

Modeling Architectures and Reference Models: Development and Maintenance Open Source ERP

Tulio Cremonini Entringer¹, Ailton da Silva Ferreira², Denise Cristina de Oliveira Nascimento³, Jorge Luiz Lourenço das Flores⁴, Ângelo Mário do Prado Pessanha⁵, Paulo Mauricio Tavares Siqueira⁶

¹Universidade Estadual do Norte Fluminense Darcy Ribeiro (UENF), Campos dos Goytacazes, Rio de Janeiro, Brazil
Email: tulio_entringer@hotmail.com

²Universidade Federal Fluminense (UFF), Macaé, Rio de Janeiro, Brazil
Email: ailtonsilvaferreira@yahoo.com.br

³Universidade Federal Fluminense (UFF), Macaé, Rio de Janeiro, Brazil
Email: denise_cristin@yahoo.com.br

⁴Universidade Federal Fluminense (UFF), Macaé, Rio de Janeiro, Brazil
Email: jorgeflores@id.uff.br

⁵Universidade Federal Fluminense (UFF), Macaé, Rio de Janeiro, Brazil
Email: angeloprado@uol.com.br

⁶Universidade Federal Fluminense (UFF), Niterói, Rio de Janeiro, Brazil
Email: paulomauricio@id.uff.br

Abstract — *The adoption Enterprise Resource Planning (ERPs) by small and medium-sized businesses may not possible its cost. At same time, whenever adapt ERP specific needs company, user becomes dependent developers due to the lack access and knowledge respective code. Free and open source software can promote advantages companies, however, for their adoption, it is necessary to develop techniques tools facilitate implementation and maintenance code. This article highlights the importance of defining modeling architectures and reference models for development and maintenance open source ERPs, especially the ERP5 project.*

Keywords — *ERP. open code. business modeling.*

I. INTRODUCTION

Organizations today must vigilant keep up market developments increasingly competitive environment. Options for companies plan their resources a better planning of processes is implementation Management Information Systems, also called ERPs (Enterprise Resource Planning), which can material and human resources company.

However, the price of such systems may be a deterrent to small and medium-sized businesses wishing to obtain this feature. Also, the adequacy of ERP modules according to each organization's characteristics may become important

for their competitiveness, but closed systems make companies dependent on the payment of these services to the system's proprietary developers for adaptations.

Open free software can alternative for small and medium-sized businesses reduce costs, example by using open source ERPs. Another advantage is the possibility of adapting the software, allowing users to adjust processes or modules of the system to the reality of their organization by changing the open source, without being dependent on proprietary developers of closed codes.

However, there are difficulties for the adoption in practice of these ERPs related to the generation of these codes and the implantation of the systems in the company. These difficulties have been addressed in the ERP5 project (Carvalho, 2003). One of the proposals is the use of a modeling architecture and reference models, since the documentation and the good understanding of business processes and the flow of information, which were considered when defining requirements and generating original codes, are essential facilitate definition requirement an enterprise to change relative codes.

This article aims to highlight the need to define modeling architectures and reference models to facilitate the change of open source ERP codes. Thus, after this introduction, a brief evolution of production management support systems and the ERP5 project is presented. The following

are some comments about software engineering, modeling architecture, reference models for companies and the UML language. Finally, it is presented Aggregate Planning modeling shows a prototype generated from the UML modeling, followed by the final considerations.

II. EVOLUTION COMPUTER SYSTEMS PRODUCTION MANAGEMENT

MRP (Material Requirements Planning), Also called MRP I, proposed Joe Orlicky in the early 60's and came up the purpose of computationally executing materials planning activities, this system being the flow of base material (GOULART, 2000).

Nevertheless, in the 1970s, this system evolved alongside the development of information technology. A computational system emerged with a broader scope, performing the main activities of production planning and control, and was renamed MRPII (Manufacturing Resources Planning).

MRP II is considered system which decision making highly centralized and based basic principle that all programs established the system fulfilled as faithfully as possible, becoming less flexible to the variation of work by the labor (CORRÊA et al., 2000).

For Goulart (2000) MRP II be a hierarchical management system, since the long-term plans are level of successive detailing, this system can reach specific components and machines.

In the United States from 1990 onwards and in Brazil after 1996 emerged ERPs (Enterprise Resource Planing) with the main purpose of integrating several areas of the company. According to Heizer and Renzer (2001) MRP II systems that connect customers to suppliers are called ERPs. When we implement an ERP, rather than putting a new program on the company's computers, you are defining or adopting a work methodology, a workflow.

III. ENTERPRISE RESOURCE PLANING 5

The case of Compiere (www.compiere.com.br) and the ERP5 project (www.erp5.org). The latter is a free-code ERP project that aims to offer a high-tech, low-cost solution for SMEs. The ERP 5 System is currently developed by a group of companies and educational and research institutions from France and Brazil. This system uses the Zope platform and is totally object-based, workflow and Web technologies.

According to Carvalho (2003) it has five innovative technologies:

- Multi-system is multi-user, multi-organization, multi-language, multi-currency, multi-cost and multi-scenario;
- Meta- provides several levels of detail for the same management process;

- Distributed - uses advanced synchronization mechanisms that allow the distribution and sharing of data without the need for permanent connection to the network;
- Object-based - the use of a set of objects allows modeling and implementation of complex decision support systems;
- Free - all information generated, technologies and methodologies developed, are freely available from the project website.

According to Solanes and Carvalho (2003), the ERP5 architecture incorporates advanced concepts such as object-oriented database and content management system, synchronization of data between different installations, and a clear method modeling method and, consequently, of source code generation.

ERP5 defines an abstract business management model, and this model is based on five classes (SOLANES and CARVALHO, 2003):

- Resource: describes an abstract resource in a business process (such as individual skills, products, machines, etc.). Relationships between nodes define bundles of materials as well as prototypes.
- Node: can receive and send resources. They may be relative to physical entities (such a manufacturing facility) or abstract entities (such a bank account). Metanodes contain other nodes, such as companies.
- Movement: describes a movement of resources between given moment for a given duration. Example a move might send raw material from stock to factory.
- Path: describes way that node accesses features that needs. They are abstract, being used planning.
- Item: physical instance asset.

ERP5 is based on a template that can link anything to a category. Some examples include a category of resources (such as services, raw materials, skill or money) or a category of organizations (such as a group of companies, a group of people or a retail chain) (SOLANES and CARVALHO, 2003).

For the development of a good Information System, as well as for the development of ERP 5, it is necessary to use adequate techniques of Software Engineering.

IV. ENGINEERING SOFTWARE AND ANALYSIS REQUIREMENTS

According Azevedo (2003), first definition software engineering proposed Fritz Bauer establishment and use of solid engineering principles that software can be obtained economically is reliable works efficiently in real machines.

For Pressman (2003) software engineering encompasses three fundamental elements: methods, tools and

procedures. Independent software development model, development process contains three generic phases:

1. Definition Phase - where the software developer tries to identify what information needs processed, functions and performance are desired, interfaces should establish, what design constraints, and what validation criteria required to define successful system. In definition phase, the methods applied vary according function model, but there are three specific steps: (i) system analysis, which defines the role of each element in a computer-based system; (ii) software project planning, which addresses risk analysis, cost estimates and task definition and work scheduling; (iii) requirements analysis, which details the scope through an analysis of the information domain and software functions.
2. Development Phase - defines how the data structure and software architecture have to be designed, the project will be translated into a programming language and how the tests have to be performed.
3. Maintenance phase - which focuses changes that associated error correction, adaptations functional improvement software.

The analysis of requirements as already mentioned is a step that is always present in the software definition phase, being formed by a set of techniques used to collect, detail, document and validate the requirements of a software product (PETERS & PEDRYCZ, 2001).

For a long time there has been a concern the analysis of software requirements, leaving in background the proper analysis business requirements, does not contribute an effective link between the needs of business process computerization and software design.

Analyzing and documenting business and software requirements through templates is essential for the development of open source ERPs, making appropriate techniques and tools necessary. In this sense, a modeling architecture that adequately contemplates the modeling of all aspects of business processes, including other aspects related to software development, can facilitate reuse, better functionality, better performance, system comprehensibility, resulting in savings of effort and resources.

V. MODELING ARCHITECTURE AND REFERENCE MODELS

According to Pidd (1998), a model is a representation of part of the reality seen by the person who wants to use that model to understand, change, manage and control part of that reality. Vernadat (1996) defines model as an abstraction of reality expressed by some formalism defined by a modeling method in function of the objective of the user. Enterprise modeling is related to the following issues: what (refers to the operations and objects processed by the company), how (refers to the way things are done), when (provides a notion of time and is associated with (for example, economic aspects), who (refers to resources or agents) and where (logistic aspects, for example).

Organizational modeling allows not only better understanding organizational requirements that will interfere with systems, but also identify alternatives to the various processes of the organization, facilitating efforts during the development of the information system and allowing organizational analysis to be better integrated into processes of system development (PÁDUA et al, 2002).

For Scheer (1998) reference models can be developed in real or theoretical situations and document the know - how of a process that can be used by others.

For Keller & Teuffel (1998) reference models can be applied to accumulate experience in a business type or to business process solutions implemented and executed in business management software.

The CIMOSA (Computer Integrated Manufacturing Open System Architecture) model considers two parts (VERNADAT, 1996): (i) a architecture and (ii) a reference architecture (Figure 1). Private architecture is a set of templates documenting the business environment. Reference architecture is used to assist business users in the process of constructing their own architecture as a set of models describing the various aspects of the enterprise at different levels of modeling (Figure 2). The reference architecture is separated into two layers: a generic layer providing generic building blocks (relative to the modeling language) and a layer of partial models consisting of a library of partial models classified and reusable for some industry sector, or models that can be adapted to the specific needs of the company.

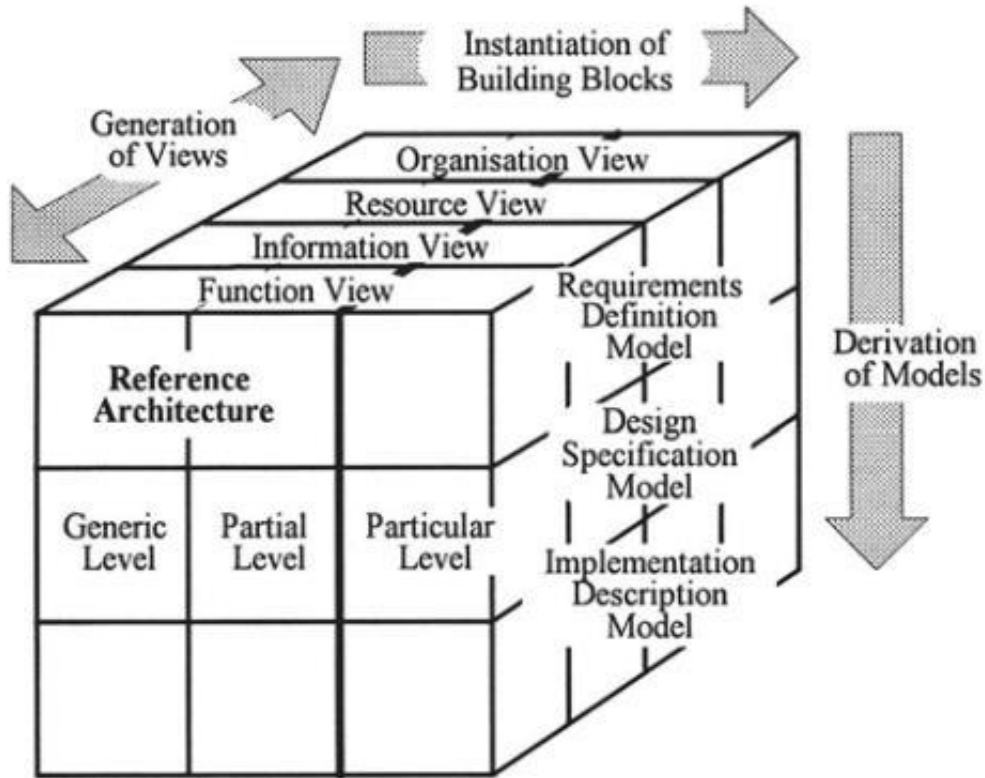


Fig. 1: CIMOSA modeling framework (adapted from VERNADAT, 1996).

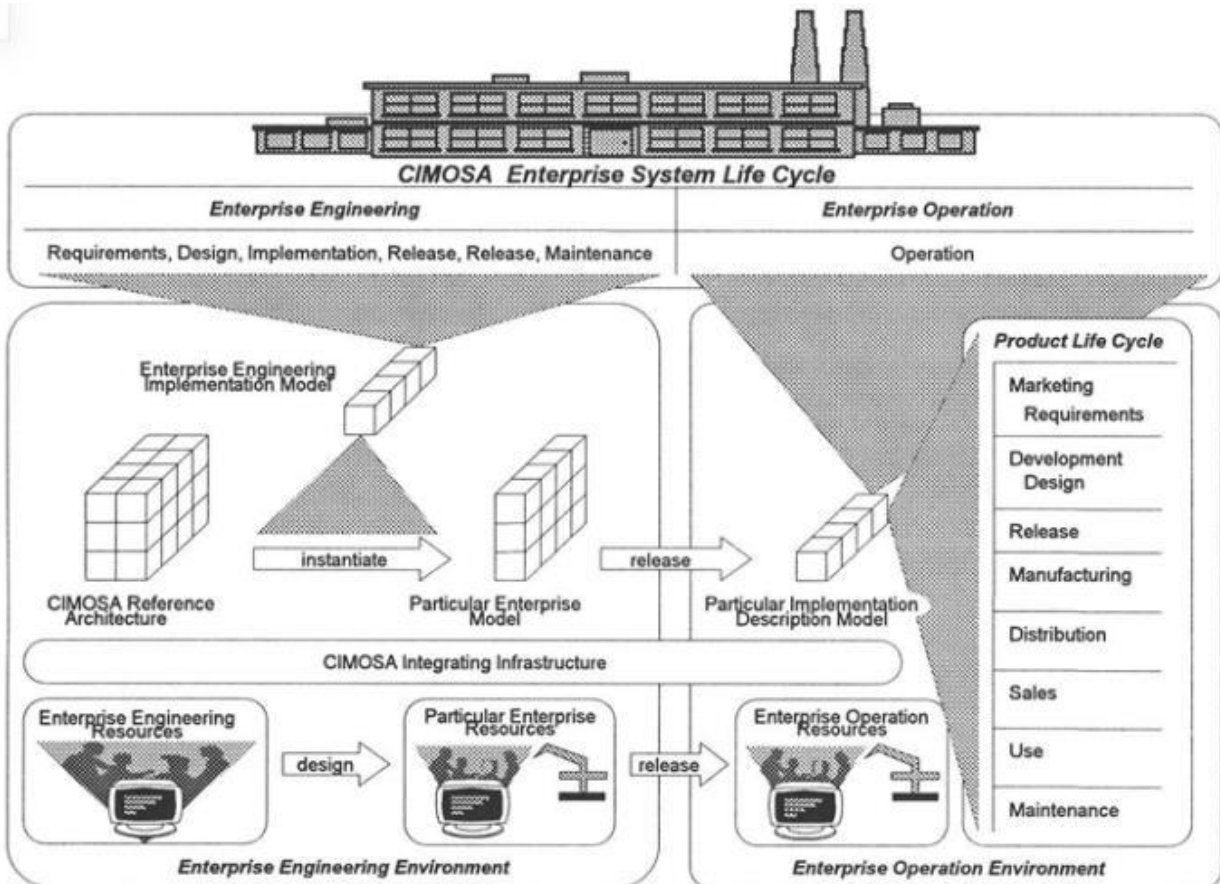


Fig. 2: CIMOSA Architectural Structure (adapted from VERNADAT, 1996).

In addition to this principle of Particularization of models (from reference models), the CIMOSA modeling structure has the principles of Derivation and Model Generation.

The Derivation principle model firms according to three successive modeling levels (iterations between these levels are, of course, allowed):

- a) definition of requirements to express the needs of the business as perceived by users;
- b) project specification to build a formal, conceptual and executable system model of the company (time is considered);
- c) description of the implementation to document implementation details, installed features, exception management mechanisms, and consider non-deterministic systems.

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

- a) the function view that represents the functionality and behavior of the company (ie, events, activities and processes) including time aspects and exception management;
- b) the information view, which represents objects of the company and its information elements;
- c) the resource view, which represents the company's means, its capabilities and management;
- d) the organizational view, which represents organizational levels, authorities, and responsibilities.

As already described, modeling architectures and reference models aim to facilitate modeling work and provide a common understanding of enterprise systems.

For the description of the models a modeling language is necessary.

VI. UNIFIED MODELING LANGUAGE (UML)

UML is a graphical language for specification, construction, visualization and documentation of a software system (BOOCH, et al., 2000).

The UML used the State Diagram, Class Diagram, Object Diagram (from which the Collaboration Diagram appeared), the Process Diagram (giving the Implementation Diagram) and the Module Diagram (resulting in the Component Diagram). The Fusion method also had its collaboration with the Object Interaction Graph. And Harel's statecharts contributed to the creation of the Activity Diagram (LARMAN, 2000).

The UML diagrams (LARMAN, 2000 and FURLAN, 1998) are described below:

It can be said that the main objective of the UML is to define a visual and expressive modeling language, in the sense of providing facilities in the visualization, that is, the full understanding of the functions of a system from diagrams that represent it, in the management of complexity, allowing a simplified representation of the activities of the system, that is, that each functional aspect of it is represented in specific models and finally in the communication, unifying the communication of the development team in the form of diagrams. The modeling of aggregate planning that is defined by Heizer Render (2001) as an elaborated activity among the commercial sector, production sector, purchasing and management of the company is shown below.

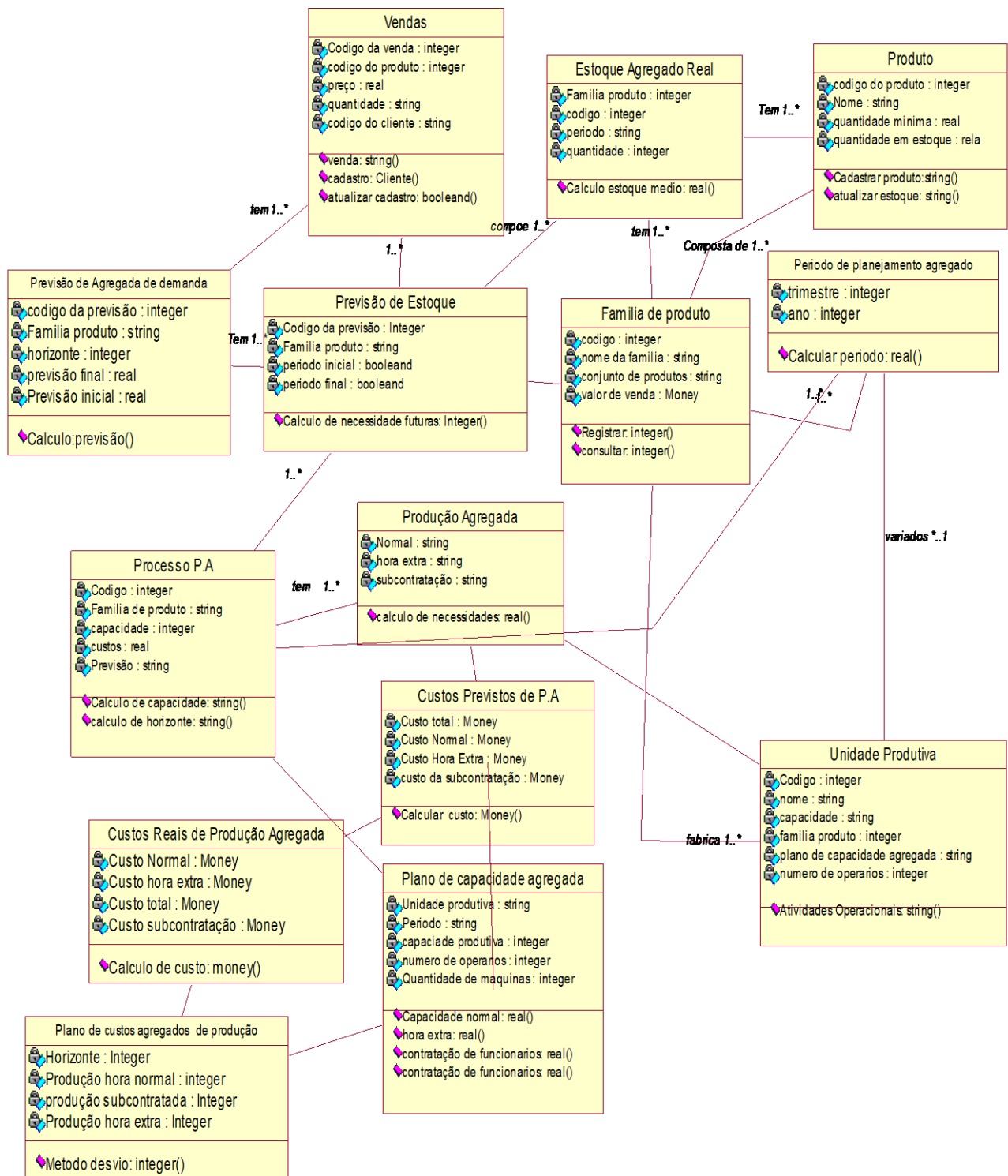


Fig. 3: Modeling Aggregate Planning UML.

With respect to the UML modeling, it was possible to generate code generating a prototype in Delphi with the

following characteristics.

Fig. 4: Diagram Class Proposed Aggregate Planning ERP Model 5

VII. FINAL CONSIDERATIONS

An ERP system can help companies quest for competitiveness, adoption is hampered due to their cost of purchase dependency supplier company for possible adaptations system due lack access and knowledge changes code. Free and open source software, such ERP systems, advantageous alternative, adoption practice necessary to develop and use techniques and tools that facilitate implementation modification this software.

A modeling architecture and reference models is essential enable development, deployment and change open source ERPs. For definition a modeling architecture RP5 project study possibility using CIMOSA modeling framework concepts and architecture proposed Eriksson & Penker (2000).

Reference models for ERP5 system modules should be generated in order to "map" and document (ie model) generic processes and information, which can serve basis adaptations (or particularizations) through new codes. In ERP5 Project UML language adopted, which became facto standard, worldwide accepted and used, which facilitates diffusion models and codes. Currently works module Planning, Programming and Production Control.

REFERENCES

- [1] AZEVEDO, D. P. J. (2003) - Application of the Business Modeling Technique with UML to Iterative processes of software development. Master's thesis in Production Engineering -UENF.Campos.
- [2] BOOCH, G.; RUMBAUGH, J. & JACOBSON, I. (2000) - UML User's Guide. Translation of Fábio Freitas da Silva - Rio de Janeiro: Campus.
- [3] CARVALHO, R. de A. (2003) - Development of ERP System Advanced and Open Code for Small and Medium Enterprises. Design assigned to CNPq. CEFET Fields.
- [4] COLANGELO, L. F. (2001) - Deployment of ERP Systems: A Long-Term Approach. São Paulo: Atlas.
- [5] CORRÊA, H. L.; GIANESI, I. G.N. & CANON M. (2000) - Planning, programming and production control: MRP / ERP: concepts, use and implantation. 3.ed. São Paulo: Gianesi Correa & Associados: Atlas.FURLAN, J. D. (1988) - Modelagem de Objetos através da UML-the Unified Modeling Language. São Paulo: Makon Brooks.

- [6] Jaiswal, M. (2018). Software Quality Testing. International Journal Of Informative & Futuristic Research, 6(2), 114-119.
- [7] GOULART, C. P. (2000) - Proposal of a reference model for Planning and Control of Production in Virtual Companies. Master's Dissertation School of Engineering of São Carlos, University of São Paulo. São Paulo.
- [8] HEIZER, Jay; RENDER, Barry (2001). Administration of Operations - Goods and Services. Trad. Dalton Conde de Alencar-5 ed. São Paulo: LTC.
- [9] KELLER, G. & TEUFEL, T. S. (1998) - Process Oriented Implementation. Harlow, Addison-Wesley.
- [10] LARMAN, C. (2000) - Using UML and Patterns: An Introduction to Object Oriented Analysis and Design. Porto Alegre: Bookman.
- [11] PÁDUA, S. I. D. & CAZARINI, E. W. (2002) - Organizational Modeling to capture organizational requirements. USP- EESC- School of Engineering of São Carlos. Annals, SIMPEPVIII.
- [12] PIDD, M. (1998) - Business Modeling, tools for decision making; trad Gustavo Severo de Borba et al.-Porto Alegre: Artes Medicas.
- [13] SCHEER, A.W. (1998) - Aris - Business process Frameworks. Berlin, Springer Verlag.
- [14] SOLANES, J. P. S. & CARVALHO, R. A. (2003) - An Abstract Model For an open source Erp system: The Erp5 proposal. Rio de Janeiro. Enegep.
- [15] VERNADAT F.B. (1996) - Enterprise Modeling and Integration, Principles and Applications, Chapman and Hall. 1996.
- [16] PETERS, James & PEDRYCZ, Witold (2001) - Software Engineering: Theory and Practice. Translation by Ana Patrícia Garcia. Ed. Campus. Rio de Janeiro.
- [17] PRESSMAN, Roger. Software Engineering. Translation José Carlos Barbosa dos Santos. São Paulo: Makon Brooks, 1999.