



Desenvolvimento de um Sistema de Informações para Manufatura utilizando Análise Orientada a Objetos

Niederauer Mastelari

Departamento de Engenharia de Fabricação, Faculdade de Engenharia Mecânica, Unicamp
Campinas, SP, CEP: 13083-970, niede@fem.unicamp.br

Nivaldo Lemos Coppini

Departamento de Engenharia de Fabricação, Faculdade de Engenharia Mecânica, Unicamp
Campinas, SP, CEP: 13083-970, coppini@fem.unicamp.br

Resumo. *Este trabalho apresenta o projeto de um sistema de informações para ambientes de manufatura. Ele é constituído de dois subsistemas, o primeiro, tipicamente funcional e apropriado para engenharia, tem a função de programar a produção de peças em centros de torneamento. O outro é um gerenciador de banco de dados, tipicamente um processador de transações, voltado à administração da área de materiais. Desta forma ele integra diferentes áreas relacionadas com a produção. Neste trabalho será utilizada a metodologia orientada a objetos apresentada por Rumbaugh et alli, denominada Técnica de Modelagem de Objetos (TMO).*

Palavras-chave: *Sistemas de Informações, Análise orientada a objetos, Torneamento.*

1. Introdução

Laudon (2001), define sistema de informações (SI) como um conjunto de componentes inter-relacionados que coleta (ou recupera), processa, armazena e distribui informação para dar suporte à tomada de decisão e ao controle da organização. Além de apoiar, coordenar e controlar a tomada de decisão, os SIs também podem ajudar os gerentes e trabalhadores a analisar problemas, visualizar assuntos complexos e novos produtos. Bio (1991), mostra a relação de interdependência entre os SIs e a estrutura organizacional das empresas. Se ambos estão inexoravelmente interligados, as alterações, as mudanças, o replanejamento de qualquer deles afetarão o outro. Stair (1998), mostra que uma das conseqüências da implantação dos sistemas de informação é o menor crescimento da estrutura organizacional ao longo do tempo. Um exemplo disto é o empowerment, técnica que utiliza os SIs como um elemento-chave para as empresas delegar poder a seus empregados. Frequentemente, o empowerment significa tornar os SIs disponíveis aos empregados de níveis inferiores na hierarquia. Empowerment significa dar aos empregados, trabalhadores e gerentes mais poder, responsabilidade e autoridade para tomarem decisões, adotarem certas atitudes e terem mais controle sobre seus trabalhos. O empowerment geralmente resulta em ações mais ágeis e uma rápida solução de problemas, podendo também reduzir custos e gerar produtos e serviços de melhor qualidade.

Este trabalho vai ao encontro desta técnica ao apresentar o projeto de um SI que integra operações de torneamento da produção com banco de dados de ferramentas da área de materiais. O programador a partir do desenho da peça, utilizando editor implementado por Mastelari (1996) com uma linguagem técnica mecânica, gera e simula o programa de comando numérico (CN) da peça com condições de corte otimizadas. Inerentes a este processo são feitos os cálculos de tempos e as requisições das necessidades de ferramentas e materiais, com conseqüente orçamento.

A finalização se dá com a requisição das ferramentas necessárias diretamente à área de materiais. Isto leva à diminuição de desperdícios com compras excessivas de ferramentas e a melhor utilização da mão de obra através da sua qualificação. O SI confere ao projetista maior autonomia, e autoridade na solução de problemas e tomada de decisões dentro do processo produtivo. Este sistema será desenvolvido usando a Técnica de Modelagem de Objetos (TMO) apresentada por Rumbaugh et alii, (1994).

2. Descrição do Sistema

O SI é constituído por duas partes principais: O editor/simulador de programas CN para torneamento e o Banco de dados de ferramentas. O editor/simulador tem interface amigável, baseada em linguagem técnica mecânica, e poucos são os detalhes computacionais necessários para a sua completa manipulação. Ao programador basta responder perguntas que são apresentadas seqüencialmente na forma de janelas de diálogos, para que percursos e condições operacionais de usinagem sejam automaticamente organizados. O simulador ajuda na descrição da peça e na verificação da execução das operações. Ao usar o editor de programas CN o usuário poderá definir se deseja trabalhar no modo de máxima produção, de mínimo custo de produção ou especificar outras condições tecnológicas para o torneamento. Ao definir as operações, o banco de dados de ferramentas é acessado de forma a facilitar a escolha do ferramental. Na programação da peça o programador estipula o número de peças que irá produzir. Com este dado, mais os dados geométricos da operação, dados das ferramentas (X e K de Taylor), dados econômicos, calcula-se: O tempo de cada operação, o número de insertos que irá necessitar para cada operação, os custos envolvidos. A partir do conjunto de dados geométricos e tecnológicos obtidos gera-se o programa CN para a máquina especificada. Finalmente, caso se decida pela produção da peça é executada a transação de requisição ao estoque de ferramentas. Para que a transação seja concluída o funcionário deverá informar seus dados e senha. O banco de dados mantém informações sobre o ferramental e é acessado pela produção, materiais e compras. Ele é programado para reposição automática dos estoques de ferramentas e ativa o processo de compra a partir de critérios previamente definidos.

3. Modelamento, Análise e Técnicas Orientadas a Objetos.

Segundo Cougo(1997), modelo é a representação abstrata e simplificada de um sistema real, com a qual se pode explicar ou testar o seu comportamento, em seu todo ou em partes. Através dos modelos aprende-se a observar e a pensar um sistema real, passando então a representá-lo. Em um modelo normalmente ressaltam-se os aspectos de interesse e minimizam-se os de menor relevância para o objetivo do trabalho. O observador verificando e experimentando, com menor risco, esta realidade, busca apreendê-la. De acordo com Coad (1992), análise é o estudo de um domínio de problemas é o processo de extrair as necessidades de um sistema, ou seja, o que o sistema tem que fazer para satisfazer o cliente. A análise de sistemas no mundo orientado a objetos é feita analisando os objetos em um ambiente e os eventos que interagem com estes objetos. O projeto de software é feito pela reutilização de classes de objetos e, quando necessário da construção de novas classes. Rumbaugh (1994) define objeto como um conceito, uma abstração algo com limites e significado em relação ao problema em causa. Todos os objetos têm identidade e são distinguíveis. Uma classe de objetos descreve um grupo de objetos com propriedades semelhantes (atributos), o mesmo comportamento (operações), os mesmos relacionamentos com outros objetos e a mesma semântica. A herança é o compartilhamento de atributos e operações entre classes em um relacionamento hierárquico. Uma classe pode ser definida de forma abrangente e depois refinada em sucessivas subclasses. A herança significa que uma classe de objetos pode ser criada a partir de outra. A herança é o mecanismo para a implementação da propriedade generalização-especialização. Segundo Martin (1992), técnicas orientadas a objetos podem ser usadas para simplificar o projeto de sistemas complexos. O sistema pode ser visualizado como uma coleção de objetos, com cada

objeto estando em um dos muitos estados especificados. As operações que mudam o estado são relativamente simples. Objetos são construídos de outros objetos. Sistemas são construídos invocando operações de componentes previamente testados que tem formato padrão. As técnicas Orientadas a Objetos sozinhas não possibilitam a magnitude de mudanças necessárias, elas tornam-se poderosas quando combinadas com outras tecnologias tais como: Sistemas gerenciadores de Banco de Dados, tecnologias Cliente servidor, Bibliotecas de classes, ferramentas CASE (Computer-Aided Software Engineering), inteligência artificial, programação visual.

4. A modelagem TMO

Esta metodologia é descrita em detalhes por Rumbaugh et al (1994). A metodologia consiste na construção de um modelo de um domínio do problema e na posterior adição dos detalhes de implementação durante o projeto do sistema. A metodologia consiste nas etapas de Análise, Projeto do sistema, Projeto dos Objetos e implementação. Na análise partindo do enunciado do problema, o analista constrói um modelo da situação do mundo real, mostrando suas propriedades relevantes. Segundo Mendes(1996), ela visa planejar um modelo do mundo real que seja conciso, entendível e correto. O produto da etapa de análise é uma abstração da situação do mundo real, composta pelos três modelos TMO: o modelo de objetos, o modelo dinâmico e o modelo funcional. O modelo de objetos descreve os objetos do sistema e seus relacionamentos, mostra a estrutura estática do sistema e a organiza em partes a serem trabalhadas; o modelo dinâmico descreve as interações entre os objetos do sistema, representa os aspectos temporais e comportamentais de controle de um sistema; e o modelo funcional, descreve as transformações dos dados do sistema. A descrição completa do sistema exige todos os três modelos. No projeto, o analista toma decisões relativas a arquitetura do sistema, alocação de recursos, organização do sistema em subsistemas, estratégias de ataque ao problema. No projeto de Objetos o enfoque principal se dá nas estruturas de dados e algoritmos necessários para implementar as classes. A implementação deve ser a parte menor e mais mecânica do ciclo de desenvolvimento, pois todas as decisões mais complexas deveriam ter sido tomadas durante as fases anteriores do projeto. Esta fase pode ter seu trabalho muito reduzido com a utilização de ferramentas CASE que geram o código automaticamente. Neste trabalho serão tratadas as etapas de análise e projeto.

4.1. Modelagem de Objetos

O modelo de objetos descreve as classes de objetos do mundo real e as suas associações. Este modelo costuma ter melhor definição, menor dependência de detalhes, maior estabilidade e ser mais fácil de ser compreendido. Ele é representado graficamente com diagramas de objetos contendo as classes dos objetos e um dicionário de dados para estas classes. O modelo de objetos é o mais importante dos três modelos, porque na TMO é dado maior ênfase na construção de um sistema em função dos objetos. O símbolo da TMO para representar uma classe é um retângulo dividido em três campos: o primeiro contém o nome da classe em negrito, o segundo os atributos da classe e o terceiro as operações definidas na classe. Uma associação é representada através de uma linha ligando as classes. A multiplicidade de uma associação especifica quantas instâncias de uma classe relacionam-se a uma única instância de uma classe associada, ela restringe a quantidade de objetos relacionados. Uma linha sem símbolos de multiplicidade indica uma associação um para um. Existem terminadores especiais de linha para indicar certos valores comuns de multiplicidade. Um círculo preenchido é o símbolo para muitos, significando zero ou mais. A agregação é o relacionamento “todo-parte” ou “uma parte de”, no qual objetos são partes ou componentes associados a um objeto que representa o todo ou estrutura inteira. A agregação na TMO é representada como uma associação com um pequeno losango que indica a extremidade do relacionamento. A representação TMO para a generalização especialização (herança) é um triângulo interligando uma superclasse a suas subclasses. O diagrama de objetos para o sistema, sem atributos e operações é apresentado pela fig. (1). O dicionário de dados, fig. (2), apresenta de forma simplificada o significado das classes do diagrama de objetos.

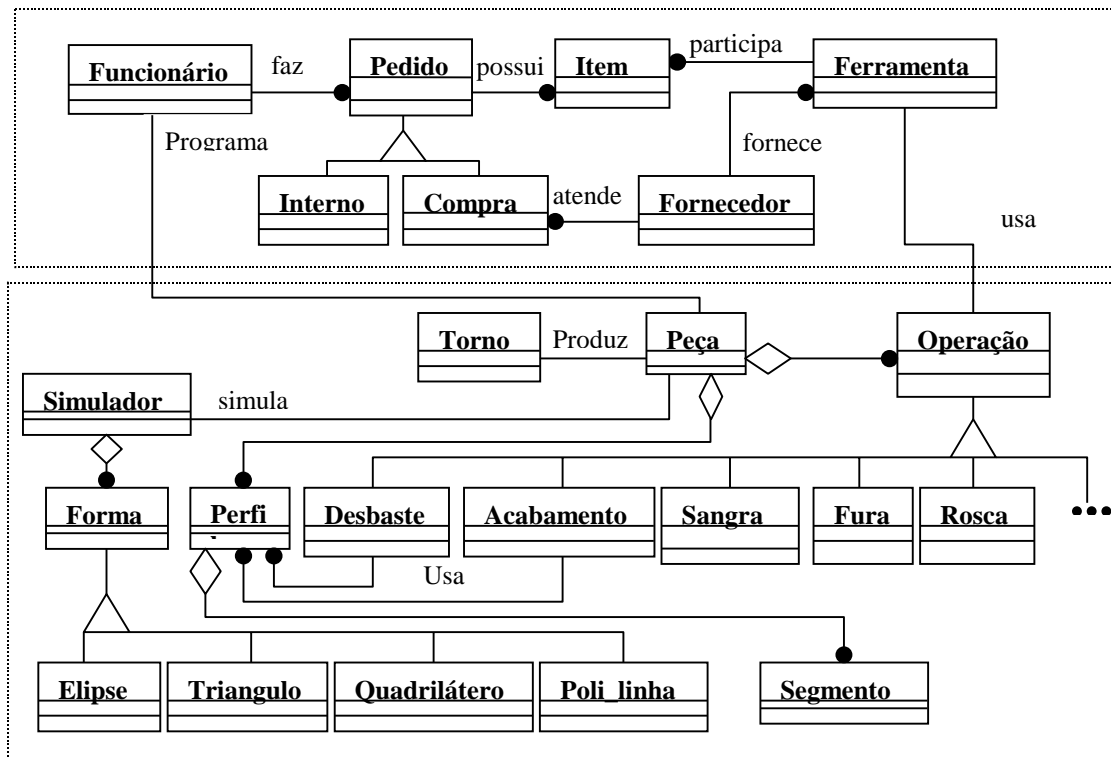


Figura1- Diagrama de Objeto

- Funcionário** – Conjunto de pessoas da empresa autorizadas para atuar sobre os processos.
- Pedidos** – Solicitações de ferramentas que possibilitam a execução dos processos.
- Interno (pedido)** – solicitação de ferramentas para o estoque ao se programar uma peça.
- Compra (pedido)** – solicitação de ferramentas enviada para o fornecedor.
- Item** – Cada uma das descrições de ferramentas que constam do pedido.
- Ferramentas** – Insertos, porta ferramentas e outros itens para o processo de torneamento.
- Fornecedor** – Empresa comerciante de ferramentas, é exclusiva para cada item.
- Torno** – Centro de torneamento que produz a peça.
- Peça** – Produto final do processo.
- Operações** – Conjunto de comandos CN com características próprias.
- Desbaste (operação)** – Retira o material entre o perfil bruto e o perfil acabado com sobremetal.
- Acabamento (operação)** - Retira o sobremetal deixado sobre o perfil acabado.
- Sangramento (operação)** – Realiza rasgos, normalmente através de bedames.
- Furação (operação)** – Realiza furos com brocas.
- Rosca (operação)** – Realiza roscas.
- Perfil** – Descrição geométrica da peça.
- Segmento** – Descrição de um trecho da peça constituído por dois pontos e uma interpolação.
- Simulador** – Responsável pela descrição gráfica do processo.
- Forma** – Classe genérica que dá origem às figuras geométricas que constituem o simulador.
- Elipse, triangulo, quadrilátero (forma)** – Figuras geométricas básicas.
- Poli_linha** – Seqüência de pontos ligados por uma linha.

Figura 2 - Dicionário de dados para as classes do sistema

4.2. Modelagem Dinâmica

Os aspectos de um sistema relativos ao tempo e às suas modificações formam o modelo dinâmico em contraste com o estático, ou modelo de objetos. O modelo dinâmico é representado

graficamente com um cenário de eventos, um traçado de eventos para o cenário, um diagrama de fluxo de eventos e diagramas de estado. Chamamos de eventos os estímulos externos que o sistema recebe. Um estímulo de um objeto para outro é também um evento. Um evento idealmente não tem duração.

Peça : Solicita dados de configuração ao **funcionário**
Funcionário: Fornece dados
Peça: Solicita dados de cada segmento de cada perfil ao **funcionário**
Funcionário: Fornece dados
Peça: Envia dados ao **simulador**
Simulador: Apresenta perfil com último segmento realçado.
Peça: solicita dados de cada uma das operações ao **funcionário**
Funcionário: Fornece dados
Peça: solicita dados de ferramentas ao objeto **ferramenta**
Ferramenta: Solicita escolha da ferramenta da operação ao **Funcionário**
Funcionário: Escolhe ferramenta
Ferramenta: envia dados ao objeto **peça**
Peça: apresenta serviços e solicita senha para requisição de ferramentas ao **funcionário**
Funcionário: Apresenta senha
Peça: Processa transação com **ferramenta**
Ferramenta: Confirma transação

Figura 3 – Cenário do Sistema

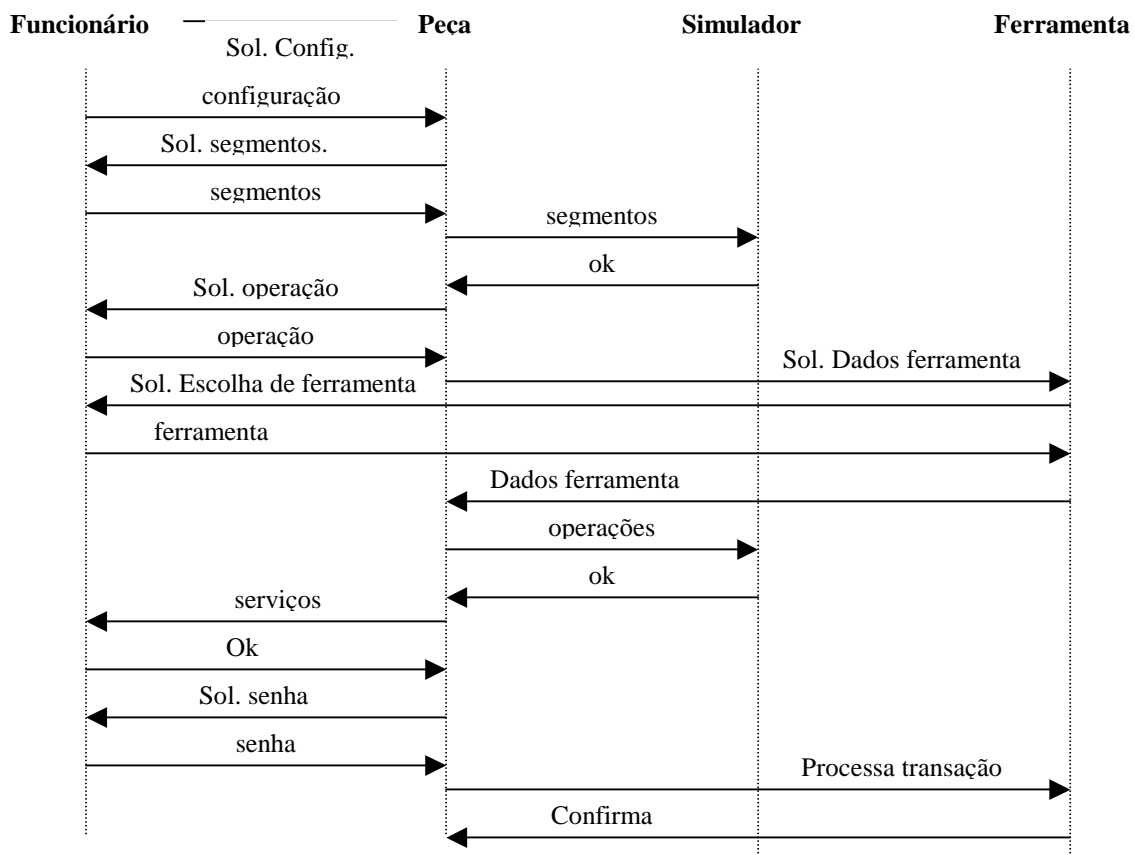


Figura 4 - Traçado de eventos para o cenário de atuação do sistema

Um cenário é uma seqüência de eventos que ocorrem durante uma determinada execução de um sistema. A abrangência de um cenário pode variar, normalmente em uma descrição de um cenário só os eventos relevantes ou aqueles que ocorrem com maior freqüência são apresentados. Os estados representam os valores que os objetos assumem devido a um evento. O padrão de estados, eventos e transições de estados para uma determinada classe de objetos podem ser abstraídos e representados por um diagrama de estados. Um diagrama de estados é um grafo direcionado, seus vértices são os estados e os arcos são transações rotuladas com os nomes dos eventos. Um estado é representado por uma figura arredondada contendo um nome opcional. O diagrama de eventos especifica as seqüências de estados causados por uma seqüência de eventos. Um diagrama de estado descreve o comportamento de uma única classe de objetos. Cada objeto é independente de outros objetos e comporta-se ao seu próprio modo. Os diagramas de estados podem representar ciclos de vida, com um estado inicial indicado por um círculo cheio, estados intermediários e um estado final representado por um círculo cheio dentro de um outro. A fig (3) mostra um cenário do sistema, a fig(4) apresenta o traçado de eventos para o cenário apresentado para o sistema, a fig(5) o diagrama de fluxo de eventos e a fig(6) apresenta o diagrama de estados para da classe Peça.

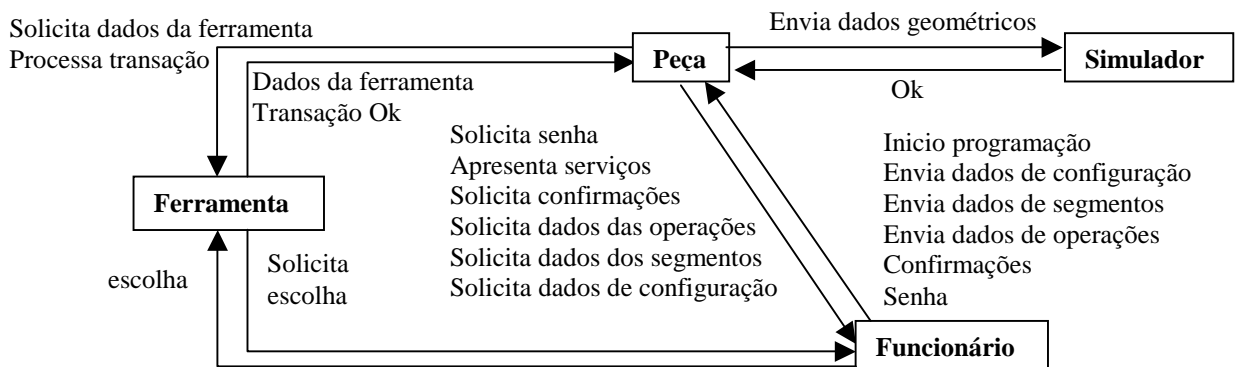


Figura 5 - Diagrama de fluxo de eventos

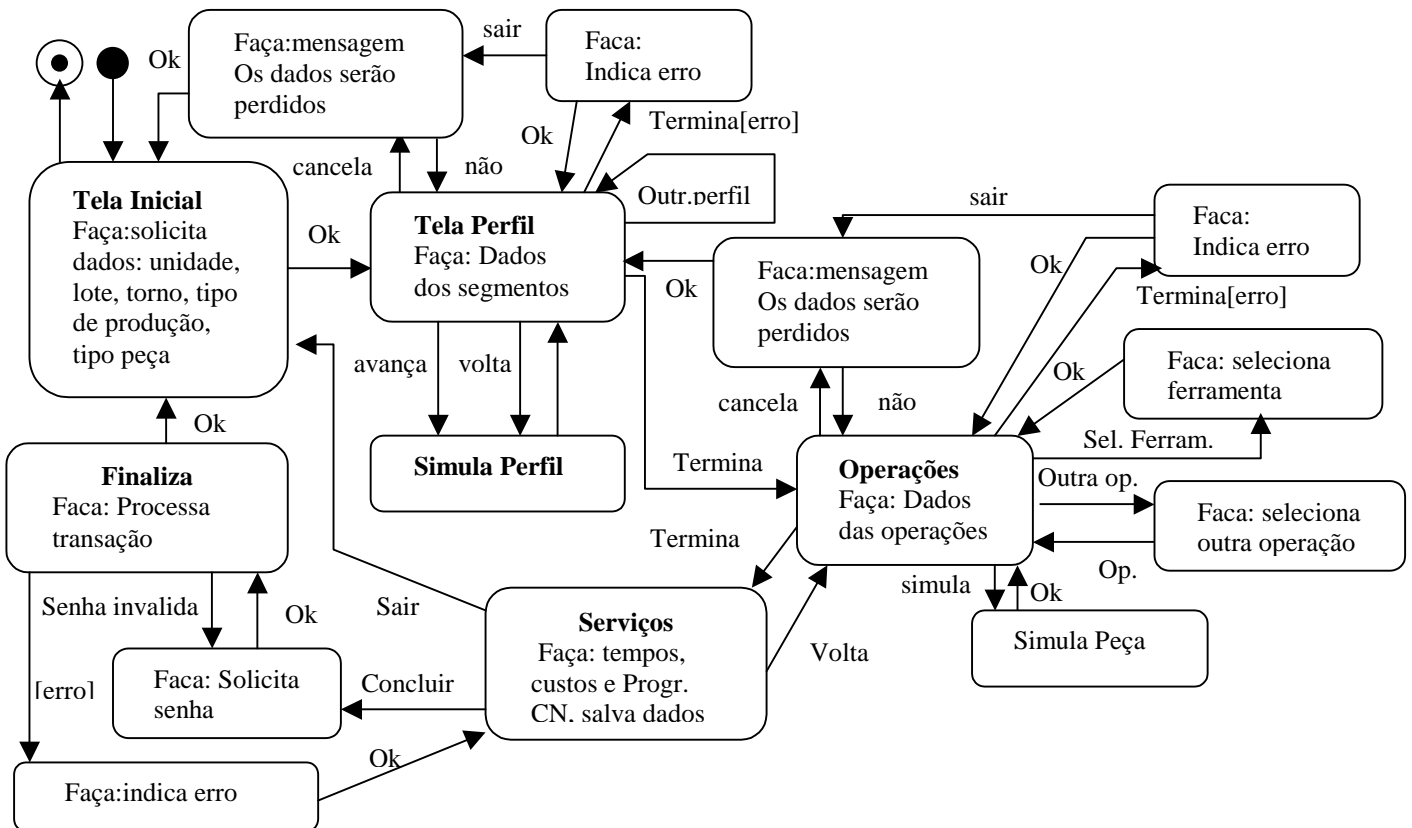


Figura 6 - Diagrama de estados da classe Peça

4.3. Modelagem Funcional

O modelo funcional descreve os cálculos executados em um sistema. O modelo funcional especifica o que acontece, o dinâmico especifica quando acontece e o modelo de objetos, o que acontece a quem. Ele especifica o significado das operações do modelo de objetos e as ações do modelo dinâmico. O modelo funcional é composto por múltiplos diagramas de fluxos de dados, os DFDs. É uma representação que mostra o fluxo de dados, desde sua origem nos objetos, seu processamento nos processos que os transformam, e os seus destinos: outros objetos ou depósitos de dados. Um processo transforma valores de dados. Os processos são implementados como métodos de operações em classes de objetos. Um ator é um objeto ativo que participa do DFD produzindo ou consumindo valores. Um depósito de dado tem a função de armazenar dados para o uso futuro, eles não geram dados apenas atende a solicitações de armazenamento e acesso aos dados, são representados nos DFDs como um par de linhas com o nome do depósito. Tanto os atores como os depósitos de dados são objetos. Um processo pode ser expandido em outro diagrama de fluxo de dados com um maior detalhamento, cada entrada ou saída do processo é uma entrada ou saída do novo diagrama. Os diagramas podem ir a um detalhamento que for necessário atingindo um detalhamento arbitrário dos processamentos. Na fig (7) temos o diagrama de fluxo de dados de alto nível do sistema.

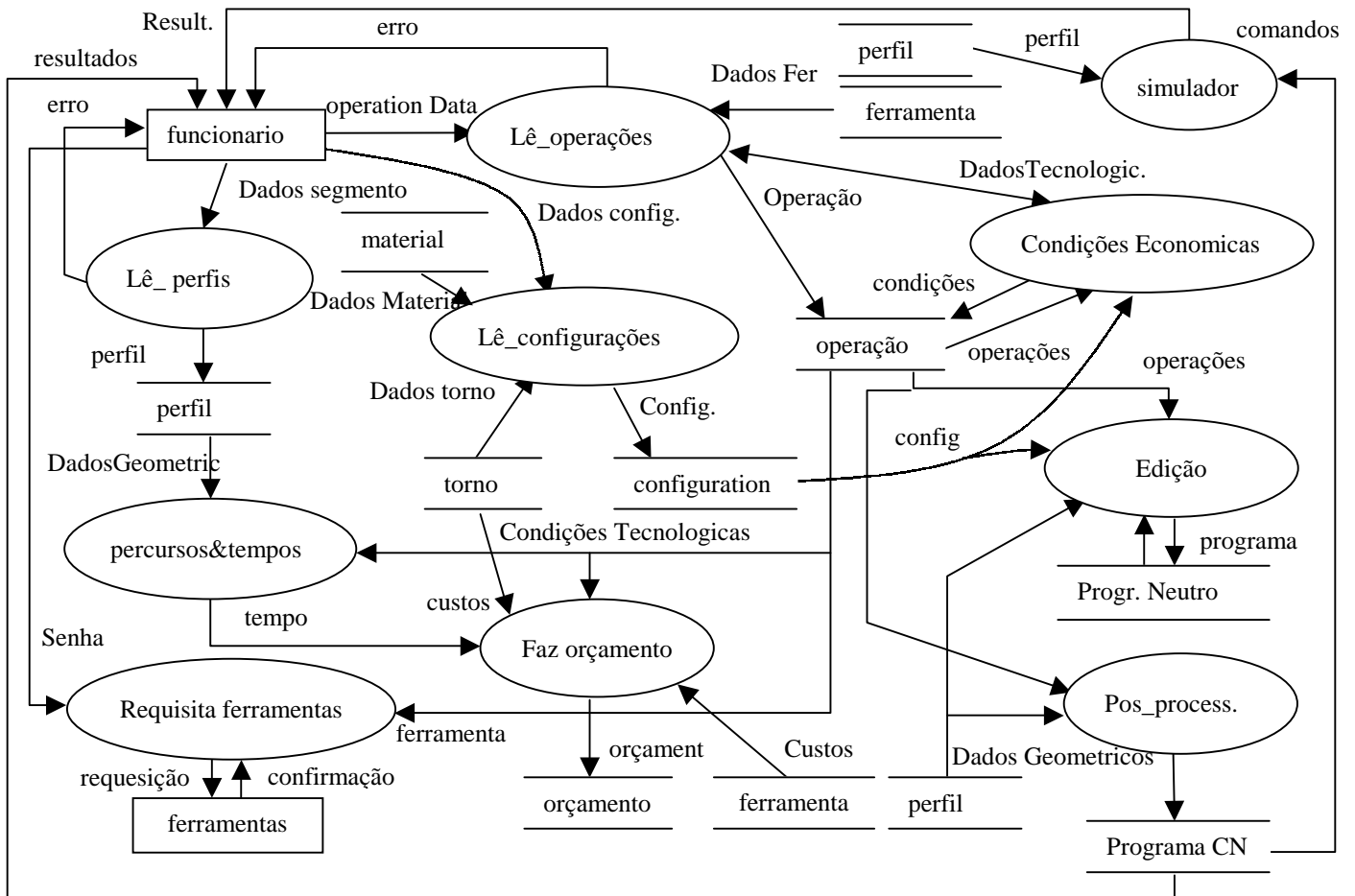


Figura 7 –Diagrama de fluxo de dados do sistema

4.4. Projeto

Este sistema será dividido em dois subsistemas o editor/simulador e o Banco de Dados de Ferramentas. O relacionamento entre estes dois tipos de sistema será do tipo cliente servidor. Os subsistemas atuam em computadores diferentes em uma mesma rede local de computadores. Os sistemas operacionais das máquinas em rede serão Microsoft Windows 98, Windows NT ou versões atualizadas destes sistemas. O controle será seqüencial baseado em procedimentos. A arquitetura

do editor/simulador será do tipo interface interativa porque ele é dominado pelas interações com agentes externos. A arquitetura do Banco de Dados de Ferramentas será do tipo processador de transações, pois sua principal função é armazenar e acessar informações. O subsistema Banco de dados de Ferramentas será implementado como um banco de dados relacional, desta forma suas classes serão implementadas como tabelas.

5. Conclusão

Este trabalho objetivou apresentar o projeto de um sistema de informações para ambientes de manufatura que integra serviços da produção com outras áreas como materiais e compras. Para isto foi utilizada a técnica TMO de modelagem. Através desta técnica os conceitos de orientação a objeto podem ser aplicados por todo o ciclo de vida do desenvolvimento de um sistema, desde a análise até a implementação do projeto. As mesmas classes definidas em um estágio podem ser passadas para outro sem mudanças na notação, embora ganhem detalhes de implementação nos últimos estágios. Procurou-se apresentar modelos que representem este sistema de forma concisa, entendível e correta de forma a serem usados como diretrizes seguras que evitam inconsistências na sua implementação e agilizem o seu desenvolvimento. Com a implantação deste sistema o programador CNC pode tomar decisões de forma autônoma, otimizada, integrada e rápida para a produção de peças e nas conseqüentes requisições de materiais.

6. Referências

- Bio S. R., 1991, "Sistemas de Informação – Um enfoque empresarial", Editora Atlas, São Paulo, Brasil, 183p.
- Coad P., Yourdon E., 1992, "Análise Baseada em Objetos", Editora Campos, São Paulo, Brasil, 225p.
- Cougo P., 1997, "Modelagem Conceitual", Editora Campus, Rio de Janeiro, Brasil, 284p.
- Laudon K. C., Laudon J. P., 2001, "Gerenciamento de Sistemas de Informação", Editora LTC, Rio de Janeiro, Brasil, 433p.
- Mastelari, N., 1996, "Desenvolvimento de um editor/simulador para centros de torneamento CN", São Paulo: Faculdade de Engenharia Mecânica, Universidade de Campinas. / Tese de Mestrado apresentada a FEM-UNICAMP.
- Martin J., Odell J.J., 1992, "Object Oriented Analysis and Design", Prentice –Hall, New Jersey, EUA, 513p.
- Mendes, R. M., 1996, "Estudo sobre a Passagem de Especificações Orientadas a Objeto para Implementação no SGBDOO O2", Universidade Federal do Rio Grande do Sul, Porto Alegre, Instituto de Informática, 116p. Tese (Mestrado).
- Rumbaugh J., Blaha M., Premerlani W., Eddy F. Lorensen W., 1994, "Modelagem e Projetos Baseados em Objetos", Editora Campus, Rio de Janeiro, Brasil, 652p.
- Stair, R.M., 1998, "Princípios de Sistemas de Informação", Editora LTC, Rio de Janeiro, Brasil, 451p.

Development of Manufacturing Information System using Object Oriented Analysis

Abstract: *This work presents the development of manufacturing Information System. This system integrates two information models, one typically functional, suitable to use in engineer applications what programming production parts. The other is a Database management system, which processes transactions used in material control. This system integrates the production and material areas. This work uses Oriented Object methods presents by Rumbaugh et alli, call Object Modelling Technique (OMT).*

Keywords: Information Systems, Object Oriented Analysis, Turning.