

MODELAGEM E ARQUITETURAS DE UM PROJETO DO ERP5

Ailton da Silva Ferreira (UENF)

Mestre em Engenharia de Produção (UENF)
Pós-Graduado em Tecnologia de Petróleo (UNESA)
ailton@unef.br

Renato Campos (UENF)

Doutor em Engenharia de Mecânica (UNICAMP)

Rogério Atem de Carvalho (CEFET-Campos)

Doutor em Engenharia de Produção (UENF)

Denise Cristina de Oliveira (UENF)

Mestre em Engenharia de Produção (UENF)

RESUMO

A adoção de sistemas ERPs por pequenas e médias empresas pode não ser possível devido ao seu custo. Ao mesmo tempo, sempre que adaptar o ERP para as necessidades específicas da empresa, o usuário fica dependente dos desenvolvedores devido à falta de acesso e conhecimento do respectivo código. Softwares livres e de código aberto podem promover vantagens para as empresas, no entanto, para a sua adoção, é necessário o desenvolvimento de técnicas e ferramentas que facilitem a sua implantação e manutenção código. Este artigo ressalta a importância de definir arquiteturas de modelagem e modelos de referência para o desenvolvimento e manutenção de ERPs de código aberto, em especial o projeto ERP5.

Palavras chave: ERP, código aberto, modelagem de empresas.

ABSTRACT

The adoption of ERPs systems by small and middle-sized companies may not be possible due to their cost. At the same time, when adapting ERP to the company's particular needs, the user keeps depending on the system's sellers 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 deployment and code maintenance. This article emphasizes the importance of defining modeling architectures and reference models for the development and maintenance of open-source ERPs, in special the ERP5 project.

INTRODUÇÃO

As organizações atualmente devem ficar atentas para acompanhar os avanços do mercado, num cenário cada vez mais competitivo. Uma das opções para que as empresas possam programar seus recursos e obter um maior planejamento de seus processos é a implementação de Sistemas de Informações Gerenciais, também chamados ERPs (*Enterprise Resource Planning*), que podem auxiliar o planejamento dos recursos materiais e humanos da empresa. Porém, o preço desses sistemas pode ser um empecilho a pequenas e médias empresas que desejem obter esse recurso. Também, a adequação dos módulos dos ERPs conforme as características de cada organização pode se tornar importante para a sua competitividade, mas os sistemas fechados fazem com que as empresas dependam do pagamento desses serviços aos desenvolvedores proprietários do sistema para que sejam feitas adaptações. O software livre e aberto pode ser uma alternativa para que pequenas e médias empresas possam diminuir custos, por exemplo, utilizando-se de ERPs de código aberto. Outra vantagem é a possibilidade de adaptação do software, permitindo aos próprios usuários ajustarem processos ou módulos do sistema à realidade de sua organização através da alteração do código aberto, sem ficarem dependentes aos desenvolvedores proprietários de códigos fechados. No entanto, existem algumas dificuldades para a adoção na prática desses ERPs relacionadas com a geração desses códigos e a implantação dos sistemas na empresa. Essas dificuldades vêm sendo tratadas no projeto ERP5 (Carvalho, 2003), sendo que uma das propostas é a utilização de uma arquitetura de modelagem e modelos de referência, já que a documentação e o bom entendimento dos processos de negócios e do fluxo de informações, os quais foram considerados quando da definição de requisitos e geração dos códigos originais, são essenciais para facilitar a definição de requisitos particulares de uma empresa e para a alteração dos códigos relativos.

Este artigo tem como objetivo ressaltar a necessidade da definição de arquiteturas de modelagem e modelos de referência para facilitar alteração de códigos de ERPs de código aberto. Assim, após esta introdução, são apresentados uma breve evolução dos sistemas de suporte a gestão da produção e o projeto ERP5. A seguir são tecidos alguns comentários sobre engenharia de software, arquitetura de modelagem, modelos de referência para empresas e a linguagem UML. Finalmente, é apresentada modelagem do Planejamento Agregado mostra um protótipo gerado a partir da modelagem em UML, seguida das considerações finais.

2. A evolução dos sistemas computacionais para gestão da produção

O MRP (Material Requirements Planning), também chamado de MRP I, foi proposto por Joe Orlicky no começo dos anos 60 e surgiu com o objetivo de executar computacionalmente as atividades de planejamento dos materiais, sendo que este sistema é limitado ao tratamento do fluxo de materiais (GOULART, 2000).

Não obstante na década de 70 esse sistema evoluiu paralelamente com o desenvolvimento da informática, surgindo um sistema computacional com objetivos mais abrangentes realizando as principais atividades do planejamento e controle da produção e passando a se chamar MRPII (Manufacturing Resources Planning). O MRP II é considerado um sistema no qual a tomada de decisão é bastante centralizada, e é baseado no princípio básico de que todos os programas estabelecidos pelo sistema serão cumpridos da forma mais fiel possível, se tornando pouco flexível à variação do trabalho por parte da mão de obra (CORRÊA et al., 2000). Para Goulart (2000) o MRP II pode ser visto como um sistema hierárquico de gestão, pois os planos de longo prazo são de um nível de detalhamento sucessivo, sendo que este sistema pode chegar ao nível de componentes e máquinas específicas.

Nos Estados Unidos a partir de 1990 e no Brasil após 1996 surgiram os ERPs (Enterprise Resource Planing) com o principal propósito de integrar várias áreas da empresa. Segundo Heizer e Renzer (2001) os sistemas MRP II que interligam clientes a fornecedores são denominados ERP. Quando implantamos um ERP, mais do que colocar um novo programa nos computadores da empresa você está definindo ou adotando uma metodologia de trabalho, um *workflow* (fluxo de trabalho).

3. ERP5

Atualmente existem algumas propostas de ERPs livres e que permitem a alteração de seus códigos, como o caso da Compiere (www.compiere.com.br) e do projeto ERP5 (www.erp5.org). Este último é um projeto de ERP de código livre que visa oferecer uma solução de alta tecnologia e baixo custo para PMEs. O Sistema ERP 5 é desenvolvido atualmente por um grupo de empresas e instituições de ensino e pesquisa da França e Brasil. Este sistema utiliza a plataforma Zope e é totalmente baseado em objetos, *workflow* e tecnologias Web. Segundo Carvalho (2003) possui cinco tecnologias inovadoras:

- Multi- o sistema é multi-usuário, multi-organização, multi-linguagem, multi-moeda, multi-custo e multi-cenário;
- Meta- oferece vários níveis de detalhes para um mesmo processo de gestão;
- Distribuído- utiliza mecanismos de sincronização avançados que permite a distribuição e compartilhamento de dados sem a necessidade de conexão permanente com a rede;
- Baseado em objetos - o emprego de um conjunto de objetos permite modelar e implementar sistemas complexos de suporte a decisão;
- Livre- toda a informação gerada, tecnologias e metodologias desenvolvidas, são livremente disponibilizadas pelo *site* do projeto.

A arquitetura do ERP5 segundo Solanes e Carvalho (2003) incorpora desde sua concepção, conceitos avançados como banco de dados orientados a objetos e sistema de gestão de conteúdo, sincronização de dados entre diferentes instalações, possuindo ainda um método claro de modelagem de processos e conseqüentemente de geração de código fonte.

O ERP5 define um modelo abstrato de gerenciamento de negócios, sendo que este modelo se baseia em cinco classes (SOLANES e CARVALHO, 2003):

- Resource: descreve um recurso abstrato em um processo de negócio (como habilidades individuais, produtos, máquinas etc). Relações entre nós (nodes) definem as listas de materiais bem como protótipos .
- Node: podem receber e enviar recursos. Podem ser relativos a entidades físicas (como uma instalação fabril) ou abstratas (como uma conta bancária). Metanodes são nós que contêm outros nós, como empresas.
- Movement: descreve um movimento de recursos entre nós, em um dado instante e por uma dada duração. Por exemplo, um movimento pode ser o envio de matéria prima do estoque para a fábrica.
- Path: descreve uma forma que um nó acessa recursos dos quais precisa. São abstratos, sendo utilizados para planejamento.
- Item: instância física de um recurso.

O ERP5 é baseado em um modelo que pode associar qualquer coisa a uma categoria. Alguns exemplos incluem uma categoria de recursos (tais como serviços, matéria-prima, habilidade ou dinheiro) ou uma categoria de organizações (tais como um grupo de empresas, um grupo de pessoas ou uma cadeia de varejo) (SOLANES e CARVALHO, 2003).

Para o desenvolvimento de um bom Sistema de Informação, assim como para o desenvolvimento do ERP 5, é necessário utilizar adequadas técnicas de Engenharia de Software.

4. Engenharia de software e análise de requisitos

Segundo Azevedo (2003) uma primeira definição de engenharia de software foi proposta por Fritz Bauer como sendo o estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais.

Para Pressman (2003) a engenharia de software abrange um conjunto de três elementos fundamentais: métodos, ferramentas e procedimentos. Independente do modelo de desenvolvimento de software, o processo de desenvolvimento contém três fases genéricas:

1. Fase de definição - onde o desenvolvedor de software tenta identificar que informações necessitam ser processadas, quais funções e desempenho são desejados, quais interfaces devem ser estabelecidas, quais restrições de projeto e quais critérios de validação são exigidos para se definir um sistema bem sucedido. Na fase de definição os métodos aplicados variam de acordo com a função do modelo, porém existem três etapas específicas: (i) análise do sistema, que define o papel de cada elemento num sistema baseado em computador; (ii) planejamento do projeto de software, que aborda a análise de riscos, estimativas de custos e a definição de tarefas e programação de trabalho; (iii) análise de requisitos, que detalha o escopo através de uma análise do domínio da informação e das funções de software.
2. Fase de desenvolvimento - defini como a estrutura de dados e a arquitetura de software têm de ser projetadas, ou seja como o projeto será traduzido numa linguagem de programação e como os testes têm de ser realizados.
3. Fase de manutenção – que se concentra nas mudanças que estão associadas à correção de erros, adaptações e melhoramento funcional do software.

A análise de requisitos como já citada é uma etapa sempre presente na fase de definição do software, sendo formada por um conjunto de técnicas empregadas para levantar, detalhar, documentar, e validar os requisitos de um produto de software (PETERS & PEDRYCZ, 2001). Durante muito tempo houve uma preocupação apenas com a análise de requisitos de software, deixando em segundo plano a adequada análise dos requisitos dos negócios, o que não contribui uma ligação eficaz entre as necessidades de informatização de processos de negócios e o projeto de software. A análise e documentação dos requisitos de negócios e de software através de modelos são essenciais para o desenvolvimento de ERPs de código aberto, tornando necessárias adequadas técnicas e ferramentas. Nesse sentido uma arquitetura de modelagem que contemple de forma adequada a modelagem de todos os aspectos dos processos de negócios, incluindo os demais aspectos relativos ao desenvolvimento de software, pode facilitar o reuso, a melhor funcionalidade, o melhor desempenho, a compreensibilidade do sistema, resultando em economias de esforço e recursos.

5 Arquitetura de modelagem e modelos de referência

Segundo Pidd (1998), um modelo é uma representação de parte da realidade vista pela pessoa que deseja usar aquele modelo para entender, mudar, gerenciar e controlar parte daquela realidade. Vernadat (1996) define modelo como uma abstração da realidade expressa por algum formalismo definido por um método de modelagem em função do objetivo do usuário. A modelagem de empresas está relacionada às seguintes questões: o que (refere-se as operações e objetos processados pela empresa), como (refere-se a maneira como as coisas são feitas), quando (fornece uma noção de tempo e está associado aos eventos representando mudanças no estado da empresa), quanto (por exemplo aos aspectos econômicos), quem (refere-se aos recursos ou agentes) e onde (aspectos logísticos, por exemplo).

A Modelagem da Organização permite não só melhor entender requisitos organizacionais que irão interferir nos sistemas, mas também identificar alternativas para os vários processos da organização, facilitando os esforços durante o desenvolvimento do sistema de informação e permitindo que a análise organizacional seja mais bem integrada aos processos de desenvolvimento do sistema (PÁDUA et al, 2002). Para Scheer (1998) os modelos de referência podem ser desenvolvidos em situações reais ou teóricas e documentam o *know – how* de um processo que pode ser utilizado por outros. Para Keller & Teuffel (1998) os modelos de referência podem ser aplicados para acumular a experiência em um tipo de negócio ou para soluções de processos de negócios implementadas e executadas em software de gestão empresariais.

A estrutura de modelagem CIMOSA (*Computer Integrated Manufacturing Open System Architecture*) considerada duas partes (VERNADAT, 1996): (i) uma arquitetura particular e (ii) uma arquitetura de referência (Figura 1). Arquitetura particular é um conjunto de modelos documentando o ambiente empresarial. Arquitetura de referência é usada para ajudar os usuários de negócios no processo de construção de sua própria arquitetura particular como um conjunto de modelos descrevendo os vários aspectos da empresa em diferentes níveis de modelagem (Figura 2). A arquitetura de referência é

separada em duas camadas: uma camada genérica proporcionando blocos de construção genéricos (relativa à linguagem de modelagem) e uma camada de modelos parciais consistindo de uma biblioteca de modelos parciais classificados e re-usáveis para algum setor da indústria, ou seja, modelos que podem ser adaptados às necessidades específicas da empresa.

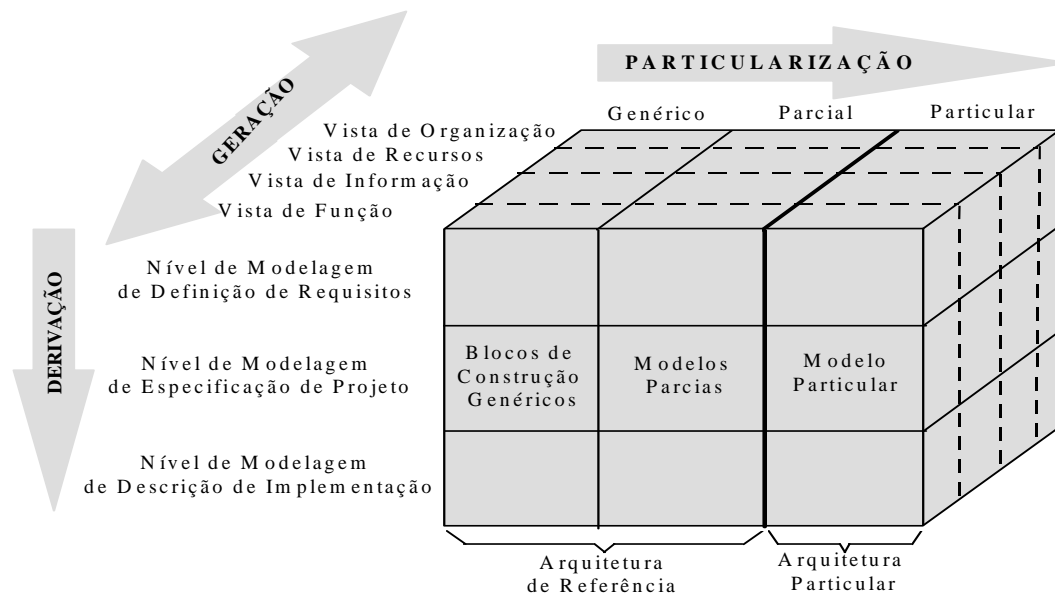


Figura 1 - Estrutura de modelagem CIMOSA (adaptado de VERNADAT, 1996).

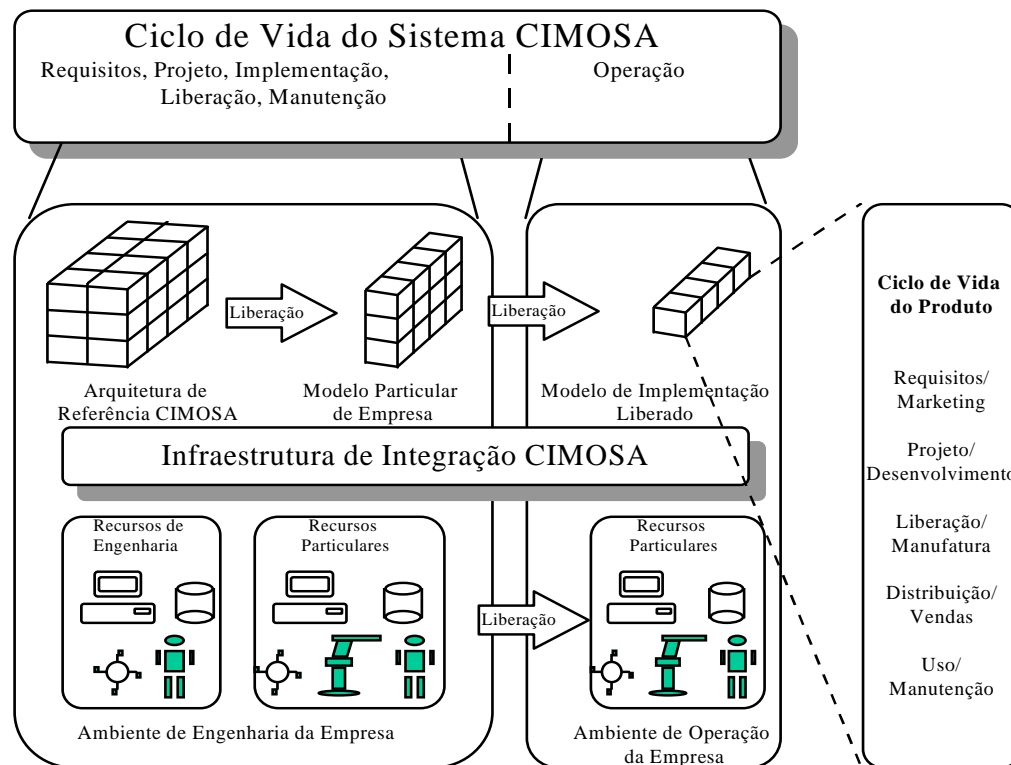


Figura 2 - Estrutura Arquitetural de CIMOSA (adaptado de VERNADAT, 1996).

Além deste princípio de Particularização de modelos (a partir de modelos de referência), a estrutura de modelagem CIMOSA possui os princípios de Derivação e Geração de modelos.

O princípio de Derivação modela as empresas de acordo com três sucessivos níveis de modelagem (iterações entre esses níveis são, é claro, permitidos):

a) definição de requisitos para expressar as necessidades do negócio como percebido pelos usuários;

- b) especificação de projeto para construir um modelo formal, conceitual e executável do sistema da empresa (tempo é considerado);
- c) descrição da implementação para documentar detalhes da implementação, recursos instalados, mecanismos de gerenciamento de exceções, e considerar sistemas não deterministas.

O princípio de Geração, o qual recomenda modelar empresas de manufatura de acordo com quatro básicos e complementares pontos de vista (outras vistas podem ser definidas):

- a) a vista de função que representa a funcionalidade e comportamento da empresa (isto é, eventos, atividades e processos) incluindo aspectos temporais e de gerência de exceções;
- b) a vista de informação, o qual representa objetos da empresa e seus elementos de informação;
- c) a vista de recursos, o qual representa meios da empresa, suas capacidades e gerenciamento;
- d) a vista de organização, o qual representa níveis organizacionais, autoridades, e responsabilidades.

Como já descrito, arquiteturas de modelagem e modelos de referência têm como objetivo facilitar o trabalho de modelagem e fornecer um entendimento comum sobre os sistemas de empresa. Para a descrição dos modelos é necessária uma linguagem de modelagem.

6. UML

A UML (*Unified Modeling Language*) “é uma linguagem gráfica para especificação, construção, visualização e documentação de um sistema de *software* (BOOCH, et al. 2000)”.

A UML utilizou o Diagrama de Estado, Diagrama de Classe, Diagrama de Objetos (de onde surgiu o Diagrama de Colaboração), o Diagrama de Processo (originando o Diagrama de Implementação) e o Diagrama de Módulo (resultando o Diagrama de Componente). O método Fusion também teve sua colaboração com o Grafo de Interação de Objetos. E o diagrama de estado (Statecharts) de Harel, contribuiu para a criação do Diagrama de Atividade (LARMAN, 2000). A seguir descrevem-se os diagramas da UML (LARMAN, 2000; FURLAN, 1998):

Pode-se dizer que o objetivo principal da UML é definir uma linguagem de modelagem visual e expressiva, no sentido de prover facilidades na visualização, ou seja, o pleno entendimento das funções de um sistema a partir de diagramas que o representem, no gerenciamento de complexidade, permitindo uma representação simplificada das atividades do sistema, ou seja, que cada aspecto funcional dele seja representado em modelos específicos e, por fim, na comunicação, unificando a comunicação da equipe de desenvolvimento na forma de diagramas. Abaixo é mostrado a modelagem do planejamento agregado que é definido por Heizer Render (2001) como uma atividade elaborada entre o setor comercial, setor de produção, compras e direção da empresa.

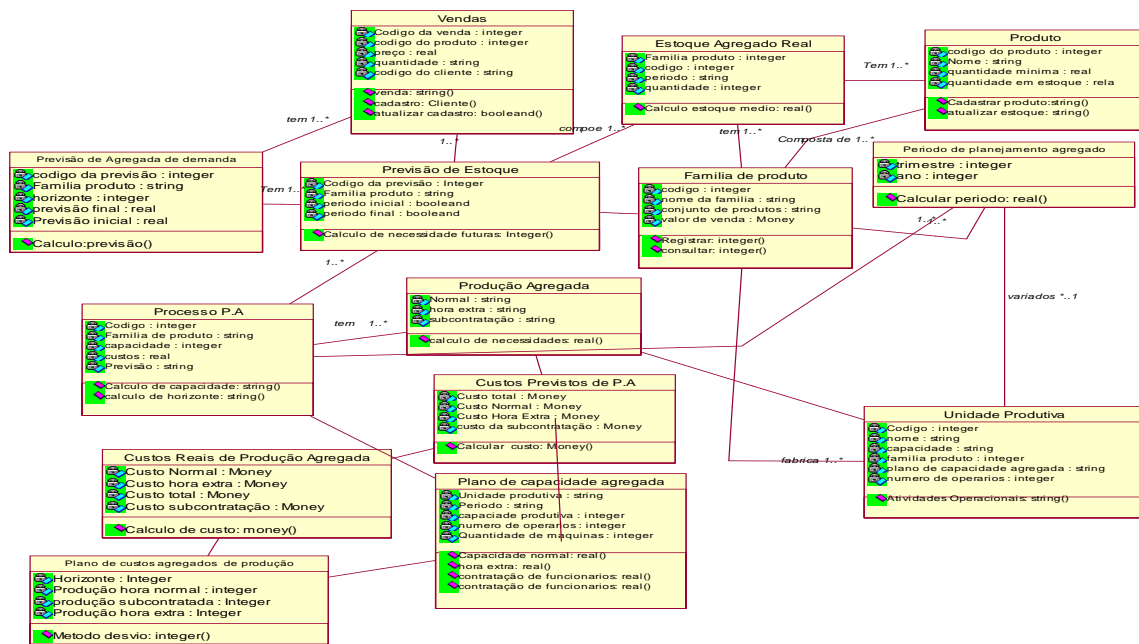


Figura 3 - Modelagem do Planejamento Agregado em UML.

No tocante a partir da modelagem em UML foi possível a geração de códigos gerando um protótipo em

Delph com as características abaixo.

Figura 4 - Diagrama de Classe Planejamento Agregado Proposto para o Modelo ERP 5

7. Considerações Finais

Um sistema ERP pode auxiliar as empresas na busca por competitividade, mas a sua adoção fica dificultada devido ao seu custo de compra e a dependência da empresa fornecedora para possíveis adaptações do sistema, devido a não se ter acesso e conhecimento para alterações no seu código. Softwares livres e de código aberto, como por exemplo sistemas ERPs, podem ser uma alternativa vantajosa, mas para sua adoção na prática são necessários o desenvolvimento e utilização de técnicas e ferramentas que facilitem a implantação e alteração desses softwares. Uma arquitetura de modelagem e modelos de referência é essencial para viabilizar o desenvolvimento, implantação e alterações de ERPs de código aberto. Para a definição de uma arquitetura de modelagem para o projeto ERP5 estuda-se a possibilidade de utilizar os conceitos da estrutura de modelagem CIMOSA e da arquitetura proposta de por Eriksson & Penker (2000). Os modelos de referência para os módulos do sistema ERP5 deverão ser gerados de forma a “mapear” e documentar (ou seja, modelar) os processos e informações genéricas, os quais podem servir de base para adaptações (ou particularizações) através de novos códigos. No Projeto ERP5 foi adotada a linguagem UML, que se tornou um padrão de fato, mundialmente aceito e usado, o que facilita a difusão de modelos e códigos. Atualmente trabalha-se no módulo de Planejamento, Programação e Controle da Produção.

REFERÊNCIAS

- AZEVEDO, D. P. J. (2003) - *Aplicação da Técnica de Modelagem de Negócio com UML a processos Iterativos de desenvolvimento de software*. Tese de Mestrado em Engenharia de Produção-UENF.Campos.
- BOOCH, G.; RUMBAUGH, J. & JACOBSON, I. (2000)- *UML Guia do Usuário*. Tradução de Fábio Freitas da Silva.-Rio de Janeiro:Campus.
- CARVALHO, R. de A. (2003) - *Desenvolvimento de Sistema ERP Avançado e de Código Aberto para Pequenas e Médias Empresas*. Projeto designado a CNPq. CEFET Campos.
- COLANGELO, L. F. (2001) - *Implantação de Sistemas ERP: Um Enfoque de Longo Prazo*.São Paulo:Atlas.
- CORRÊA, H. L.; GIANESI, I. G.N. & CANON M. (2000) - *Planejamento, programação e controle da produção: MRP/ERP:conceitos, uso e implantação*.-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.
- GOULART, C. P. (2000) - *Proposta de um modelo de referência para planejamento e Controle de Produção em Empresas Virtuais*. Dissertação de Mestrado Escola de Engenharia de São Carlos da Universidade de São Paulo.São Paulo.
- HEIZER, Jay; RENDER, Barry.(2001). *Administração de Operações- Bens e Serviços*. Trad. Dalton Conde de Alencar-5 ed.São Paulo: LTC.
- KELLER,G. & TEUFEL, T. S. (1998) - *Process Oriented Implementation*.Harlow,Addison-Wesley.
- LARMAN, C. (2000) - *Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientados a Objetos*.Porto Alegre: Bookman,.
- PÁDUA, S. I. D. & CAZARINI, E. W. (2002) - *Modelagem Organizacional para capturar os requisitos organizacionais*. USP- EESC- Escola de Engenharia de São Carlos. Anais, SIMPEPVIII.
- PIDD, M. (1998) - *Modelagem Empresarial, ferramentas para tomada de decisão*; trad Gustavo severo de Borba et al.-Porto Alegre:Artes Medicas.
- SCHEER, A.W. (1998) – *Aris - Bussines process Framewors* . Berlin, Sringer Verlag.
- SOLANES, J. P. S. & CARVALHO, R. A. (2003) *An Abstract Model For an open source Erp system: The Erp5 proposal*.Rio de Janeiro.Enegep.
- VERNADAT F.B. (1996) - *Enterprise Modeling and Integration, Principles and Aplications*, Chapman e Hall.1996.

PETERS, James & PEDRYCZ, Witold.(2001) *Engenharia de Software: Teoria e Prática*. Tradução de Ana Patrícia Garcia. Ed.Campus.Rio de Janeiro.

PRESSMAN, Roger. *Engenharia de Software*. Tradução José Carlos Barbosa dos Santos. São Paulo: Makon Brooks, 1999.