeopardizes the advantages of source code availability. Moreover, the distributed, decentralized decision-making, and source-code driven development culture of open source communities, generally does not rely on methods for modeling the higher abstraction levels necessary for an ERP solution. The aim of this paper is to present a development process and supportive tools for the open source enterprise system ERP5, which covers the different abstraction levels involved, taking into account well established standards and practices, as well as new approaches, by supplying Enterprise, Requirements, Analysis, Design, and Implementation workflows and tools to support them.

1. INTRODUCTION

Enterprise Resource Planning (ERP) software is, by definition, integrated business software. Therefore, modeling ERP means dealing with the aspects related to the different abstraction layers that must be taken into account on integrated management. The ultimate goal of developing an ERP system should be going from the highest abstraction level considered - enterprise modeling, down to code generation, without losing modeling information. In other words, it is the ideal situation of guaranteeing

DOI: 10.4018/978-1-60566-731-7.ch027
that the software is in complete conformity with business requirements. To accomplish this, it is necessary to define methods that can improve quality and provide persistence for modeling information throughout each abstraction level considered.

For the specific case of Free/Open Source ERP Systems (FOS-ERP), modeling methods have their importance increased, given that they can empower the availability of source code. Modeling is many times devaluated in the normally distributed, decentralized decision-making, and source-code driven development environment of open source projects. This matter becomes important since FOS-ERP are increasingly gaining acceptance for many reasons. One reason is direct cost, since they impose no licensing costs in general. Other reason is the perception that if customization is inevitable, why not adopt a solution that exposes its code to the adopting organization, which can freely adapt the system to its needs (Carvalho, 2006)?

Experience has shown that the analysis and documentation of business and software requirements by means of models are essential for the enterprise systems development, making necessary the use of proper techniques and tools (Odeh, Kamm, 2003). In this sense, a modeling architecture that properly contemplates business processes aspects can facilitate reuse and promote better functionality, performance, and system understanding, avoiding waste of efforts and resources (Campos, Carvalho, Rodrigues, 2007). Moreover, in the case of FOS-ERP systems, the advantage of free access to code can be jeopardized by the lack of references from where specializations of this code can be derived. Thus, for a FOS-ERP, the use of modeling methods and tools can lower risks and enhance competitive advantage through the access of every aspect that forms the development of ERP software. In other words, from enterprise models to source code, everything is opened for the adopter, that can freely adapt them to its needs (Carvalho, 2006).

This work presents a development process for the FOS-ERP ERP5, composed by set of activities and tools applied to each abstraction level considered: enterprise and business modeling, analysis, design, and code generation. The following sections will briefly present ERP5 framework, basic GERAM (Generalized Enterprise Reference Architecture and Methodology) concepts and their relation to ERP5, the process workflows and their specific tools, the tools that support the process as a whole, and finally conclusions and future directions.

2. ERP5

The ERP5 project (Smets-Solanes, Carvalho, 2002; Smets-Solanes, Carvalho, 2003) is a FOS-ERP that aims at offering an integrated management solution based on the open source Zope platform, written in the Python scripting language. This platform delivers an object database (ZODB), a workflow engine (DCWorkflow), and rapid GUI scripting based on XML. Additionally, ERP5 incorporates data synchronization among different object databases and a object-relational mapping scheme that stores indexing attributes of each object in a relational database, allowing much faster object search and retrieval, in comparison to ZODB, and also analytical processing and reporting. This project was initiated in 2001 by two French companies, Nexedi – its main developer, and Coramy – its first user, and since then is in development and use by a growing community from France, Brazil, Germany, Poland, Senegal, Japan, and India, among others. ERP5 is named after the five core business entities that define its Unified Business Model (UBM, Figure 1):

- **Resource**: describes an abstract resource in a given business process (such as individual skills, products, machines etc). Material lists, as well as prototypes are defined by the relationship between nodes.
Related Content

An Empirical Bandwidth Analysis of Interrupt-Related Covert Channels
[www.irma-international.org/article/an-empirical-bandwidth-analysis-of-interrupt-related-covert-channels/136464/](www.irma-international.org/article/an-empirical-bandwidth-analysis-of-interrupt-related-covert-channels/136464/)

Information Systems Curriculum Design Processes
[www.irma-international.org/chapter/information-systems-curriculum-design-processes/21091/](www.irma-international.org/chapter/information-systems-curriculum-design-processes/21091/)

Secure Key Generation for Static Visual Watermarking by Machine Learning in Intelligent Systems and Services
[www.irma-international.org/article/secure-key-generation-static-visual/39098/](www.irma-international.org/article/secure-key-generation-static-visual/39098/)

Formal Reliability Analysis of Embedded Computing Systems
[www.irma-international.org/chapter/formal-reliability-analysis-embedded-computing/76949/](www.irma-international.org/chapter/formal-reliability-analysis-embedded-computing/76949/)

Systematic Use of Software Development Patterns through a Multilevel and Multistage Classification
[www.irma-international.org/chapter/systematic-use-software-development-patterns/49164/](www.irma-international.org/chapter/systematic-use-software-development-patterns/49164/)