007-0782

ENTERPRISE MODELING FOR DEVELOPMENT PROCESSES OF OPEN-SOURCE ERP

Renato de Campos, UNESP-BRAZIL, rcampos@feb.unesp.br

Rogério Atem de Carvalho, CEFET-Campos, BRAZIL, ratem@cefetcampos.br

José S. Rodrigues, UNESP-BRAZIL, jsrod@feb.unesp.br

POMS 18th Annual Conference

Dallas, Texas, U.S.A.

May 4 to May 7, 2007

Abstract: The open source and free software might be an alternative to the Small and Medium Enterprises in order to cut down expenses, for example, using open-source ERPs. However, when adapting ERP to the enterprise's particular needs, the user keeps depending on the developers due to the lack of access and knowledge of the respective code. Free and open-source software may promote advantages to the enterprises, however, for its adoption it is necessary the development of techniques and tools in order to facilitate its implantation and code maintenance, targeting adaptations. This process should start on the enterprise modeling level and finish at the coding level, going down through different abstraction layers. After emphasizes the importance of enterprise modeling to development and adaptation process of open-source ERPs, this article presents a brief description of the ERP5 project and the architecture components to developments and adaptation process for ERP5 system.

Key words: ERP, open-source, free software, enterprise modeling, business process.

1 - INTRODUCTION

The MRP (Material Requirements Planning) approach was proposed by Joe Orlicky in the early 60s and came out with the purpose of executing the material planning activities by computer. This system is limited to the material flow treatment. However, this system evolved in the 70s, at the same time the information technology was being developed with the creation of a computing system with more comprehensive objectives, performing the main production planning and control activities, being called MRP II (Manufacturing Resources Planning) from that time on. The MRP II is considered to be a system in which decision-making is quite centralized, and based on the following basic principle: all programs established by the system will be accomplished in the most faithful manner, becoming little flexible when considering the work variations by labor force. The MRP II can be seen as a hierarchical management system, because the long-term plans is detailed by successive planning levels, since the plan reach the level of component and specific machines in the a more short-term planning level (SCHEER, 1989; CORRÊA *et al*., 2000).

The evolution of the MRP II systems and the interconnection of clients to suppliers results the creation of the system denominated Enterprise Resource Planning (ERP) (CHUNG & SNYDER, 2000). Then, from 1990 on, in the United States, and after 1996 in Brazil, the ERPs appeared, mainly targeted to integrate several enterprise departments.

Nowadays, the market is more and more competitive, thus the companies must react in order to follow the customer requirements (DAVENPORT, 1994). For the companies to organize their resources and obtain better planning to their production processes, one of the options is the implantation of the ERPs (Enterprise Resource Planning), which can help in the planning of the enterprise’s human and material resources (KALPIC & BERNUS, 2002). However, the price of these systems may be an obstacle for small and medium-sized enterprises (SME) willing to obtain such systems. The adaptation of the ERPs modules according to the characteristics of each enterprise may also become important to its competitiveness, but the closed systems make the companies depend on the payment of such services to the system developers, to that adaptations may be performed. The open source and free software might be an alternative to the SME in order to cut down expenses, for example, using open-source ERPs. Another advantage is the possibility of the software adaptation, allowing for the users to adjust processes or modules system to the reality of their enterprise by means of altering the open source, without being dependent on the developers or the owner of closed codes. However, it is somewhat difficult to adopt these ERPs, regarding the generation of these codes and the implantation of the systems in the enterprise. This matter is being dealt with on the ERP5 project (SMETS-SOLANES & CARVALHO, 2002). One of the proposals is the utilization of modeling architecture and reference models (SCHEER, 2000; MERTINS & JOCHEM, 1999; VERNADAT, 1996) since the documentation and good understanding of the business processes and information flow, which were considered when defining the ERP requirements and the original code generation, are essential to facilitate the definition of the enterprise's particular requirements and for the alteration of the relative codes (CAMPOS, CARVALHO & FERREIRA, 2006).

This article emphasizes the importance of enterprise modeling to development and adaptation process of open-source ERPs, and presents a brief description of the ERP5 project and the architecture components to developments and adaptation process for ERP5 system.

2 - ERP5 PROJECT

Nowadays, there are some free Free Open Systems (FOS) ERPs proposals, allowing the alteration or maintenance of their codes, as the ERP5 project (SMETS-SOLANES & CARVALHO, 2003). The latter is a free-open code ERP project which aims at offering a high technology solution and low cost for SMEs.

The ERP5 project 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 business and academical community from France, Brazil, Germany, Luxembourg, Poland, and India, among others.

The ERP5 is a FOS-ERP and is based on the open source Zope platform, written in the Python scripting language. This platform offers an object database (ZODB), a workflow engine (DCWorkflow), a content management infrastructure (CMF – Content Management Framework), and rapid GUI scripting based on XML (ZPT – Zope Page Templates). In addition, ERP5 incorporates data synchronization among different object databases, through the implementation of the SyncML XML based protocol, and a Object to Relational mapping scheme that stores indexing attributes of each object in a MySQL database, allowing much faster object search and retrieval, in comparison to ZODB. This system has five innovative technologies:

a) Multi - the system is multi-user, multi-enterprise, multi-lingual, multi-currency, multi-cost e multi-scenario;

b) Meta - offers several levels of details for the same management process;

c) Distributed - uses advanced synchronization mechanisms, allowing the distribution and data sharing without the need for permanent web connection;

d) Object-based - the employment of a set of objects allows for modeling and implementing complex support systems to the decision-making process;

e) Free - all the generated information, developed technologies and methodologies are freely available on the project website.

The structure of ERP5 instances is defined through subclasses of the five main concepts and other supportive classes. Its behavior is described through workflows, which implement business processes that manipulate the objects in accordance to descriptions supplied by Path objects. ERP5 is named after the five main business classes that define its core abstract model:

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;

2) Node: a business entity that receives and sends resources. They can be related to physical entities (such as industrial facilities) or abstract ones (such as a bank account). Metanodes are nodes containing other nodes, such as companies;

3) Path: describes how a node accesses needful resources. Paths are abstract entities;

4) Movement: describes a movement of resources among nodes, in a given moment and for a given period of time. For example, such movement can be the shipping of raw material from the warehouse to the factory. Movements are realizations of Paths;

5) Item: a physical instance of a resource.

The ERP5 is based on a model that can connect anything to a category. Some examples include a resource category (such as services, raw material, skills or money) or an enterprise category (such as a group of companies, a group of people or retail chain stores).

When an ERP is implemented, more than installing a new program in the enterprise's computers, a work methodology, or workflow, is being defined or adopted. For the development and implementation of a good information system, as well as for the development of the ERP5 system, it is necessary to use proper techniques of Software Engineering.

3 - SOFTWARE ENGINEERING AND REQUIREMENT ANALYSIS

A definition of software engineering was proposed is the establishment and use of solid engineering principles, so that trustable software can be obtained at a low cost, and work efficiently in real machines. Software engineering comprehends three fundamental elements: methods, tools and procedures. Regardless of the software development model, the development process contains three generic phases (PRESSMAN, 2002):

a) Definition phase – the software developer tries to identify which pieces of information need processing, which functions and performances are required, which interfaces must be established, which project restrictions and which validation evaluation criteria are required to define a successful system. In the definition phase, the applied methods vary according to the model function, however there are three specific steps: (i) system analysis, which defines the function of each element in a computer-based system; (ii) software project planning, which comprehends risk analysis, cost estimates and task definition and work programming; (iii) requirement analysis, which details the scope by means of the information domain analysis and the software function.

b). Development phase - define how the data structure and the software architecture must be designed, that is, how the project will be translated into a programming language and how the tests must be performed.

c). Maintenance phase – focus on the changes which are associated to error correction, adaptations and the functional improvement of the software.

As quoted, the requirement analysis is always present in the software definition phase, and it is formed by a group of techniques employed to search, detail, document and validate the requirements of a software product. For a long time, the concern was only about the software requirement analysis, in detriment of the proper analysis of the business requirements, which don't assure an effective connection between the needs for the business process informatization and the software project. The analysis and documentation of the business and software requirements by means of models are essential for the open source ERPs development, making necessary the use of techniques and tools. In this sense, a modeling architecture that properly contemplates the modeling of all aspects of the business processes, including the other aspects related to the software development, can facilitate the reuse, better functionality, better performance, and the better system understanding, avoiding waste of efforts and resources.

4 - FREE OPEN SOURCE ERP DEVELOPMENT AND ADAPTATION

ERP software is, by definition, integrated business software. Thus, modeling ERP means to deal with aspects related to the different abstraction layers that must be taken into account. The ultimate goal of developing an ERP system is to go from the highest abstraction level considered, Enterprise Modeling, down to code generation, without loosing modeling information. In other words, it is the ideal situation of guaranteeing that the software is in complete conformity with business requirements, from an integrated point of view. To accomplish this, it is necessary to define methods that can improve quality and modeling information persistence through each abstraction level considered (CAMPOS, CARVALHO, & FERREIRA).

For the specific case of FOS-ERP, modeling methods increase their importance, since they can empower the availability of source code through the capability for extending and changing source code that is adherent to enterprise modeling. And this kind of view is sometimes lost in the normally decentralized decision-making, distributed, and source-code driven development environment of open source systems.

This matter becomes important since FOS-ERP are increasingly gaining acceptance for many reasons. One reason is the perceptions 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.

Adapting is a crucial point to ERP , given that, despite being called *software*, enterprise systems in general have nothing to do with 'shrink-wrapped', 'off-the-shelf' items that can be used instantaneously. This fact reinforces the freedom of manipulating the code by itself in FOS-ERP: if the vendor changes its contract terms, the client company is not locked in to a particular solution supplier. If for every ERP the fact that integration among processes can by itself becomes a source of competitive advantage, it may be extrapolated to the possibility of changing source code to drive an even better advantage.

The analysis and documentation of business and software requirements by means of models are essential for the FOS-ERP development, making necessary the use of proper techniques and tools. In this sense, a modeling architecture that properly contemplates the modeling of all aspects of the business processes, including aspects related to software development, can facilitate the reuse, better functionality, better performance, and a better system understanding, avoiding waste of efforts and resources. Moreover, in the case of FOS-ERP systems, the advantage of free code modification 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 by the adopting organization will lower risks and enhance its competitive advantage through the access of every aspect that forms the development of ERP software. In other words, from enterprise models to generated code, everything is opened for the adopter, that can freely adapt it to its needs.

One of the proposals in the ERP5 project is the utilization of modeling architecture and reference models, not only to construct a generic ERP. The documentation and good understanding of the business processes and information flow, which were considered when defining the generic requirements and the original code generation, are essential to facilitate the definition of the enterprise's particular requirements and for the alteration of the relative codes.

5 – ENTERPRISE MODELING FOR THE ERP5 PROJECT

As highlighted, to properly develop and adaptation ERP, it is necessary enterprise modeling concepts, techniques and tools. In this section, some of them are briefly described, which can be used by the ERP5.

5.1 - MODELING ARCHITECTURE AND REFERENCE MODELS

A model is a representation of part of reality, from the point of view of the person wishing to use that model to understand, change, manage and control part of such reality (KOSANKE e NELL, 2000; SCHEER, 2000). Vernadat (1996) defines model as an abstraction of the reality expressed by some formalism defined by a modeling method based on the user's objective. Enterprise modeling is related to the following matters: what (it refers to the operations and objects processed by the enterprise), how (it refers to the manner how things are made), when (it provides a notion of time and is connected to the events representing changes in the state of the enterprise), how much (for example, the economic aspects), who (it refers to the human or machine resources or agents) and where (logistic aspects, for example).

Enterprise modeling allows not only for the better understanding of the enterprise's requirements which will interfere in the systems, but also for the identification of alternatives for the several processes of the enterprise, reducing efforts during the information system development and allowing for the enterprise analysis to have a better integration with the system development processes (KIRIKOVA, 2000; ODEH & KAMM, 2003; SHEN *et al*.,2004). The reference models can be developed in real or theoretical situations, and they document the know-how of a process that can be utilized by other users (SCHEER, 2000).

They can be applied to accumulate experience in some type of business or for business process solutions implemented and performed by a management information system, for example.

Modeling reference architectures and frameworks can be used to develop and organize enterprise models (MERTINS & JOCHEM, 2005; VERNADAT, 2002). For example, in the CIMOSA (Computer Integrated Manufacturing Open System Architecture) modeling framework two parts are considered (CIMOSA, 1996; KOSANKE, & ZELM, 1999): (i) a particular architecture and (ii) a reference architecture. Particular architecture is a set of models documenting the business environment. Reference architecture is used to help the business users in the construction process of their own particular architecture as a set of models describing the several aspects of the enterprise in different levels of modeling (model instantiation principle). The reference architecture is separated in two layers: a generic layer providing generic construction blocks (related to the modeling language) and a layer of partial models consisting of a library of classified and reusable partial models for some industrial sector, that is, models that can be adapted to enterprise's specific needs.

In addition to the models Instantiation Principle, the CIMOSA modeling framework has the Derivation and Generation principles.

The Derivation principle models the enterprise according to three successive modeling levels (iterations among these levels are, of course, allowed):

a) requirements definition, in order to express the business needs as realized by the users;

b) specification design, in order to build a formal, conceptual and executable model of the enterprise system;

c) description implementation, in order to document implantation details, installed resources, exception management mechanisms and to consider non-deterministic systems.

The Generation principle, which recommends the modeling of the enterprise according to four basic and complementary viewpoints (other views can be defined):

a) the view of function, which represents the functionality and the enterprise behavior (that is, events, activities and processes) including time aspects and exception management;

b) the view of information, which represents enterprise objects and its information elements;

c) the view of resources, which represents the enterprise means, capacities and management;

d) the view of enterprise, which represents organizational levels, authorities and responsibilities.

As described, modeling architectures and reference models aim to facilitate the work of modeling and to provide a common understanding about the enterprise's systems. A modeling language for the description of the models is needed. The next section describe the GERAM

5.2 - GERAM

GERAM (*Generalized Enterprise Reference Architecture and Methodology*) is a generalization of the GIM, PERA and CIMOSA architectures , which uses the best parts of these architectures to provide a reference to the enterprise integration area (IFIP-IFAC, 1999; NORAN, 2003). GERAM provides a description of all elements recommended in enterprise engineering and a collection of tools and methods to perform enterprise design and changes with success. It also considers enterprise models as an essential approach to support enterprise engineering and integration.

The most important component of GERAM is the *Generic Reference Architecture* (GERA), which defines a set of concepts to be used in enterprise engineering (enterprise entities, life cycle, life histories of enterprise entities, and others). GERAM defines too, as important components, enterprise engineering methodologies (EEM) and enterprise modeling languages

(EML) used by the methodologies to describe aspects of the enterprise, like the structure, content, and behavior of the entities to be modeled. The modeling language semantics may be defined by ontologies, metamodels and glossaries that are collectively called generic enterprise modeling concepts (GEMC). The modeling process (methodology) results in enterprise models (EM) that represent all or part of the enterprise operations (manufacturing or services tasks), its organization and management, and its control and information systems. These models can be transformed in an enterprise operational system (EOS), or be used to realize simulations, and to promote changes in the enterprise. To facilitate the modeling process, partial models (PEM), which are reusable models of human roles, processes and technologies, are used.

An EOS is a set of enterprise modules (EMO) that support the operational use of enterprise models. EMO provide prefabricated products like human skills, common business procedures or IT infrastructure services, used as components in the implementation of the EOS. Enterprise engineering tools (EET) support the methodologies and languages used for enterprise modeling.

GERAM also defines seven life-cycle phases (Figura 1) for any enterprise or any of its entities that are pertinent during the life of the entity. These phases, which can be subdivided further into several lower level activities, can be summarized as follows (IFIP-IFAC,1999):

- Identification: identifies the particular enterprise entity in terms of its domain and environment.
- Concept: conceptualizes an entity’s mission, vision, values, strategies, and objectives.
- Requirements: comprise a set of human, process, and technology oriented aspects and activities needed to describe the operational requirements of the enterprise.
- Design: models the enterprise entity and helps to understand the system functionalities.
- Implementation: the design is transformed into real components. After tested and approved

the system is released into operation.

- Operation: is the actual use of the system, and includes user feedback that can drive to a new entity life cycle.

- Decommission: represents the disposal of parts of the whole entity, after its successful use.

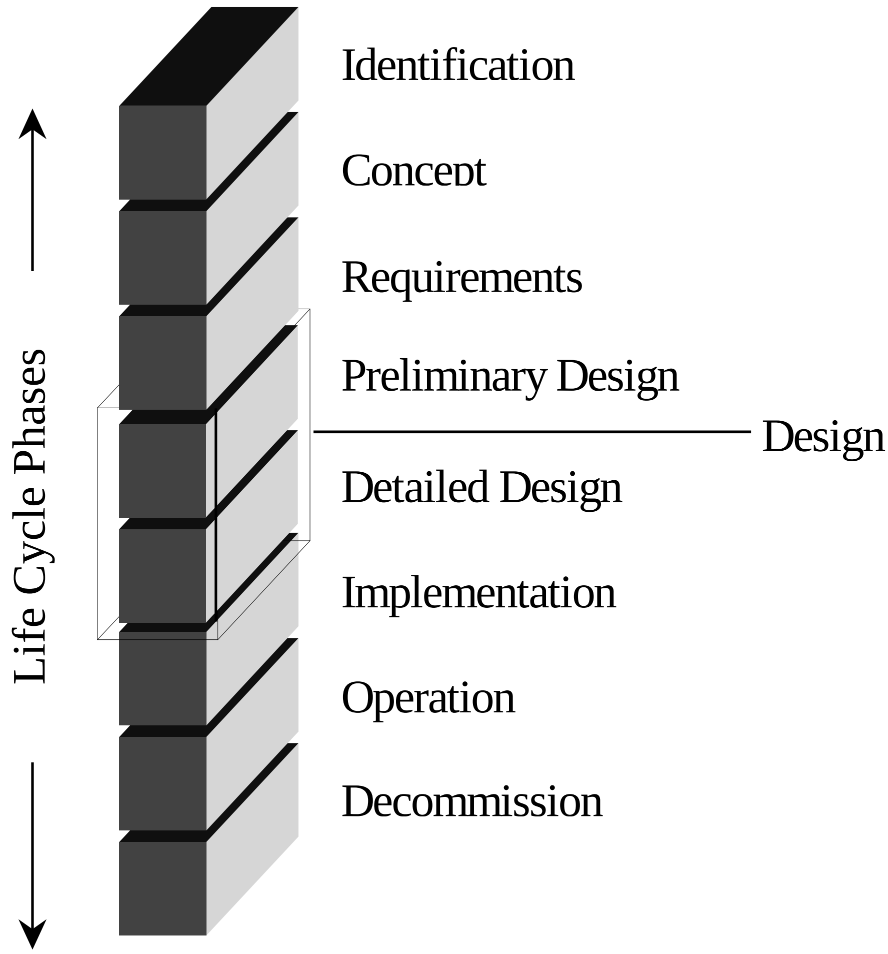

Figure 1 - GERAM Life Cycle Phases (IFIP-IFAC, 1999).

5.3 - UML LANGUAGE

The UML (Unified Modeling Language) is a graphical language for specification, design, visualization and documentation of a software system, as well for business modeling (BOOCH *et al.*, 2000). The UML 2.0 defines thirteen types of diagrams, divided into three categories, representing different aspects of interactions:

- Structure Diagrams include the Class Diagram, Object Diagram, Component Diagram, Composite Structure Diagram, Package Diagram, and Deployment Diagram.

- Behavior Diagrams include the Use Case Diagram (used by some methodologies during requirements gathering); Activity Diagram, and State Machine Diagram.

- Interaction Diagrams (derived from the more general Behavior Diagram) include the Sequence Diagram, Communication Diagram, Timing Diagram, and Interaction Overview Diagram.

The main objective of the UML is to define a visual and expressive modeling language, providing easy visualization, that is, the full understanding of the functions of a system from diagrams representing it. Also, in the management of the complexity, it allows for a simplified representation of the system activities, that is, each functional aspect is represented in specific models. Finally, in the communication, for example, the development team's communication is unified and facilitated through the diagrams.

5.4 – ERP5 ARCHITETURE COMPONENTS

As highlighted, enterprise modeling concepts, techniques and tools to properly develop and adapt FOS-ERP are necessary. This section briefly describes some architecture components defined to be used in the ERP5 project. The objective is defining the elements of ERP5 architecture based on GERAM (section 5.2).

For the enterprise reference architecture, the proposal is to use the reference architecture of GERAM (GERA), because it uses the best parts of others architectures to provide a complete set of methodological components to enterprise engineering.

The proposal to the development process or methodology (EEM) to the ERP5 is a process based on the Unified Process (UP), since it uses the best practices in software engineering JACOBSON, BOOCH & RUMBAUGH, 1999; KRUCHTEN, 2003). The process is been created including enterprise modeling activities to consider aspects, as businesses processes, resources and organizations of an enterprise. The activities of the process will be based on an interactive and incremental development process, considering the life cycle phases of GERA (CARVALHO & CAMPOS, 2006).

The UML is the modeling language (EML) in the project, because it is a *de facto* standard for modeling object oriented information systems. Adaptation with extensions of the language to enterprise modeling is been studied (ERIKSSON & PENKER, 2000). Some computer tool to support modeling (EET) with UML has been tested. It is being researched characteristics of possible tools, as free software, extensions support, and information changes with XMI (XML Metadata Interchange).

Besides the definition of methodological aspects for the ERP5 project, actually partial models to document a Production Planning and Control module are being currently developed with UML. Others partial models (PEM) will be create related to other enterprise areas or process.

Partial models will be utilized to the system documentation and, then, users will customize the partial models, utilized as reference to define the particular models (EM) according the requirements of the enterprise.

These facilitate the development and adaptation to particular codes of the particular ERP system (EOS).

6. CONCLUSIONS

The open source and free software might be an alternative to the SME in order to cut down expenses, for example, using open-source ERPs. Another advantage is the possibility of adaptation the software, allowing for the users to adjust processes or modules system to the reality of their enterprise by means of altering the open source, without being dependent on the developers or the owner of closed codes. However, it is somewhat difficult to adopt these ERPs, regarding the generation of these codes and the implantation of the systems in the enterprise. This matter is being dealt with on the ERP5 project. One of the proposals is the utilization of modeling architecture and reference models, since the documentation and good understanding of the business processes and information flow, which were considered when defining the requirements and the original code generation, are essential to facilitate the definition of the enterprise's particular requirements and for the alteration of the relative codes. This paper highlighted the need of enterprise modeling concepts, techniques and tools to properly develop and adapt FOS-ERP, and briefly described some of them, which can be used by the ERP5. The objective is defining methodological components of the ERP5 architecture based on GERAM. Some architecture components are defined, others not well defined, and, as GEMC, might be entirely defined.

REFERENCES

BOOCH, G.; RUMBAUGH, J. & JACOBSON, I. UML Guia do Usuário.Rio de Janeiro: Ed.Campus, 2000.

CAMPOS, R.; CARVALHO, R. A.; FERREIRA, A. S. *Modeling Architecture and Reference Models for the ERP5 Project*. In: First IFIP TC8 International Conference on Research and Practical Issues of Enterprise Information Systems, 2006, Viena.

CARVALHO, R. A.; CAMPOS, R. *A Development Process Proposal for the ERP5 System*. In: 2006 IEEE International Conference on Systems, Man, and Cybernetics, 2006, Taipei. Proceedings of the 2006 IEEE International Conference on Systems, Man, and Cybernetics, 2006.

CHUNG, S. H.; SNYDER, C. A. ERP Adoption: a technological evolution approach, International journal of Agile Management Systems. Vol. 2, n. 1, pg 24-32, 2000.

CIMOSA Association, *CIMOSA technical baseline*, CIMOSA Association, Stockholmerst 7, D-70731, Boblingen, Germany, 1996.

CORRÊA, H. L., GIANESI, I. G. N., CAON, M. Planejamento, Programação e Controle da Produção - MRPII/ERP: Conceitos, Uso e Implantação, São Paulo, Ed. Atlas, 2000.

DAVENPORT, T. H. Reengenharia de Processos: como inovar a empresa através da Tecnologia de Informação, Rio de Janeiro, Editora Campus, 1994.

ERIKSSON, H.; PENKER, M. Business Modeling with UML. Estados Unidos: Wiley & Sons, 2000. 459p.

IFIP – IFAC Task Force on Architectures for Enterprise Integration GERAM: Generalised Enterprise Reference Architecture and Methodology – Version 1.6.3, Março 1999.

JACOBSON, I., BOOCH, G., RUMBAUGH, J.. The Unified Software Development Process. Addison Wesley, 1999.

KALPIC, B. e BERNUS, P. *Business process modelling in industry - the powerful tool in enterprise management*, Computers in Industry, v. 47, p. 299-318, 2002.

KIRIKOVA, M. *Explanatory capability of enterprise models,* Data & Knowledge Engineering, v. 33, p.119-136, 2000.

KOSANKE, K.; ZELM, M. CIMOSA modelling processes. Computers in Industry – Volume 40, Pages 141-153, 1999.

KOSANKE, K., NELL, J. G. Standardisation in ISO for Enterprise Engineering and Integration, Computers in Industry, v.40, p. 311-319, 2002.

KRUCHTEN, P. Introdução ao RUP – Rational Unified Process, Rio de Janeiro: Editora Ciência Moderna Ltda., 2003.

MERTINS, K. & JOCHEM R. Architectures, methods and tools for enterprise engineering, International Journal of Production Economics, V. 98, N. 2, P. 179-188, 2005.

NORAN, O. An analysis of the Zachman framework for enterprise architecture from Geram perspective, Annual Reviews in Control, v. 27, p. 163-183, 2003.

ODEH, M.; KAMM, R. Bridiging the gap between business models and system models, Information and Software Technology, v. 45, p. 1053-1060, 2003.

PRESSMAN, R. S. Engenharia de software. 5 Ed. Rio de Janeiro: McGraw-Hill, 2002.

SCHEER, A. W. *Enterprise-Wide Data Modeling - Information System in Industry*, Springer-Verlag, 1989.

SCHEER, A. ARIS – Business Process Modeling. 3[a.] ed. Berlim: Springer-Verlag, 2000.

SHEN, H.; WALL, B.; ZAREMBA, M.; CHEN, Y.; BROWNE J. Integration of business modelling methods for enterprise information system analysis and user requirements gathering. Computers in Industry, v.54, n. 3, p. 307-323, 2004.

SMETS-SOLANES, J., CARVALHO, R. A. An Abstract Model for An Open Source ERP System: The ERP5 Proposal. In: VIII International Conference on Industrial Engineering and Operations Management, Curitiba, 2002.

SMETS-SOLANES, J.P., CARVALHO, R. A. ERP5: A Next-Generation, Open-Source ERP Architecture. IEEE IT Professional. , v.5, n.4, p.38 - 44, 2003.

VERNADAT F. B. *Enterprise modeling and integration, Principles and Applications,* Chapman & Hall, 1996.

VERNADAT, F. B. Enterprise Modeling and Integration (EMI): current status and research perspectives, Annual Reviews in Control, v. 26, pg 15-25, 2002.