

**Design de um ambiente computacional de
modelagem
e simulação para formação de pessoal na
indústria**

Edmar Lourenço Borges

Dissertação de Mestrado

**Design de um ambiente computacional de
modelagem
e simulação para formação de pessoal na indústria**

Edmar Lourenço Borges

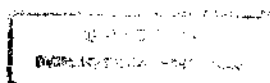
Abril de 1998

Banca Examinadora:

- Profa. Dra. Maria Cecília Calani Baranauskas ¹ (Orientadora)
- Prof. Dr. Alexandre Ibrahim Direne ²
- Profa. Dra. Eliane Martins ¹
- Profa. Dra. Heloisa Vieira da Rocha ¹ (Suplente)

¹Instituto de Computação, Unicamp.

²Departamento de Informática, UFPR.



UNIDADE	BC
N.º CHAMADA:	
CLASS.	347.21
EX.	395/98
C I	D ☒
PREÇO	R\$ 11,00
DATA	11/08/98
N.º CPD	

CM-00114615-5

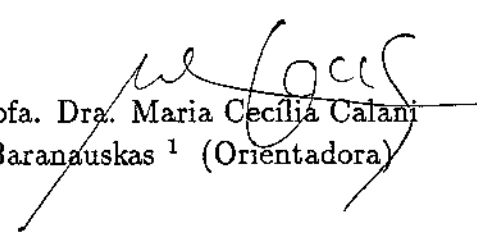
FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

B644a	Borges, Edmar Lourenço Design de um ambiente computacional de modelagem e simulação para formação de pessoal na indústria / Edmar Lourenço Borges -- Campinas, [S.P. :s.n.], 1997. Orientador : Maria Cecília Calani Baranauskas Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica. I. Simulação e modelagem. 2. Indústria - Treinamento. 3. Ensino auxiliado por computador. 4. Projeto de sistema. I. Baranauskas, Maria Cecília Calani. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.
-------	--

Design de um ambiente computacional de modelagem e simulação para formação de pessoal na indústria

Este exemplar corresponde à redação final da
Dissertação devidamente corrigida e defendida
por Edmar Lourenço Borges e aprovada pela
Banca Examinadora.

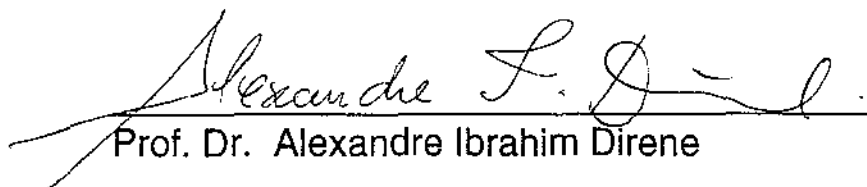
Campinas, 12 de novembro de 1997.

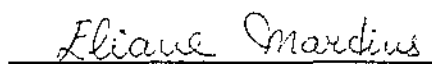


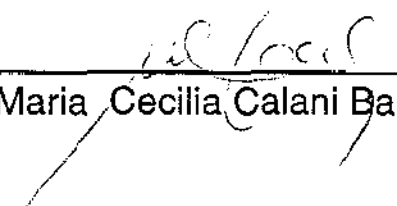
Prof. Dra. Maria Cecília Calani
Baranauskas ¹ (Orientadora)

Dissertação apresentada ao Instituto de Com-
putação, UNICAMP, como requisito parcial para
a obtenção do título de Mestre em Ciência da
Computação.

Tese de Mestrado defendida e aprovada em 11 de dezembro de 1997 pela Banca Examinadora composta pelos Professores Doutores


Prof. Dr. Alexandre Ibrahim Direne


Prof^a. Dr^a. Eliane Martins


Prof^a. Dr^a. Maria Cecilia Calani Baranauskas

Resumo

O mundo atual exige que as indústrias sejam dinâmicas na absorção de novas tecnologias. Isto requer que tenham à sua disposição métodos eficientes de formação de pessoal, com o intuito de tornar os funcionários cada vez mais capacitados, possibilitando a prática de novas tecnologias na linha de produção.

Este trabalho discute a construção de um ambiente computacional de modelagem e simulação para o contexto de formação na indústria. Aborda a problemática envolvida em se projetar e desenvolver um ambiente de modelagem e simulação que permita a funcionários do chão de fábrica praticar novas filosofias de trabalho em um contexto de manufatura.

Como um projeto voltado para treinamento em manufatura, considera-se o contexto de uma fábrica real, com o intuito de entendimento do domínio da aplicação como um todo e a posterior tradução desse entendimento para o design de um ambiente de aprendizado baseado na mídia do computador. Envolve técnicas de design de interfaces que utilizam design centrado no usuário e incorporam as principais características do paradigma construcionista de aprendizagem.

O trabalho descreve o processo de desenvolvimento do ambiente, discutindo a metodologia adotada, os ambientes de implementação e avaliação informal de usabilidade. O protótipo resultante do ambiente é apresentado juntamente com uma descrição de sua funcionalidade. No final são discutidas perspectivas futuras de continuidade do trabalho.

Abstract

The current world demands of the industries being dynamic in the absorption of new technologies. This requests them efficient methods of personnel's formation, with the aim of making the employees more and more qualified, facilitating the practice of new technologies in the production line.

This work discusses the construction of a modelling and simulation computational environment for the training context in the industry. It approaches the problem in the design and development of a modelling and simulation environment which allows the shop floor workers to practice new work philosophies in a manufacture context.

As a project aimed at training in manufacture, it considers the context of a real factory, with the goal of understanding the domain of the application as a whole, and the posterior translation of that understanding for the design of a learning environment based on the media of the computer. It involves techniques of interfaces design based on the user-centered approach and it incorporates the main characteristics of the constructionist learning paradigm.

The work describes the process of development of the environment, discussing adopted methodologies, implementation, and informal evaluation of usability. The resulting prototype of the environment is presented together with a description of its functionality. Perspectives of continuity of the work are, finally, presented.

Agradecimentos

Eu gostaria de agradecer a várias pessoas, que graças aos seus esforços e apoio ao longo destes anos tornaram a realização deste trabalho possível.

Aos meus pais, Edson e Maria Célia, por toda dedicação e por terem abdicado de seus sonhos para tornar os sonhos de seus filhos uma realidade. Espero não decepcioná-los, retribuindo-lhes todo esse esforço.

Aos meus irmãos, Elen, Elaine, Eliana e Eduardo, aos meus avós, à Mariana e Gilberto e todos os outros familiares pelo exemplo de união, dedicação e incentivo de cada um.

À Terezinha, pelo apoio e carinho em etapas anteriores, nesta e, com certeza, em etapas futuras da minha vida. Aos seus familiares como exemplo de persistência na busca dos objetivos almejados.

À Maria Cecília, pelo exemplo de organização e profissionalismo, pois apesar de momentos difíceis, não me falhou como orientadora e amiga.

Aos meus "irmãos" Aredis e Emerson que me ajudaram a dividir por três as dificuldades e a multiplicar por três as alegrias e conquistas alcançadas.

Aos companheiros Marcos Augusto e Lúcio pelas reuniões, debates e idéias que muito enriqueceram este trabalho. E aos alunos Sérgio e Wesley pela avaliação heurística.

Ao Maurício, Ana Waleska, Raul, Neiva, Jeanne, Geraldo Magela, Paulo Drummond, Edilmar, Alexandre Mello, Dinalva†, Eustáquio e tantos outros amigos de mestrado que alegraram os dias e noites deste período.

Ao pessoal da área de conectividade, assim como outros amigos do Centro de Computação da Unicamp pela paciência e amizade.

Aos amigos do atletismo, pelas tardes de treinamentos árduos ensinando que os limites existem até que você os ultrapasse.

Aos pesquisadores e professores do Núcleo de Informática Aplicada à Educação, do Departamento de Informática da Universidade Federal de Uberlândia, do Instituto de Computação da Unicamp pelas mensagens transmitidas e pelo caminho indicado.

À Harrison Thermal Systems (GMC), pela oportunidade de interação, obtenção de informações e a realização de testes em um ambiente real de manufatura.

Ao Conselho Nacional de Pesquisa (CNPq) pelo apoio financeiro.

Conteúdo

Resumo	v
Abstract	vi
Agradecimentos	vii
1 Introdução	1
2 Objetivos e metodologia	5
2.1 Conhecimento do domínio de aplicação	5
2.2 Incorporando características de treinamento ao ambiente Enxuto	6
2.3 Análise e projeto orientado a objetos	7
2.4 Contribuições	8
3 O domínio da aplicação - A produção enxuta	10
3.1 Introdução	10
3.2 A produção artesanal	11
3.3 A produção em massa	12
3.3.1 A força de trabalho	13
3.3.2 As ferramentas	14
3.3.3 A organização	15
3.3.4 O cliente	15
3.4 A produção enxuta	16
3.4.1 A empresa como uma comunidade	16
3.4.2 O poder de decisão distribuído	17
3.4.3 O achatamento na hierarquia	17
3.4.4 O trabalho em equipe	18
3.4.5 Os equipamentos flexíveis	18
3.4.6 Um sistema orientado para o cliente: Puxar x Empurrar .	19
3.4.7 A rede de fornecedores	20

3.5	Exemplos de produção enxuta e produção em massa	21
3.5.1	Funcionamento da fábrica no sistema de produção em massa	21
3.5.2	Funcionamento de uma fábrica no sistema de produção enxuta	22
3.6	Conclusões sobre sistema de produção enxuta	23
4	Modelagem, simulação e o contexto de formação na indústria	25
4.1	A necessidade de novos métodos de treinamento	25
4.2	Modelagem e simulação em contexto de treinamento	27
4.3	Ambientes computacionais de modelagem e simulação	29
4.4	Um ambiente computacional de modelagem e simulação para treinamento em um domínio específico	34
5	Ambiente Enxuto, uma visão geral de sua funcionalidade	36
5.1	Características gerais do ambiente Enxuto	36
5.2	Descrição do ambiente Enxuto	38
5.3	Criação de modelos	42
5.4	Configuração do modelo criado	44
5.5	Execução e acompanhamento do modelo criado	48
6	Aspectos de implementação do ambiente Enxuto	57
6.1	Arquitetura do ambiente Enxuto	57
6.1.1	Modelagem	58
6.1.2	Simulação	59
6.1.3	Sistema especialista	59
6.2	Metodologia utilizada no ambiente Enxuto	60
6.2.1	Metodologia do módulo de modelagem	60
6.2.2	Metodologia do módulo de simulação	64
6.3	O módulo de modelagem	65
6.4	O módulo de simulação	66
6.4.1	Objetos da simulação	67
6.4.2	A simulação no ambiente Enxuto	68
6.4.3	Objetos da simulação no ambiente Enxuto	69
6.4.4	Distribuição Uniforme	76
6.4.5	Distribuição Exponencial	77
6.4.6	Distribuição Geométrica	79
6.5	Interação entre o módulo de modelagem e o de simulação	82

7	Um exemplo completo no ambiente Enxuto	84
7.1	Uma fábrica no sistema de produção enxuta	84
7.1.1	Cenário A	85
7.1.2	Cenário B	86
7.1.3	Cenário C	90
7.2	Interação com o sistema especialista Jonas	91
7.3	Avaliação informal de usabilidade	91
7.3.1	Avaliação do ambiente Enxuto	94
8	Discussão, conclusão e proposta de continuidade do trabalho	98
8.1	Síntese	98
8.2	O processo de design do ambiente Enxuto	100
8.3	Trabalhos futuros	101
A	Análise heurística	103
A.1	Observações do avaliador A	103
A.2	Observações do avaliador B	104
A.3	Observações do avaliador C	106
B	Diagramas da análise orientada a objetos	108
B.1	Modelo de objeto e Modelo dinâmico	108
	Bibliografia	127

Lista de Tabelas

3.1	Comparação entre os sistemas de produção, adaptado de Mazzone, 1993. .	24
7.1	Informações de entrada do ambiente Enxuto.	87
7.2	Informações de saída do ambiente Enxuto.	88
7.3	Informações de saída do ambiente Enxuto (continuação).	89
7.4	Informações de saída do ambiente Enxuto (continuação).	90

Lista de Figuras

3.1	Interação entre processos da fábrica no sistema de produção em massa. . .	22
3.2	Interação dos processos em uma fábrica no sistema de produção enxuta. . .	23
4.1	Relação entre o ciclo construcionista e um ambiente de modelagem e simulação computacional.	28
4.2	Primitivas do ambiente de modelagem e simulação Ithink.	31
4.3	Modelo de inter-relação presa e predador no Ithink.	32
4.4	Primitivas do ambiente de modelagem e simulação Extend.	33
4.5	Modelo de inter-relação presa e predador no Extend.	34
5.1	Objetos gráficos que representam os elementos do domínio de atuação do ambiente Enxuto.	37
5.2	Janela principal do ambiente Enxuto.	40
5.3	Marcadores de seleção.	43
5.4	Marcadores de conexão.	44
5.5	Nível dos elementos criados.	44
5.6	Diálogo de edição das características de entrada da Estação.	46
5.7	Diálogo de edição das características de entrada do Container.	46
5.8	Diálogo de edição das características de entrada do Cliente.	47
5.9	Diálogo de edição das características de entrada do Fornecedor.	47
5.10	Diálogo de edição das características de entrada do Estoque matéria prima / produto.	48
5.11	Diálogo de edição das características de entrada do Técnico.	48
5.12	Menu de escolha do método de produção.	49
5.13	Mensagem indicando a falta de conexão.	50
5.14	Diálogo de edição do tempo total de simulação.	50
5.15	Diálogo de visualização dos resultados de saída da Estação.	52
5.16	Diálogo de visualização dos resultados de saída do Container.	53
5.17	Diálogo de visualização dos resultados de saída do Cliente.	54
5.18	Diálogo de visualização dos resultados de saída do Fornecedor.	54

5.19	Diálogo de visualização dos resultados de saída do Estoque matéria prima / Produto.	55
5.20	Diálogo de visualização dos resultados de saída do Técnico.	56
6.1	Arquitetura do ambiente Enxuto.	57
6.2	Objetos que podem ser criados no Toolbook.	62
6.3	Hierarquia de objetos no ambiente Toolbook.	63
6.4	Interação entre os objetos na simulação orientada a relógio.	68
6.5	Interação entre os objetos na simulação orientada a eventos.	69
6.6	Diagrama de classe dos objetos que compõem uma simulação discreta e orientada a eventos.	70
6.7	Diagrama de classe dos objetos ativos que compõem uma simulação no ambiente Enxuto.	71
6.8	Diagrama de classe dos objetos passivos que compõem uma simulação no ambiente Enxuto.	71
6.9	Estrutura de dados que representa um calendário de filas de eventos. . . .	73
6.10	Localização do intervalo de ocorrência dentro do espaço amostral.	81
6.11	Interação entre o módulo de modelagem e o de simulação.	83
B.1	Modelo de objeto do ambiente Enxuto.	109
B.2	Modelo de objeto da classe base.	110
B.3	Modelo de objeto da classe objeto ativo.	111
B.4	Modelo de objeto da classe objeto passivo.	112
B.5	Modelo de objeto da classe objeto controle.	113
B.6	Modelo de objeto da classe objeto conector.	114
B.7	Modelo de objeto da classe objeto estação.	115
B.8	Modelo de objeto da classe Fábrica.	116
B.9	Modelo dinâmico do objeto Cliente.	117
B.10	Modelo dinâmico do objeto Conector Empurrar.	118
B.11	Modelo dinâmico do objeto Conector Puxar.	119
B.12	Modelo dinâmico do objeto Estação.	120
B.13	Modelo dinâmico do objeto Estação Empurrar.	121
B.14	Modelo dinâmico do objeto Estação Puxar.	122
B.15	Modelo dinâmico do objeto Estoque de Matéria Prima.	123
B.16	Modelo dinâmico do objeto Estoque Produto Empurrar.	124
B.17	Modelo dinâmico do objeto Estoque Produto Puxar.	125
B.18	Modelo dinâmico do objeto Fornecedor.	126

Capítulo 1

Introdução

Este trabalho propõe a utilização de tecnologias de modelagem e simulação computacional no contexto da manufatura fabril como apoio ao ensino/aprendizado de novos conceitos de manufatura a funcionários da área de produção, na indústria.

A introdução de novas tecnologias no processo produtivo vem, há algum tempo, alterando não somente a forma de desenvolvimento das tarefas na empresa, mas a própria maneira de se entender e organizar o trabalho. A nova organização tem implicado numa nova ordem social, onde somente trabalhadores bem qualificados terão lugar. Instrução ou treinamento nos moldes que eram feitos dentro do sistema de produção em massa já não são suficientes e não captam a necessidade de formação de pessoal das novas empresas, no chamado "sistema de produção enxuta". Ao mesmo tempo, a natureza mutante e a crescente diversidade da força de trabalho requerem novas abordagens ao treinamento, onde o novo modelo é aprender enquanto se trabalha. Portanto, é necessário buscar uma nova abordagem ao treinamento de funcionários que se apresente interessante, estimulante e que possa propiciar resultados dentro da nova visão empresarial. Expandir a visão dos funcionários para além da sua tarefa rotineira, em busca de uma visão global da participação de seu trabalho específico no processo como um todo, levantando questões do tipo como interpretar e tentar aprimorar os processos, é um desafio que pode ser facilitado pela mídia do computador.

Estando ciente desta problemática, este trabalho utilizou os conceitos de modelagem e simulação para desenvolver um ambiente computacional de modelagem e simulação que permita aos trabalhadores do "chão de fábrica" ¹ praticar estas novas filosofias de trabalho no contexto de manufatura.

¹Chão de fábrica corresponde à linha de produção de uma fábrica.

Um modelo é uma representação de um sistema real que inclui entidades, juntamente com suas interações e comportamentos [Pollacia,1989], podendo ser usado para descrever como uma atividade do mundo real irá funcionar, possibilitando testar hipóteses sobre estas, sem ter a necessidade de construí-las realmente. Modelagem é a atividade de criar o modelo a partir do uso de uma linguagem formal. A vantagem de se usar modelos é o fato de que na construção de um modelo que represente algum processo real, pode-se iniciar com uma aproximação do processo e gradualmente refiná-lo, obtendo assim um melhor entendimento dele. Este refinamento do modelo leva a aproximações boas para problemas complexos.

A simulação de um modelo envolve observar o seu comportamento ao longo do tempo. A simulação é um estudo frequentemente construído com o intuito de compreensão do comportamento do sistema, ou para avaliar efeitos de vários parâmetros e políticas operacionais através da realização de experimentos sobre estes, permitindo a validação de novas idéias, de maneira rápida e fácil [Pollacia,1989]. A técnica de simulação vem sendo utilizada em pesquisas e em áreas industriais e empresariais ao longo de vários anos, mostrando-se imprescindível na análise de soluções teóricas ou em casos que envolvem um alto risco financeiro para a sua construção com objetivos experimentais [Zuk, 1990].

O uso da simulação no contexto educacional traz vários benefícios com a exploração do domínio que está sendo simulado. Estes benefícios podem ser incrementados quando ao usuário for permitida a criação de modelos deste domínio, podendo usar o computador para criar seus próprios modelos, executar a simulação e analisar os resultados obtidos, possibilitando uma reavaliação dos modelos construídos e do próprio conhecimento sobre o domínio considerado. Além disso o uso do sistema computacional de modelagem e simulação exige que o aprendiz formalize o seu conhecimento sobre o assunto, isto faz com que ele pesquise novas informações e questione a veracidade das mesmas.

Ambientes computacionais baseados em modelagem e simulação, disponíveis comercialmente, têm sido tradicionalmente dirigidos a profissionais experientes, com competência formal em matemática. Tais ambientes podem, em geral, ser situados como ambientes de propósito geral, permitindo a modelagem em áreas diferentes do conhecimento. Pode-se citar-se nesta categoria o **Ithink** [High Performance Systems Inc.,1994] e o **Extend, 1993** [Imagine That Inc., 1993], ambos destinados a usuários no nível gerencial. Estes ambientes apresentam uma problemática quando transpostos para o contexto de treinamento em uma área de produção. A generalidade e consequente ausência de suporte a um domínio específico de trabalho pode levar o usuário a se sentir perdido com a manipulação aleatória dos vários recursos do sistema, que não fazem parte de seu cotidiano, ou pode requerer do usuário um nível de conhecimento mais profundo em modelos matemáticos [Borges et

al, 1995], o que não ocorre no mundo real do ambiente de atuação deste trabalho onde os usuários são funcionários do chão de fábrica.

Pouco consta na literatura sobre ambientes computacionais baseados em modelagem e simulação voltados para a formação de funcionários de uma linha de produção. O uso de um ambiente de modelagem e simulação voltado a um domínio específico que objetiva resolver problemas associados a esse domínio parece mais adequado do que um de propósito geral. Isso porque o usuário dentro do domínio de seu trabalho, que é composto por um conjunto bem definido de fronteiras, poderá sentir-se mais situado nesse ambiente, uma vez que estará manipulando com entidades que fazem parte diretamente do seu cotidiano.

Cientes dessa problemática, este trabalho envolve o projeto de um ambiente computacional de modelagem e simulação que permita o treinamento de funcionários do chão de fábrica em novas filosofias de manufatura, possibilitando que eles testem novas idéias e novos procedimentos de trabalho que possam vir a ser implementados em seus ambientes reais.

O ambiente computacional de modelagem e simulação proposto visa possibilitar ao usuário um nível de abstração de seu mundo real, provendo entidades e interconexões relacionadas à manufatura: estações ou células de trabalho, *containers* ou transportadores de matéria-prima/produto, estoques de matéria-prima/produto, manutenção e outras informações operacionais de uma linha de produção. Além disso, provê um meio de comunicação apropriado a não especialistas, através de interfaces gráficas e suporte à análise de resultados. Construindo e simulando o modelo da linha de produção de uma fábrica "virtual" o aprendiz é levado a pensar de forma lógica e consistente nos problemas, de forma a determinar a relação de "causa e efeito" entre suas ações e resultados.

O desenvolvimento deste ambiente faz parte de um projeto maior, denominado "Projeto de dinamização da formação e da aprendizagem nas empresas" [Nied, 1995]. Este projeto objetiva o desenvolvimento de um ambiente de aprendizado baseado no computador, juntamente com uma metodologia de formação de pessoal. A motivação para a criação deste ambiente originou-se da interação entre o NIED ² e a Harrison Thermal Systems ³, dando subsídios para o seu desenvolvimento e também permitindo uma análise mais precisa das necessidades reais que uma empresa tem a nível de treinamento de funcionários.

O Capítulo 2 apresenta os objetivos do trabalho e metodologias adotadas em seu desenvolvimento. O capítulo 3 descreve as características do domínio de aplicação - a

²Núcleo de Informática Aplicada a Educação da Unicamp.

³Subsidiária da General Motors Corporation (GMC), localizada em Piracicaba SP.

produção "enxuta" - dando uma visão geral de sua filosofia e também abordando o processo evolutivo dos sistemas de produção. O capítulo 4 apresenta a necessidade de novos métodos de treinamento para a formação de mão-de-obra para a indústria, tratando de meios de aprendizagem que aproveitam mais eficientemente a tecnologia disponível. O capítulo 5 procura caracterizar o ambiente de modelagem e simulação Enxuto do ponto de vista do usuário, apresentando o ambiente Enxuto em sua funcionalidade. No capítulo 6 há uma descrição da metodologia utilizada no desenvolvimento do ambiente Enxuto, apresentando características e aspectos de implementação. O capítulo 7 ilustra um modelo criado no ambiente Enxuto, apresentando os resultados de sua simulação, assim com uma avaliação informal de sua usabilidade. E por fim o capítulo 8 expõe conclusões e apresenta perspectivas futuras. Um apêndice contendo o resultado da avaliação heurística e um outro apêndice mostrando os diagramas da análise orientada a objetos do desenvolvimento do ambiente Enxuto são apresentados.

Capítulo 2

Objetivos e metodologia

O objetivo global deste trabalho é o *design*, projeto e desenvolvimento de um ambiente computacional para o contexto de formação na indústria.

Para alcançar este objetivo foram necessários: 1) o entendimento do domínio da aplicação como um todo, o funcionamento de uma linha de produção, os processos envolvidos, as filosofias dos sistemas de produção desde a produção artesanal, passando pela produção em massa e a tendência mundial, sistema de produção enxuta, enfocando suas necessidades em termos de formação de funcionários do chão de fábrica. 2) "tradução" do entendimento desse contexto para o *design* de um ambiente de aprendizado baseado no computador, "adequado" às necessidades contextuais levantadas.

2.1 Conhecimento do domínio de aplicação

Para a obtenção do conhecimento do domínio da aplicação fez-se necessário uma interação com uma fábrica típica. Isto foi realizado através de um projeto de cooperação existente entre NIED e a Harrison Thermal Systems no âmbito de formação de funcionários do chão de fábrica [NIED, 1995]. Este trabalho é um dos projetos de pesquisa derivados desta cooperação.

A fábrica da Harrison Thermal Systems é composta por aproximadamente cem funcionários, que possuem o segundo grau completo, alguns fazendo algum curso superior. A fábrica apresenta um balanceamento dos processos, possuindo um sincronismo entre os seus fornecedores, a linha de produção e seus clientes, utilizando a maioria das características do sistema de produção enxuta, onde podem ser observadas características importantes que devem ser abordadas em um ambiente computacional para treinamento neste contexto.

Considerando as filosofias envolvidas no sistema de produção enxuta a fábrica apresenta uma deficiência em metodologias de treinamento aos seus funcionários. Este sistema de produção requer melhorias contínuas nos processos envolvidos, implicando em treinamentos constantes dos funcionários e a maioria das metodologias de treinamento utilizadas na fábrica requer um deslocamento do pessoal da linha de produção para uma sala de treinamento, acarretando um impacto sobre a produção.

Uma das principais características do sistema de produção enxuta é aprender enquanto se trabalha. Com este intuito o ambiente Enxuto deverá ser implantado no próprio chão de fábrica, pois à medida que os funcionários vão tendo dúvidas eles podem interagir com o ambiente Enxuto para o esclarecimento das mesmas, ou podem aproveitar eventuais tempos ociosos para interagir com o ambiente.

2.2 Incorporando características de treinamento ao ambiente Enxuto

Para o desenvolvimento de um ambiente computacional destinado à formação de funcionários do chão de fábrica foi necessário o estudo e uso de técnicas de *design* de interface, com o intuito de prover um ambiente de aprendizado baseado no computador "adequado" às necessidades levantadas anteriormente. Foi utilizado *design* centrado no usuário, onde este teve uma constante participação no decorrer do desenvolvimento, objetivando mapear características do seu cotidiano, para a interface, tornando-a mais próxima de sua realidade. Com isto busca-se a construção de uma interface que seja atrativa, motivadora e de fácil manuseio. Se um sistema oferece para seus usuários características que não lhes sejam comuns, mesmo que poderosas, o sistema acaba sendo substituído por outras ferramentas mais familiares [Laurel, 1990].

Além das técnicas de *design* de interfaces um ambiente computacional destinado ao treinamento deve estar fundamentado em um paradigma educacional, definindo qual a estratégia a ser utilizada para que o usuário aprenda interagindo com o ambiente. A construção do ambiente Enxuto foi influenciada pelo paradigma construcionista de Papert [Papert, 1980]. Nessa abordagem o aprendizado é adquirido a partir da interação com os objetos do domínio sem um ensino explícito e a meta é uma exploração ativa deste domínio com o intuito de construir um conhecimento sobre o mesmo. O aprendiz construcionista é ativo no processo, sendo obrigado a questionar, buscar respostas em diferentes fontes, considerar diversas perspectivas, trocando visões com outros para construir o seu próprio conhecimento, sendo o seu esforço o principal responsável por aprender [Valente, 1993].

Este paradigma é adequado para ambientes que objetivam o treinamento no contexto de filosofias do sistemas de produção enxuta, pois este visa tornar seus profissionais cada vez mais ativos em suas tomadas de decisões.

O paradigma construcionista busca o aprendizado através de um ciclo formado pela descrição, execução, reflexão e depuração, o que pode ser alcançado por um ambiente computacional de modelagem e simulação. A etapa de descrição é realizada com a criação, de um modelo da linha de produção, a etapa de execução corresponde a própria execução da simulação, a reflexão é obtida com o acompanhamento da simulação e análise dos seus resultados e a etapa de depuração pode ser relacionada com melhorias impostas sobre o modelo. Portanto o ambiente Enxuto incorpora as principais características do paradigma construcionista.

2.3 Análise e projeto orientado a objetos

Após a definição dos tópicos a serem abordados e das características da interface do ambiente fez-se necessário o estudo e uso de técnicas de engenharia de software para projeto e desenvolvimento. Foi definido o uso da análise e projeto orientado a objeto, mais especificamente a metodologia OMT ¹ [Rumbaugh et al, 1991]. Na fase de análise foi realizado um levantamento inicial do contexto de manufatura e dos requisitos a serem supridos, resultando em um texto dissertativo, onde os substantivos são candidatos a classes de objetos e os adjetivos são os seus possíveis atributos. Com isto foi possível a identificação das classes de objetos relevantes ao domínio da aplicação e seus respectivos atributos.

O próximo passo foi a criação do modelo de objetos e do modelo dinâmico finalizando com a utilização de linguagens orientadas a objeto tanto na implementação do módulo de modelagem, quanto no módulo de simulação. Mais especificamente, a interface com o usuário foi desenvolvida usando a linguagem **OpenScript** do Toolbook [Asymetric, 1992], enquanto que no módulo de simulação utilizou-se a linguagem **C++** para a sua implementação.

Muitos desenvolvedores de software não seguem por completo as fases de uma metodologia, realizando algumas fases iniciais e indo para a fase de implementação. Com a metodologia orientada a objetos o *designer* poderá sentir-se perdido na fase de implementação caso ele não tenha seguido as fases da metodologia, principalmente se o estilo da linguagem de programação a ser utilizada for uma linguagem orientada a objetos. Ele não

¹Object Modeling Technique

saberá quais as classes de objetos a serem definidas, como é a interação entre os objetos destas classes e etc. A fase de implementação torna-se simples e automática desde que todas as outras fases anteriores tenham sido realizadas.

A programação orientada a objetos pode melhorar expressivamente os software, mas ela pode tornar uma programação obscura devido a negligência de alguns desenvolvedores. Como em qualquer ofício uma boa programação requer disciplina e a execução dos princípios do estilo a ser seguido.

2.4 Contribuições

Uma das questões que pode ser destacada no desenvolvimento do ambiente Enxuto é o *design* contextualizado, enfocando características e situações do contexto da fábrica para realizar o *design* de um ambiente de aprendizado. Isto requer um envolvimento do usuário, contribuindo com idéias, críticas e com o seu conhecimento sobre o contexto. A identificação de se o que está sendo desenvolvido está ou não de acordo com as expectativas do usuário, permite uma flexibilidade de mudanças caso sejam necessárias, sem grandes impactos no projeto. Portanto é importante que antes de iniciar uma implementação, os desenvolvedores identifiquem as necessidades e expectativas do usuário e o envolva no decorrer do projeto. Apenas realizando um levantamento inicial e depois de tudo desenvolvido apresentar o sistema ao usuário, corre-se o risco de não atender as suas expectativas, ocasionando um retrabalho que poderia ser evitado.

Um outro tópico que pode ser destacado é exemplificação do conceito de modelagem orientada a domínio [Baranauskas, Oliveira, 1995]. O desenvolvimento de um ambiente computacional de modelagem e simulação, destinado ao treinamento de um determinado domínio de atuação, está concentrado na definição de que características este ambiente deve possuir de modo a facilitar o aprendizado por parte do usuário. Isto requer que o ambiente direcione a atenção do usuário para o contexto do domínio alvo. Um ambiente de modelagem orientado a domínio, permite ao aprendiz concentrar-se em aspectos do domínio relevantes ao seu aprendizado, ao invés de se preocupar com outros aspectos que são oferecidos por um ambiente de propósito geral. Em tais ambientes o usuário terá que desenvolver conhecimentos extras para manipulação de linguagens de programação ou elementos genéricos que possuem um nível de granularidade diferente do nível de elementos mais conceituais, que devem ser usados para descrever o domínio de atuação.

Os objetivos específicos deste trabalho envolvem, portanto:

- Estudo do domínio e contexto da aplicação (manufatura fabril);

- Proposta de um ambiente computacional de modelagem e simulação para treinamento em conceitos de manufatura fabril;
- *Design* e projeto do ambiente Enxuto;
- Implementação de um protótipo do ambiente;
- Avaliação informal de usabilidade.

Capítulo 3

O domínio da aplicação - A produção enxuta

Este capítulo visa abordar o processo evolutivo dos sistemas de produção, desde o sistema de produção artesanal, descrevendo suas características e problemas que acarretaram a transição para o sistema de produção em massa. São descritos os seus pontos fortes e fracos, pois foram as suas fraquezas que acabaram se tornando fonte de inspiração para o avanço seguinte dos sistemas de produção, a produção enxuta. São examinados a sua gênese e como a produção enxuta se consolidou. No final do capítulo faz-se uma discussão comparativa entre os sistemas de produção.

3.1 Introdução

Antes de se descrever os sistemas de produção tem-se que ressaltar a importância da indústria automobilística, pois ela sempre foi a propulsora dos sistemas de produção. Ela alterou as noções mais fundamentais de como produzir bens e também mostra que a maneira como são produzidos determina, não somente o método de trabalho, mas ainda a forma de pensamento das pessoas, o que elas compram e como elas vivem [Womack,1992].

Portanto, para cada um dos sistemas de produção será descrito como eles são aplicados na indústria automobilística, embora possam ser implantados em quaisquer outros tipos de indústria.

3.2 A produção artesanal

No início deste século quando uma pessoa interessava-se em adquirir um carro, ela não se dirigia a uma revendedora de veículos, pois na época não existia, nem entrava em contato com um fabricante de automóveis, eles também não existiam. Pelo contrário, a pessoa dirigia-se a uma fábrica de máquinas e ferramentas, encomendando um automóvel [Womack,1992]. Eram fabricantes muito mais de serras para metais do que de automóveis e usavam o clássico sistema de produção artesanal, onde a força de trabalho compunha-se, na maior parte, por artesãos habilidosos, montando cuidadosamente à mão um pequeno número de carros.

Estes trabalhadores conheciam com minúcia os princípios de mecânica e os materiais com que trabalhavam. Além do mais, muitos eram seus próprios patrões, muitas vezes trabalhando como empreiteiros independentes ou, o que era ainda mais frequente, como proprietários independentes de instalações fabris às quais eram encomendados componentes ou peças específicas.

Alguns artesãos eram responsáveis, inclusive, pelos contatos com consumidores, para determinar as especificações exatas dos veículos, encomendando peças necessárias e montando o produto final. Grande parte do trabalho, inclusive projeto e engenharia, eram feitos em oficinas artesanais individuais espalhadas pela região.

Neste sistema de produção o custo unitário não decresce à medida que aumenta o volume de produção. Se a companhia fabricasse 50 mil carros idênticos a cada ano, o custo unitário provavelmente não ficaria muito abaixo do custo unitário para fazer dez carros. Ainda mais, a companhia nunca teria conseguido fazer dois carros idênticos, quanto mais 50 mil, mesmo que fossem construídos de acordo com os mesmos projetos. A razão é que os fornecedores não utilizavam um sistema de metrologia e as máquinas e ferramentas disponíveis na época eram incapazes de cortar aço de alta dureza. Portanto as peças produzidas passavam por um forno, para endurecerem suas superfícies para resistirem a uso contínuo. Entretanto, elas frequentemente entortavam no forno, necessitando de um retrabalho.

Quando essas peças finalmente chegavam ao salão de montagem, suas especificações eram na melhor das hipóteses, aproximadas. A primeira tarefa dos habilidosos montadores consistia em ajustar as primeiras duas peças até atingir a perfeição. A seguir, encaixavam a terceira peça até que se ajustasse às duas primeiras e assim sucessivamente até todo o veículo, com suas centenas de peças, estar completo.

O custo, facilidade de dirigir e manutenção simples não se incluíam entre as principais preocupações das indústrias artesanais. Mas a personalização de cada carro sim, pois seus clientes não queriam um carro qualquer, mas um construído exatamente conforme seu gosto e necessidade.

Com base em Deming [1986], a produção artesanal possuía as seguintes características:

- Uma força de trabalho altamente qualificada em projeto, operação de máquinas, ajuste e acabamento. Muitos trabalhadores progrediam através de um aprendizado abrangendo todo um conjunto de habilidades artesanais. Muitos podiam esperar administrarem suas próprias oficinas, tornando-se empreendedores autônomos.
- Organizações extremamente descentralizadas, onde a maioria das peças e grande parte de um projeto provinham de pequenas oficinas. O sistema era coordenado por um proprietário de uma oficina, em contato direto com todos os envolvidos: consumidores, empregados e fornecedores.
- Um volume de produção baixíssimo. Não conseguia produzir dois produtos finais idênticos, pois as técnicas artesanais geravam, por sua própria natureza, variações.
- Custos de produção eram elevados e não diminuíam com o volume de produção.

3.3 A produção em massa

A produção em massa foi oriunda de um conjunto de técnicas lançado por Henry Ford, que se beneficiou dos recentes avanços nas máquinas e ferramentas, que passaram a ser capazes de trabalhar com metais pré-endurecidos.

A padronização das medidas por todo o processo decorreu do fato de que Henry Ford percebeu os benefícios financeiros que resultariam nos custos de montagem. Pois, tomadas conjuntamente, a intercambialidade, simplicidade e a facilidade de ajuste proporcionariam tremendas vantagens. Por exemplo a eliminação dos ajustadores qualificados, que sempre haviam constituído o grosso da força de trabalho de montagem. É digno de nota o fato de Henry Ford ter percebido que a padronização do sistema de medida usado para todas as peças ao longo de todo o processo de fabricação, acarretaria uma intercambiabilidade das peças e na facilidade de ajustá-las entre-si [Womack,1992].

Henry Ford também introduziu plataformas de montagem, sobre as quais um carro inteiro era construído, geralmente por um só montador. Cada trabalhador montava grande

parte de um carro, antes de prosseguir para o próximo, com isso eles repetiam o mesmo conjunto de atividades em suas plataformas fixas de montagem.

Em busca de um processo mais eficiente Henry Ford buscou algumas inovações, a primeira consistia em levar as peças a cada estação de trabalho, permitindo aos montadores ficarem no mesmo local o dia todo. Depois decidiu que cada montador executaria uma única tarefa, movimentando-se de veículo para veículo através da área de montagem.

Essas medidas acarretaram um aumento considerável na produtividade, não só porque a completa familiaridade com uma só tarefa permitia ao trabalhador executá-la rapidamente, mas também porque todo o ajuste de peças havia sido eliminado. Os trabalhadores simplesmente posicionavam partes que automaticamente se ajustavam. A minuciosa divisão do trabalho e o perfeito ajuste das peças também ocasionaram uma imensa economia no processo de produção, provavelmente bem maiores que a economia do passo subsequente, a introdução de uma linha de montagem de fluxo contínuo.

Henry Ford logo reconheceu o problema trazido pela movimentação dos operários de uma plataforma de montagem para outra, onde o deslocamento demandava tempo. Logo, decidiu criar uma linha de montagem móvel, em que o carro era movimentado em direção ao trabalhador estacionário. Tal inovação diminuiu o tempo gasto para realizar um ciclo de trabalho, reduziu o esforço humano necessário para montar um automóvel, além disso quanto mais veículos produzia, mais o custo por veículo caía. As seções seguintes apresentam as principais características da produção em massa.

3.3.1 A força de trabalho

A produção em massa não se limitou a aperfeiçoar a peça intercambiável como também aperfeiçoou o operário "intercambiável". O montador da linha de montagem tinha apenas uma tarefa: ajustar duas porcas em dois parafusos ou talvez, colocar uma roda em cada carro. Não tinha que solicitar peças, ir atrás das ferramentas, reparar seu equipamento, inspecionar a qualidade ou mesmo entender o que os operários ao seu redor estavam fazendo; pelo contrário, mantinha baixa sua cabeça, pensando em outras coisas. Um trabalho mal feito só era descoberto no final da linha de montagem, onde um outro grupo de trabalhadores desempenhava seu papel, verificava a qualidade e realizava reparos necessários.

É claro que alguém tinha que pensar como todas essas peças iriam se juntar e exatamente o que cada montador deveria fazer. Essa era a tarefa de uma profissão recém-criada, a de engenheiro de produção ou engenheiro industrial. Alguns deles especializavam-se em

operações de montagem, outros em projetar maquinaria para montagem, ou em máquinas específicas para cada peça. Alguns engenheiros de produção especializaram-se em motores, outros em carrocerias e outros ainda em suspensão ou sistemas elétricos. Eles eram "trabalhadores do conhecimento", indivíduos que manipulavam idéias e informações, mas raramente tinham contato com um automóvel concreto e quase nunca entravam na fábrica [Womack,1992].

Os trabalhadores deste sistema de produção não davam, voluntariamente, qualquer informação sobre as condições operacionais e muito menos sugeriam meios de aperfeiçoar os processos. Estas funções foram atribuídas respectivamente ao supervisor e ao engenheiro de produção. O trabalhador do chão de fábrica não tinha uma carreira promissora pela frente, podendo no máximo chegar a supervisor. Já para os engenheiros, havia uma carreira a ser escalada, desde estagiário até sênior, dominando todo o conhecimento de sua profissão, onde encarregava-se em coordenar os engenheiros dos escalões inferiores. Alcançar o topo da profissão de engenheiro significava muitas vezes mudar de uma empresa para outra no decorrer da vida profissional.

Por fim, o sistema de produção em massa considerava a força de trabalho como um custo variável, não hesitando em demitir os trabalhadores ao primeiro sinal de queda nas vendas.

3.3.2 As ferramentas

Na produção de peças sob medidas padronizadas, o baixo custo só seria propiciado por ferramentas que desempenhavam tal tarefa em grandes volumes e com baixo ou nenhum custo de preparação de máquinas. Mas máquinas genéricas que produzem duas ou mais peças, têm um gasto de tempo de preparação, quando estão aptas a produzir uma determinada peça e se deseja que passem a produzir um outro tipo de peça. Portanto os engenheiros projetavam máquinas capazes de realizar uma só tarefa de cada vez e aperfeiçoaram instrumentos e suportes que prendiam a peça a ser trabalhada à máquina. Ao trabalhador desqualificado bastava pegar a peça e apertar um botão ou empurrar uma alavanca para a máquina executar a tarefa necessária.

Com isto, estas máquinas podiam ser dispostas numa sequência de modo que cada passo de fabricação é seguido imediatamente do próximo, dando a impressão de que a fábrica era realmente uma imensa máquina, com cada passo da produção intimamente ligado ao subsequente. A única desvantagem desse sistema era a inflexibilidade; adaptar tais máquinas dedicadas para uma nova tarefa consumia tempo e dinheiro.

3.3.3 A organização

As indústrias no sistema de produção em massa são "inchadas", no que diz respeito a fabricação de todas as partes que compõem o produto final. Todos ou a grande maioria dos componentes de um produto são de fabricação própria. O restante dos componentes e alguns serviços eram encomendados a empresas independentes sem qualquer relação com o comprador. As transações eram baseadas no preço, prazo de entrega e qualidade, sem qualquer expectativa de relacionamento a longo prazo ou contínuo com seus fornecedores.

A administração de uma indústria na produção em massa pode ser considerada como gerenciamento pelos "números", onde um grupo de executivos solicita com frequência relatórios detalhados sobre as vendas, participação no mercado, estoques, bem como lucro e perdas e orçamentos reservados de capital. Caso estes números exibissem mau desempenho, estava na hora de trocar de gerente geral. Caso contrário, se mostrassem consistentemente bons resultados eram candidatos potenciais para promoção. Com isto novas profissões de gerente financeiro e especialistas em *marketing* foram criadas para complementar as profissões de engenheiro, de modo que toda a área funcional passou a ter seu especialista.

3.3.4 O cliente

A produção em massa solucionou o conflito entre a necessidade de padronização, para reduzir custos de fabricação e a diversidade de modelos, exigida pela variedade da demanda dos consumidores da seguinte forma: usando novamente a indústria automobilística como exemplo, ela padronizou vários itens mecânicos para todos seus modelos; ao mesmo tempo, alterava anualmente a aparência externa e introduzia uma série enorme de acessórios, como transmissões automáticas, sistemas de ar condicionado, rádio, possíveis de serem instalados nos modelos existentes para sustentar o interesse dos consumidores. Um outro fato é que o pessoal de vendas pouco se interessa pelas necessidades ou desejos dos clientes. Eles querem fechar logo o negócio, restringindo as informações sobre o produto ao que lhes interessa para alcançar tal meta. Uma vez feito o negócio, o vendedor perde o interesse pelo cliente. Todo o sistema de vendas e negociações se baseia em fornecer ao cliente o mínimo de informações possíveis.

Uma outra característica marcante da produção em massa é o elevado nível de estoque em todas as etapas de produção. Próximas de cada estação de trabalho tem-se pilhas de peças. No final da linha, depara-se com o melhor indício da produção em massa, uma enorme área de trabalho atulhada de produtos acabados, cheios de defeitos, necessitando

de reparos antes de serem expedidos. Também há estoques sobressalentes de peças no almoxarifado e uma grande armazenagem de produtos prontos.

3.4 A produção enxuta

O objetivo principal da produção "enxuta" ¹ é acabar com o desperdício de trabalho, energia, tempo, dinheiro e materiais. Envolve, em cada momento, a preocupação de reduzir o custo ao mínimo absolutamente necessário.

A filosofia da produção enxuta em boa parte nasceu do trabalho em métodos de controle de qualidade do engenheiro americano W. Edwards Deming, que foi consultor das forças de ocupação dos Estados Unidos no Japão logo após a Segunda Guerra. Deming argumentava que a produtividade não era necessariamente diminuída pela ênfase na qualidade. Ao contrário, Deming discutia com os industriais japoneses, que um sistema industrial organizado para produzir qualidade poderia apoiar e aumentar a produtividade [Deming, 1986]

Estas idéias foram adotadas com mais sucesso no Japão pela montadora Toyota nos anos 50. Mas desde o início o engenheiro chefe e vice presidente da Toyota, Taiichi Ohno e sua equipe de engenharia desenvolveram a sua própria versão das estratégias de controle de qualidade de Deming, superando com muita criatividade vários tabus do sistema de produção em massa e introduziram outras estratégias especificamente japonesas [Moura, 1989].

Uma outra característica da produção enxuta é o *Kaizen* que em japonês significa "melhoramento contínuo". Uma das diferenças fundamentais entre o pensamento ocidental e o oriental é a ênfase do segundo na maneira como as coisas são feitas, isto é, o sistema ou processo, enquanto que no ocidente o resultado final é considerado mais importante. Isto estimula todos os funcionários de uma empresa, desde o presidente até os operários de linha, a procurarem constantemente, melhorar o que fazem e o que está ao redor deles, não somente no trabalho mas também em casa. A seguir serão descritas algumas características fundamentais do sistema de produção enxuta.

3.4.1 A empresa como uma comunidade

O fator mais importante que viabiliza a produção enxuta é o emprego vitalício que é uma promessa dada pela empresa com base em um acordo não escrito de obrigações

¹Enxuta é a tradução da palavra inglesa "lean" que significa, magra, sem gordura.

recíprocas, onde os funcionários se comprometem totalmente com a empresa, dedicando-se a suas tarefas com mais energia e responsabilidade, inclusive aceitando encargos com poder de decisão que, normalmente, seriam exclusivos da gerência. Com isto tornam-se flexíveis para a atribuição de tarefas e ativos na promoção dos interesses da empresa, introduzindo melhoramentos, em vez de apenas reagirem aos problemas.

O pensamento da empresa com este compromisso é o de aproveitar ao máximo seus recursos humanos, desde o momento em que um novo trabalhador ingresse na empresa, até que atinja a aposentadoria, aproveitando as qualificações dos trabalhadores, seus conhecimentos e experiências e não só a sua força física. Assim há um reconhecimento de uma interdependência entre gerência e trabalhadores.

3.4.2 O poder de decisão distribuído

Na linha de montagem enxuta o poder de decisão é dado para aqueles que estão mais próximos dos problemas. Há uma corda ao alcance de cada operário da linha que, se puxada, pára a linha toda de produção. Os operários têm a responsabilidade de apontar todo e qualquer defeito na hora em que o identifiquem ou, mesmo antes, quando há suspeita de um problema. Esta é uma das regras básicas do sistema de produção enxuta que a diferencia da linha de montagem da produção em massa. As peças ou montagens com defeitos não são colocadas de lado para posterior conserto ou mandadas para frente para correção pelo controle de qualidade; a correção é feita na hora e também a identificação da origem do problema. A idéia básica é que somente montagens livres de defeitos podem seguir para frente.

Esta busca constante da perfeição na linha de produção tem uma influência importante na política do poder de decisão, pois os operários têm, de fato, mais possibilidades imediatas de detectar imperfeições e assegurar nível de qualidade, isto é, de acrescentar valor ao produto. Desta maneira um forte relacionamento de confiança mútua tende a crescer entre os operários, onde a única preocupação é a perfeição do produto e a necessidade de evitar defeitos e custosas interrupções no andamento da linha.

3.4.3 O achatamento na hierarquia

Para viabilizar as duas primeiras características descritas anteriormente e de uma certa forma, como resultados delas, há um achatamento da pirâmide hierárquica da empresa em termos de responsabilidades, prestígio, poder e salário mais alto e a média de salários é bem menor do que em uma fábrica tradicional de produção em massa.

No sistema de produção enxuta não é somente a diferença salarial que é achatada, os privilégios de executivos são reduzidos a um mínimo. Não há refeitórios executivos, abonos gordos ou outras vantagens que diferenciam os executivos dos operários da linha de produção.

3.4.4 O trabalho em equipe

Todo o trabalho em uma fábrica de produção enxuta é feito por equipes. Há equipes de usinagem, outra de montagem de uma parte do produto e assim por diante. Mas há um intercâmbio de membros entre as equipes, com o intuito de estimular os operários a buscar novos conhecimentos e também a obterem um conhecimento do processo global. Quando um defeito aparece o responsável é toda a equipe. Se um membro da equipe atrasa, falta ou simplesmente trabalha mal, toda a equipe sofre. Assim há um esforço máximo para não prejudicar os companheiros. Em algumas fábricas a equipe tem a palavra final na aceitação de novos membros ou nas demissões.

O sistema de produção enxuta destaca a ação cooperativa, formando círculos de qualidade para atacar problemas e melhorar os métodos de produção em todos os níveis. Membros de várias equipes desenvolvem um trabalho contínuo para encontrar meios de melhorar a qualidade, reduzir defeitos, acelerar a produção e diminuir custos. Há também incentivo à criatividade e inteligência de seus trabalhadores e com este objetivo em mente são oferecidos um grande número de oportunidades de treinamento, aperfeiçoamento e desenvolvimento pessoal.

3.4.5 Os equipamentos flexíveis

Uma das características mais marcantes do sistema de produção enxuta é o desenvolvimento de ferramentas flexíveis. No início do desenvolvimento da produção enxuta um dos principais obstáculos encontrados pelos seus idealizadores nas linhas de montagem, eram a presença de prensas e máquinas que não podiam ter seus padrões modificados sem ajustes que, às vezes, demoravam vários dias. Depois de persistente experimentação, conseguiram reduzir para minutos o tempo necessário para modificar estas ferramentas.

Com esta flexibilidade consegue-se produzir pequenos lotes de peças para determinados modelos, o que implica em um número menor de defeitos e redução no desperdício de peças, pois as máquinas podem ser reajustadas imediatamente quando surge algum defeito, acarretando assim uma redução no custo de produção. Isto contraria uma das leis mais sagradas da produção em massa, que diz que quanto maior é o lote produção, menor

é o custo de cada unidade. Também com o aumento da flexibilidade das ferramentas, obtém-se uma maior capacidade de atender a pedidos menores e específicos.

3.4.6 Um sistema orientado para o cliente: Puxar x Empurrar

No sistema de produção enxuta, o cliente faz parte da equipe e é colocado no começo do ciclo de produção e não no fim deste ciclo, como nos modelos tradicionais de produção em massa. Os desejos do cliente, suas sugestões, reclamações, problemas, são consideradas com seriedade por todos em toda a linha de produção, tendo uma influência decisiva no fluxo de produção.

O sistema de produção enxuta determina o fluxo de produção através de um cartão denominado *kanban* que identifica cada lote de peças ou montagens e o setor ou o operário que os produziu. Com o uso do cartão *Kanban* cada operário apanha as peças que necessita em pontos de fornecimento denominados de "supermercados" e deixa o cartão *kanban* num quadro [Moura, 1989]. Cada setor de produção, por sua vez, somente produz e coloca novas peças no "supermercado" quando há um cartão no quadro assinalando que as peças já produzidas foram usadas. Este sistema evita a superprodução e simplesmente atende à demanda dos clientes no começo do processo, nos pontos de venda. Assim, é o cliente quem "puxa" a produção.

O sistema de produção puxar, é um sistema revolucionário no sentido de que o último processo da linha de produção retira matéria-prima do processo anterior, pois somente ele pode saber com exatidão o tempo e quantidade de matéria-prima necessários para produzir uma determinada requisição de produto. O penúltimo processo, por sua vez, vai ao processo anterior a ele, para obter a matéria prima necessária para poder realizar sua produção repondo o que foi retirado pelo último processo. E assim sucessivamente até atingir o início da linha de produção. O processo atua como cliente do processo anterior a ele, requisitando matéria-prima, "puxando" a produção, sendo a base do sistema puxar.

No sistema tradicional de controle de produção, as quantidades necessárias e o tempo gasto são determinadas preparando-se programações de produção para todos processos. O primeiro processo da linha de montagem, obtém a matéria prima necessária para realizar sua produção de acordo com um programa e passa o seu produto para o processo seguinte, que o utilizará como matéria-prima para realizar a sua produção, considerando a quantidade de matéria-prima disponível e a informação contida na sua programação de produção pré-estabelecida. Ao término de sua produção ele a passa para o processo seguinte, onde toda essa sequência se repetirá até atingir o último processo da linha de produção. Portanto um processo "empurra" sua produção para o próximo processo, formando assim o sistema de produção empurrar.

No sistema de produção em massa, as metas de produção são calculadas e anunciadas pela direção com base em dados estatísticos e pesquisas de mercado e então "empurrada" para os clientes. A mesma filosofia é usada para determinar o tipo ou estilo do produto a ser fabricado. Uma vez ponderado o quê o cliente deve desejar é organizada uma campanha publicitária para convencer os compradores em potencial de que o quê será produzido é o que eles devem querer. No sistema de produção enxuta, custo e estilo são determinados nos pontos de vendas do sistema depois de amplas consultas com os clientes. De acordo com Womack et al:

"O revendedor tornou-se parte do sistema de produção, à medida que a Toyota foi parando de construir carros antecipadamente para compradores desconhecidos, convertendo-se para sistema de encomendas, onde o revendedor era o primeiro passo do sistema kanban, enviando pedidos de carros previamente vendidos a determinados clientes, para entrega em duas ou três semanas" [Womack, 1992; p. 57].

3.4.7 A rede de fornecedores

Os fornecedores e a fábrica terminal no sistema enxuto têm um relacionamento de total confiança mútua. Não há segredos de um lado ou de outro. É um relacionamento entre colaboradores e não entre adversários. O fabricante terminal tenta sempre garantir um nível relativamente constante de pedidos, mesmo na época de recessão quando as vendas diminuem. Por outro lado os fornecedores são confiáveis, eficientes e flexíveis, garantindo ao fabricante terminal preços estáveis mesmo em épocas de muitas vendas. Este relacionamento estreito e de confiança, possibilita a implantação de sistemas de entregas *just-in-time* cortando a necessidade de grandes estoques, que por sua vez ajuda a baixar os custos de produção e o preço final do produto. *Just-in-time* é o processo em que o número exato de unidades de matéria-prima é entregue pelo fornecedor na fábrica terminal poucas horas antes de seu uso na linha de produção. Por esta razão, para que funcione adequadamente, é necessário que não haja ou haja o mínimo possível de unidades com defeito e poucas falhas no que se refere às entregas, pois a fábrica fica bastante vulnerável a paradas nas linhas por falta de matéria-prima.

Já no sistema de produção em massa o fabricante terminal esconde informações sobre seus novos produtos e métodos de produção, mas o fornecedor faz a mesma coisa. Quando há uma queda nas vendas o fabricante imediatamente reduz os seus pedidos. Ao mesmo tempo, para se protegerem e se prepararem para recessões, os fornecedores aumentam os seus preços ao máximo possível quando a demanda é grande.

3.5 Exemplos de produção enxuta e produção em massa

Para efeito de comparação entre os sistemas de produção é apresentada na tabela 3.1 uma síntese das suas principais características. Como ilustração do desenvolvimento dos processos em ambos os sistemas de produção (em massa e enxuta), desde o pedido feito pelo cliente, passando pelos processos de produção do chão de fábrica, até o retorno do produto final ao cliente, será utilizada uma fábrica hipotética de produção de canetas. A composição de cada caneta é dada por uma tampa frontal, carga, corpo e uma tampa traseira. A seguir será descrito o funcionamento desta fábrica tanto na forma de produção em massa quanto na forma de produção enxuta.

3.5.1 Funcionamento da fábrica no sistema de produção em massa

Em fábricas que utilizam o sistema de produção em massa, os processos acontecem da seguinte forma: o cliente faz o pedido de produtos junto ao setor de expedição da fábrica, podendo ser atendido imediatamente caso haja produto em estoque ou tendo que aguardar por um determinado período caso o produto esteja em falta. No setor de expedição uma ordem de produção é gerada segundo uma análise da quantidade em estoque, das quantidades requisitadas nos pedidos pendentes e mais uma quantidade adicional, denominada de margem de segurança. Esta ordem de produção é passada para a gerência de produção onde ela será escalonada de acordo com uma sequência de prioridade, para posteriormente ser repassada para o setor de almoxarifado. Neste ponto os conjuntos de matéria prima que são necessários para a realização de cada operação são entregues nas respectivas estações.

O processo de produção inicia quando a ordem de produção é recebida pela primeira estação no chão de fábrica, onde operações são aplicadas sobre o seu conjunto de matéria prima, resultando em um novo item de matéria prima, que juntamente com a ordem de produção são passadas para a próxima estação, que irá realizar as suas operações sobre o seu conjunto de matéria prima e repassar seu resultado para a estação subsequente no chão de fábrica. Assim transcorre sucessivamente até atingir a última estação onde o resultado final será o produto acabado, podendo ser encaminhado ao setor de expedição, para posterior entrega junto ao cliente. A figura 3.1 ilustra a interação entre os setores da fábrica utilizando o sistema de produção em massa.

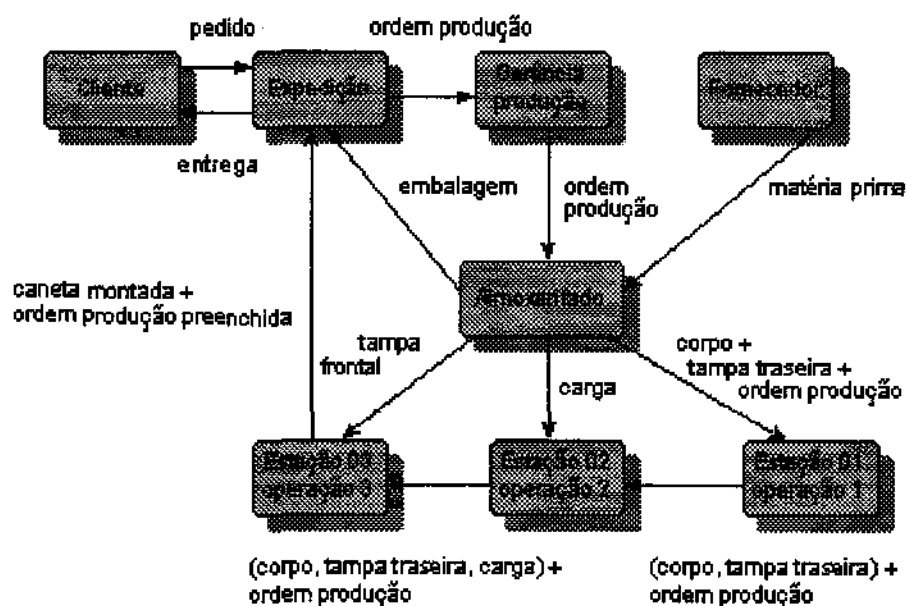


Figura 3.1: Interação entre processos da fábrica no sistema de produção em massa.

3.5.2 Funcionamento de uma fábrica no sistema de produção enxuta

Nas fábricas que utilizam o sistema de produção enxuta, o cliente realiza um pedido (encomenda) de produtos para o setor de expedição da fábrica. O setor de expedição será responsável por "puxar" a produção, controlando o ritmo de produção da fábrica, de acordo com a demanda dos clientes. Ele verifica se tem lote de produto acabado em seu cartão *kanban* de requisição, caso haja ele transfere o lote de produtos acabados para o lote de venda. O cartão *kanban* de requisição torna-se vazio, que é o mesmo cartão de produção da estação final da linha de montagem. Essa estação verifica que o seu cartão *kanban* de produção está vazio e analisa se possui lote de matéria prima em seu cartão de requisição, caso não haja a estação fica parada esperando que a estação anterior disponibilize-as. Caso contrário ela começa a produzir utilizando matéria prima que o fornecedor disponibilizou, mais todo o lote contido em seu cartão *kanban* de requisição, tornando-o vazio. Esse cartão é o mesmo cartão *kanban* de produção da estação anterior, a qual verifica que está vazio e tenta produzir. Este procedimento é repetido por todas as estações anteriores até atingir a primeira estação da linha de produção. A figura 3.2 ilustra a interação entre os setores de uma fábrica utilizando o sistema de produção enxuta.

	Produção artesanal	Produção em massa	Produção enxuta
Profissionais	multiqualificados	especializados	multiqualificados
Ferramentas	flexíveis	específicas	flexíveis
Produtos	por encomenda	padronizados	por encomenda
Qualidade	alta	"aceitável"	alta
Custo	alto	baixo	baixo
Relação fornecedor/empresa	estável	frio	confiança mútua
Identificação e correção defeitos	ponto da linha onde aparece	no final da linha de produção	ponto da linha onde aparece
Poder de decisão	distribuído	centralizado	distribuído
Relação operário/empresa	estável	instável	vitalícia
Relação cliente/empresa	momentânea	momentânea	a longo prazo
Volume de produção	baixo, sob demanda	alto, conforme estatísticas	alto, sob demanda
Nível de estoque	baixo	alto	baixo
Organização	descentralizada	centralizada	descentralizada
Ênfase	no produto	no produto	no processo
Organização do trabalho	individual	individual	equipe

Tabela 3.1: Comparação entre os sistemas de produção, adaptado de Mazzone, 1993.

tudo o que não acrescente valor ao produto. O resultado é a produção de bens altamente competitivos em termos de preço e qualidade. Mas para alcançar este objetivo tem que ser oferecido a estes funcionários oportunidades de treinamento, que os conduzam a um melhoramento contínuo e que incentive o aperfeiçoamento por parte de todos. A visão de treinamento é direcionada para um conceito mais amplo de "formação na empresa", onde as organizações buscam métodos de treinamentos nos quais os funcionários aprendem fazendo. Portanto serão necessários novos meios de ensino/aprendizagem que terão que ser desenvolvidos, aproveitando eficientemente a tecnologia disponível. Isto motivou a realização deste trabalho, onde foi realizado um estudo do sistema de produção enxuta e o desenvolvimento de um protótipo de um ambiente destinado ao funcionário de chão de fábrica abordando conceitos da produção enxuta. Esse ambiente de modelagem e simulação foi denominado "Enxuto".

Capítulo 4

Modelagem, simulação e o contexto de formação na indústria

Este capítulo investiga a necessidade de novos métodos de treinamentos para a formação de mão-de-obra para a indústria. Aborda meios de aprendizagem que aproveitam mais eficientemente a tecnologia disponível e propõe o uso de modelagem e simulação como método para uma melhor qualificação do aprendiz no contexto das necessidades dos novos sistemas de produção.

4.1 A necessidade de novos métodos de treinamento

O surgimento de novos processos de trabalho, manufatura e comercialização têm permitido um grau de qualidade, produtividade e rapidez nunca antes atingido [Mazzone, 1993]. As organizações têm procurado empregar menos gente para fazer um determinado trabalho, com uma qualificação mais completa, pois se a fragmentação do trabalho é evitada e há um esforço para integrar várias atividades nas mãos de um único funcionário, este tem que ser um generalista ao invés de um especialista [Mazzone, 1995]. Portanto os métodos de instrução ou treinamento precisam ser reestruturados para atender a necessidade de formação de pessoal destas novas organizações. As empresas estão associando aprendizado à produtividade e denominando de aprendizado *just-in-time* [Reinhardt, 1995] ao invés de treinar antes do trabalho. As pessoas aprendem uma habilidade no momento em que precisam dela.

O treinamento deve ser totalmente remodelado, o funcionário necessita de treinamento no local de trabalho para aprender a conhecer a empresa, desde a recepção de matéria-prima até o cliente, tendo uma visão global da produção, obtendo assim consciência da

importância de seu trabalho no âmbito da organização. O sistema de produção enxuta tende a formar uma classe de executivos e operários com uma visão mais abrangente da empresa, seus produtos e os processos de trabalho. Portanto torna-se uma necessidade nas organizações empresariais preparar o indivíduo para o novo papel, mais ativo nas tomadas de decisão em participação direta no processo de produção.

O sistema educacional, responsável pela formação mais ampla do indivíduo, não consegue acompanhar o ritmo das mudanças tecnológicas e do processo que têm acontecido no sistema produtivo. O segmento da educação que cresce mais rapidamente está dentro da indústria, que enfrenta a questão de como fazer com que a força de trabalho acompanhe seu passo [Reinhardt, 1995]. Novos métodos de aprendizagem que utilizam a tecnologia de computadores podem ser a resposta para esta deficiência. Muitas organizações estão utilizando métodos conhecidos como "treinamento conforme demanda", consistindo em disponibilizar aos seus funcionários computadores localizados próximos ao local de trabalho, tendo livre acesso e podendo ser consultados no momento de uma dúvida ou de algum problema [Reinhardt, 1995]. Estes computadores podem conter CD-ROM explicativos, sistemas especialistas integrados à área de trabalho, uma base de informações constantemente atualizada ou software de modelagem e simulação que permitem ao funcionário criar uma abstração de seu mundo real para poder auxiliar na resolução de sua dúvida ou problema.

Esta nova abordagem de treinamento evita os problemas causados pelo treinamento em sala de aula como, por exemplo, constrangimentos de alunos: o computador deixa que ele faça tudo sem receio de cometer erros. Não há a necessidade de interromper o tempo de produção dos funcionários para realização do treinamento, não sendo necessário deslocá-los até uma sala de aula. As implicações dessa transformação afetam tanto os alunos quanto os professores. Os professores passam de instrutores para facilitadores, atuando como um guia enquanto que os alunos passam a ter atitudes de aprendiz, ficando livres para construir o conhecimento.

As organizações brasileiras investem em treinamento menos que 1% das horas trabalhadas durante o ano, por empregado. É de 6% a média mundial, enquanto que na indústria japonesa os empregados passam em média, 10% do tempo em treinamento [Folha de S. Paulo, Sebrae, 1994]. Este índice pode ter como sua principal fonte a carência de metodologias, ferramentas e ambientes computacionais adequados à formação de recursos humanos no país. Portanto é necessário buscar uma nova abordagem ao treinamento de funcionários de acordo com o novo contexto de formação na indústria que se apresente interessante, estimulante e que possa propiciar resultados dentro da nova visão empresarial. Expandir a visão dos funcionários para além do domínio de atuação durante o

desenvolvimento de suas tarefas, em busca de uma visão global da participação de seu trabalho específico no processo como um todo, levantando questões do tipo como interpretar e tentar aprimorar os processos, é o desafio que pode ser facilitado pela mídia do computador.

4.2 Modelagem e simulação em contexto de treinamento

As definições de simulação encontradas na literatura, em sua maioria tentam descrever operações que podem ser realizadas e os objetivos que podem ser alcançados com o uso da simulação sobre um determinado sistema ou entidade. Dentre elas pode-se destacar a de Shubik, que define a simulação de um sistema como a operação de um modelo que representa esse sistema: *"O modelo é passível de manipulações que seriam difíceis de levar a cabo na entidade que ele representa, quer pelo preço, quer pela impraticabilidade, ou impossibilidade de fazê-las. As propriedades concernentes ao comportamento de um sistema ou sub-sistema podem ser inferidas estudando-se a operação do modelo"* [Shubik,1960].

Tem sido bastante difundido o uso de simulação em contextos educacionais. Segundo a American Association for Artificial Intelligence, em relatório sobre sistemas inteligentes do próximo século, "simulação inteligente" é uma das quatro áreas apontadas como sendo "sistema de aplicações de alto impacto" [Graszy, 1994]. Segundo a associação, esses sistemas podem proporcionar formação e treinamento extensos e ricos em recursos disponíveis em quaisquer tempo e lugar.

Os benefícios do uso de simulação em contextos educacionais já são bastante conhecidos [De Jong, 1991; Pagano, 1992; Hebensbreit, 1991]. Tais benefícios ganham uma nova dimensão se for dada ao usuário da simulação a possibilidade de criação de modelos que ele próprio deseja simular. Uma das técnicas que permite a criação de modelos é a modelagem computacional, que é a atividade de usar o computador para expressar o modelo, com o objetivo subsequente de explorar possíveis consequências do modelo e reavaliar, a partir de um *feedback*, não apenas o modelo construído, mas o próprio conhecimento sobre o sistema ou fenômeno alvo [Baranauskas, Oliveira, 1994].

Sistemas computacionais para modelagem e simulação podem constituir ambientes de aprendizado poderosos, por envolver o aprendiz no ciclo básico de expressão, avaliação e reflexão sobre o domínio considerado. A exigência do computador pela expressão formal de um modelo leva o aprendiz a definir mais precisamente seu conhecimento sobre o assunto. Além disso, a execução do modelo no computador possibilita uma avaliação que

pode motivar o usuário a questionar o modelo, reavaliar seu conhecimento e reexpressá-lo continuando um ciclo de ações, ao estilo construcionista de aprendizagem [Papert, 1980]. Portanto pode-se destacar a relação entre as etapas do ciclo construcionista e as fases de um ambiente computacional de modelagem e simulação:

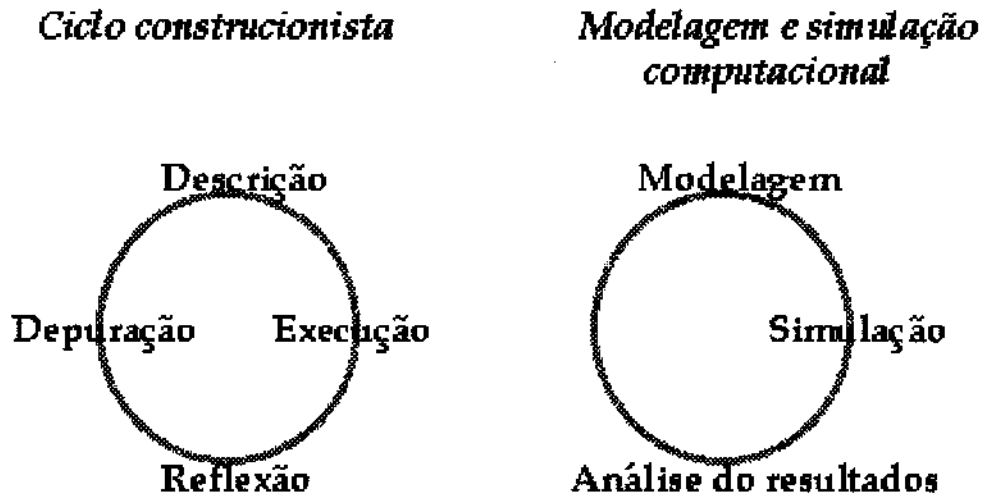


Figura 4.1: Relação entre o ciclo construcionista e um ambiente de modelagem e simulação computacional.

A etapa de **descrição** corresponde a fase de **modelagem**, onde o usuário cria o seu modelo. A etapa de **execução** é realizada pela fase de **simulação**. Já a **reflexão** é obtida através da **análise dos resultados** via ferramentas de *feedback* que o ambiente computacional provê. E por fim a etapa de **depuração** está relacionada com a fase de **modelagem** onde o usuário pode remodelar ou refinar o seu modelo, fechando assim, um ciclo.

Os sistemas computacionais de modelagem e simulação também abrangem as principais características das novas formas de aprender, onde eles podem funcionar como mentores eletrônicos, assumindo o papel de um consultor, enquanto que seus usuários passam a ter um comportamento de aprendiz. Isso permite e torna economicamente viável o retorno de um método educacional muito antigo no contexto de formação profissional "mestre-aprendiz", onde era comum pessoas mais experientes (mestre), que possuíam uma profissão definida, "adotarem" outras pessoas inexperientes (aprendiz) e os ensinarem os ofícios da profissão. O método do mestre-aprendiz sempre foi eficiente no contexto da indústria, quer seja através de uma outra pessoa ou através de simulações [Reinhardt,1995]. Os sistemas computacionais de modelagem e simulação permitem o uso

desse método com sucesso em campos nos quais é difícil ou impossível fazê-lo na vida real, como em cirurgias ou pilotagem de aviões.

A modelagem e simulação como uma ferramenta poderosa na resolução de problemas e apoio a tomada de decisão não se restringe apenas ao contexto de formação de recursos humanos mas, atua também no próprio contexto da fábrica como um todo. Várias justificativas podem ser dadas para essa afirmação. Entre elas podemos citar:

- Modelagem e simulação fornecem estimativas de *performance* de um sistema para novas estratégias, parâmetros ou condições operacionais;
- Projetos alternativos podem ser comparados e avaliados;
- Modelagem e simulação permitem experimentar sem perturbar o sistema real;
- O sistema pode ser estudado através da identificação e manipulação das suas variáveis e avaliação da influência que cada uma exerce sobre o sistema.

A modelagem e simulação também podem ser indicadas para o tratamento de outros problemas, onde manipulações por métodos tradicionais tornam-se difíceis [Exame, 1995]. A seguir é apresentada uma lista de problemas para os quais modelagem e simulação são indicadas como forma de análise:

- Se o desenvolvimento de um modelo matemático é muito difícil ou, talvez, impossível;
- Se o sistema tem uma ou mais variáveis aleatórias;
- Se a dinâmica do processo é extremamente complexa;
- Se o objetivo é observar o comportamento de um sistema em um determinado período de tempo;
- Se o uso de animação é importante para visualizar o processo.

4.3 Ambientes computacionais de modelagem e simulação

Um ambiente para modelagem e simulação, em geral, provê suporte às fases de criação do modelo de algum sistema (modelagem), assim como execução (simulação), análise e

interpretação dos resultados. A criação do modelo emprega alguma linguagem formal para sua representação. Esta linguagem formal, consiste de um conjunto de símbolos, juntamente com um conjunto de regras para combiná-los. Estes símbolos constituem as primitivas de modelagem, a partir das quais modelos podem ser construídos. Este conjunto de primitivas formam um vocabulário que pode incluir elementos simbólicos. O ambiente de modelagem e simulação proposto neste trabalho provê o suprimento de tais elementos para o domínio de manufatura.

Uma vez estabelecidos os componentes e relações entre os elementos de um modelo, tem início a fase de simulação, onde os valores das variáveis do modelo são definidas, assim como a realização da execução do modelo para uma posterior análise e interpretação dos resultados.

Estão disponíveis comercialmente vários tipos de ambientes computacionais para modelagem e simulação, que vão desde ambientes puramente de simulação até ambientes de modelagem e simulação de propósito geral. Os primeiros são em sua maioria programas desenvolvidos em linguagens de programação, como **Fortran**, **C** e etc, onde o usuário está limitado a alterar valores de variáveis de um modelo específico e observar os resultados finais retornados pelo ambiente, não podendo realizar a atividade de modelagem propriamente dita, não tendo nenhum controle sobre a lógica do modelo. Como consequência o nível de aprendizado obtido pelos usuários deste tipo de ambiente pode não levar o aprendiz a questionar, ou avaliar o modelo considerado, bem como seu próprio conhecimento sobre o assunto alvo.

Já os ambientes de modelagem e simulação de propósito geral dão uma maior autonomia ao usuário, permitindo que ele crie o seu próprio modelo e direcione a simulação de acordo com suas necessidades. Estes ambientes também permitem a realização de modelagem e simulação em diferentes domínios do conhecimento. Nesta categoria incluem-se software tais como **Ithink** [High Performance Systems, 1994] e **Extend** [Imagine That Inc., 1993]. Ambos permitem a realização de modelagem e simulação em uma variedade de domínios, como por exemplo:

- Na matemática: resolver numericamente equações diferenciais;
- Na hidrologia: os efeitos da evaporação, chuva, condensação e transpiração de plantas sobre um volume de água de um lago;
- Na biologia o crescimento de uma planta, a descarga de um neurônio, as batidas do coração, poluição de um lago, inter-relação entre presa e predador;
- Na geografia: crescimento populacional;

- Na economia ou no comércio: efeito da propaganda na venda de um produto, as filas típicas de um sistema de atendimento bancário;
- Em processos industriais: linha de montagem de um produto, sistemas de distribuição.

No software Ithink os modelos são uma composição de diagramas e fórmulas algébricas construídas em dois níveis. Em um nível superior, as estruturas do modelo são descritas por diagramas em termos de primitivas representando tanques (ou acumuladores), fluxos e conversores que podem ser unidos através de conectores [High Performace Systems,1994]. Em um segundo nível, a partir de um duplo *click* em cada um dos elementos envolvidos no modelo, são definidas relações algébricas entre eles. Cada primitiva tem um ícone correspondente associado, sendo que o diagrama pode ser composto a partir de ações com o *mouse*, figura 4.2-a. Uma primitiva do tipo tanque possui um valor inicial e é capaz de acumular valores quando a ela são conectados elementos do tipo fluxo, figura 4.2-b. Um fluxo pode ser positivo ou negativo: fluxos positivos incrementam o valor do tanque; fluxos negativos por sua vez, o decrementam. Um conversor é uma primitiva capaz de transformar o valor de suas conexões de entrada em um valor que é colocado em suas conexões de saída. Tipicamente realiza uma transformação algébrica como, por exemplo, na figura 4.2-c o conversor de **área retângulo** torna disponível em sua conexão de saída o valor da área de um retângulo dado como o produto de suas conexões de entrada. A fórmula que permite a transformação realizada por um conversor pode ser vista e/ou editada dando-se um duplo *click* no mesmo. Por exemplo, no caso do conversor **área retângulo** sua fórmula é dada por **base * altura**.

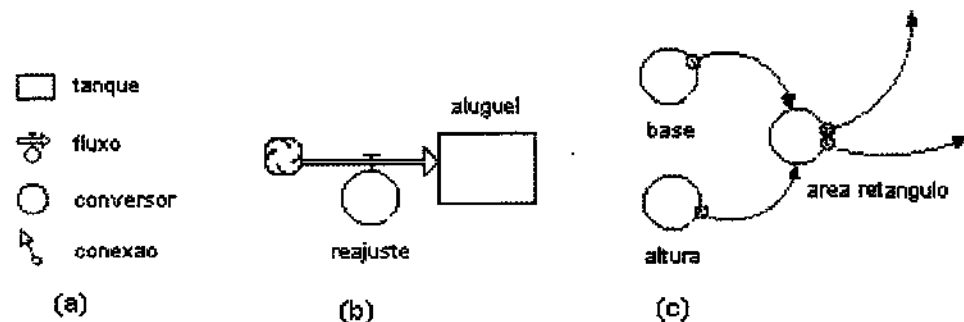


Figura 4.2: Primitivas do ambiente de modelagem e simulação Ithink.

O Ithink permite que um modelo seja executado, provendo ao usuário exibição de tabelas, *displays* numérico ou gráficos mostrando a variação das quantidades envolvidas

ao longo do tempo. Ou ainda, o usuário pode solicitar a animação do diagrama construído. A animação mostrará a variação do conteúdo dos elementos do tipo tanque e conversor imitando o enchimento ou esvaziamento de uma caixa d'água. A figura 4.3 ilustra um modelo no Ithink que estabelece uma inter-relação entre presa e predador em um ecossistema de 100 hectares contendo, inicialmente, 6000 presas e 125 predadores. O modelo supõe que cada presa procria a uma taxa tal que sua população aumenta em 25% por mês. Já os predadores se reproduzem a uma taxa menor, equivalente a 2.5% por mês. O modelo supõe, ainda, que a densidade populacional de presas é diretamente proporcional à mortalidade dos mesmos ($0.3 * \text{densidade de presas}$) e é inversamente proporcional à mortalidade dos predadores ($1.2 / \text{densidade de presas}$). A lógica do modelo no que diz respeito à mortalidade de presas e predadores reside no fato de quanto maior for a densidade das presas, maior será a mortalidade das mesmas, pois serão capturadas mais facilmente pelos predadores (relação direta) e também, pelo fato de que uma maior densidade de presas ocasiona menor mortalidade de predadores pois os mesmos terão maior oferta de alimentos (relação inversa).

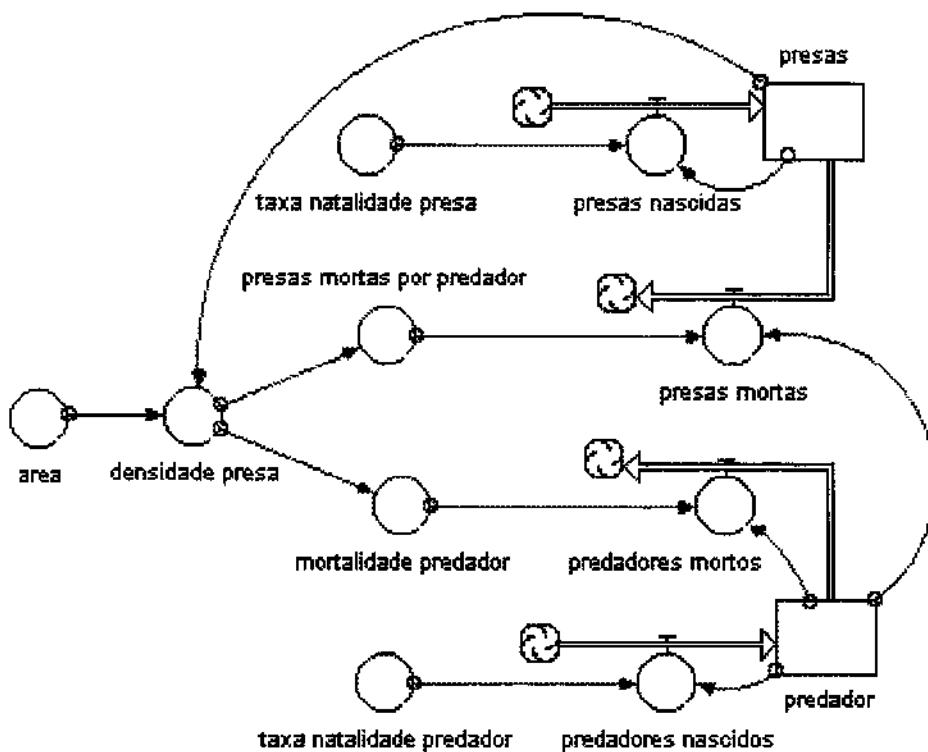


Figura 4.3: Modelo de inter-relação presa e predador no Ithink.

O Extend também é um ambiente de modelagem e simulação de propósito geral, possuindo um conjunto de primitivas gráficas, denominadas de "blocos", que podem ser

utilizados para construir modelos através de uma interface de manipulação direta. Estes blocos executam funções específicas sobre os dados que são conduzidos aos seus conectores de entrada e tornam-se disponíveis em seus conectores de saída. Os dados são transformados segundo a sua funcionalidade. Por exemplo, os blocos da figura 4.4-a possuem a função de multiplicar ou dividir os dados que chegam aos seus conectores de entrada colocando o resultado em seu conector de saída. O bloco da figura 4.4-b permite que o usuário digite uma equação que relacione os dados que chegam aos seus conectores de entrada (à esquerda do bloco) para compor uma única saída (à direita do bloco). O bloco da figura 4.4-c assemelha-se ao elemento do tipo tanque do Ithink, onde os dados podem incrementar o valor corrente do bloco se forem conduzidos a este pelo seu conector "torneira" localizado no topo do bloco, ou o valor do bloco pode ser decrementado em quantidades iguais ao valor do dado conectado ao seu conector *want* e ainda o valor corrente acumulado pelo bloco pode ser conhecido através de seu conector *c*. O bloco da figura 4.4-d destina-se a tornar disponível um valor constante em seu conector de saída ou uma transformação algébrica realizada sobre o seu único conector de entrada.

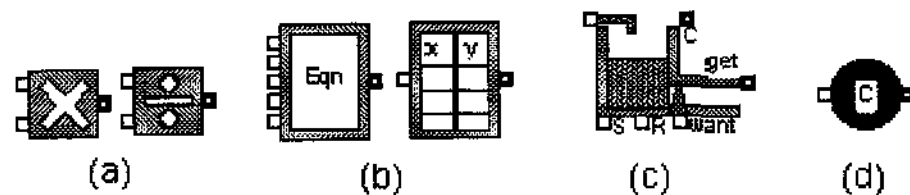


Figura 4.4: Primitivas do ambiente de modelagem e simulação Extend.

A figura 4.5 ilustra o modelo Extend correspondente ao modelo de inter-relação entre presa e predador discutida anteriormente.

O Extend permite a animação dos blocos envolvidos em um modelo durante a simulação, onde os blocos do tipo acumuladores enchem e esvaziam durante a execução e é possível a visualização de sinais luminosos correspondentes aos dados trafegando pelo modelo. O usuário pode construir blocos que contém em seu interior um submodelo, criando algo que pode ser comparado a um subprograma nas linguagens de programação. O Extend também provê uma linguagem compilada denominada de **ModL** bastante semelhante à linguagem de programação **C**, que permite construir novos blocos ou alterar o comportamento dos blocos predefinidos. O Extend é absolutamente extensível não havendo, virtualmente, nenhum limite à criação de modelos segundo, a Imagine That Inc. (1993).

4.4. Um ambiente computacional de modelagem e simulação para treinamento em um

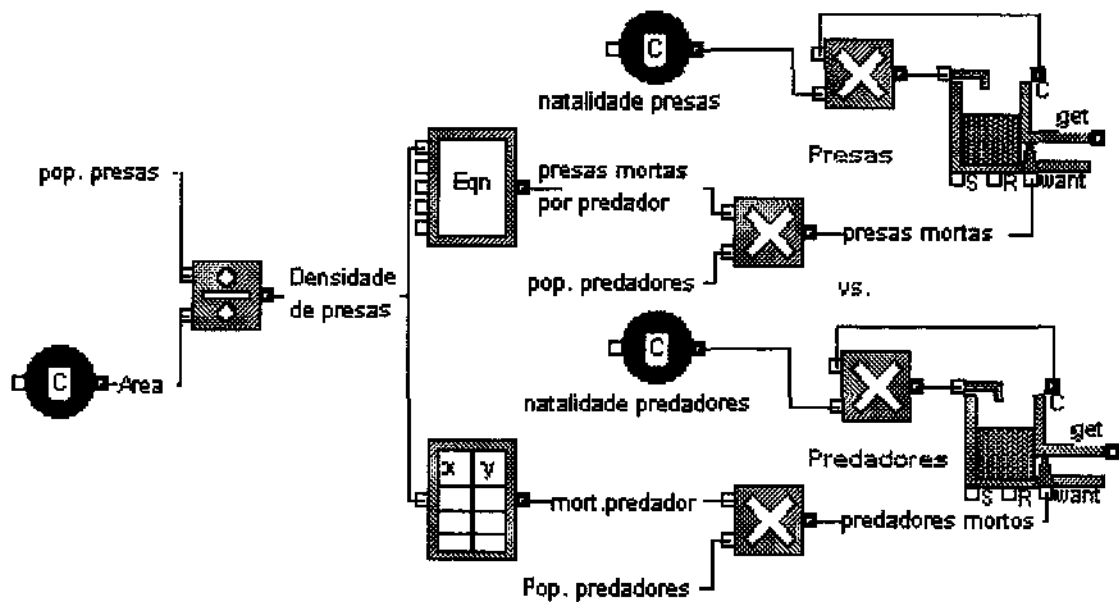


Figura 4.5: Modelo de inter-relação presa e predador no Extend.

Estes ambientes apresentam uma problemática quando transpostos para o contexto de treinamento, o que está sendo referido neste trabalho. Eles têm sido tradicionalmente dirigidos a profissionais experientes, com competência formal em matemática, dos quais se espera habilidade em manipular estruturas simbólicas que representam fórmulas algébricas, ou seja são destinados a usuários no nível gerencial, não podendo ser aplicados a outros funcionários da empresa que necessitam de treinamento. A generalidade destes ambientes solicita do usuário atenção com aspectos nem sempre vinculados ao domínio sobre o qual o modelo está sendo desenvolvido. Esta ausência de suporte a um domínio específico de trabalho pode levar o usuário a se sentir perdido com a manipulação aleatória dos vários recursos do sistema, que não fazem parte do seu cotidiano, ou pode requerer do usuário um nível de conhecimento mais profundo em modelos matemáticos, o que não ocorre no seu mundo real da aplicação proposta neste trabalho.

4.4 Um ambiente computacional de modelagem e simulação para treinamento em um domínio específico

A proposta deste trabalho é estabelecer um equilíbrio entre os benefícios e dificuldades encontradas nos ambientes de modelagem e simulação descritos na seção anterior.

4.4. Um ambiente computacional de modelagem e simulação para treinamento em

Propõe-se o uso de um ambiente computacional de modelagem e simulação voltado para um domínio específico, objetivando o ensino/aprendizado de conceitos associados a esse domínio. Busca-se, assim, um equilíbrio entre ambientes puramente de simulação, que são fáceis de serem utilizados, mas provêem um baixo nível de aprendizado e os ambientes de modelagem e simulação de propósito geral, que são complexos de serem utilizados e nem sempre os índices de aprendizados são satisfatórios [Oliveira, 1995]. O usuário, dentro do domínio do seu trabalho, que é composto por um conjunto específico de fronteiras, poderá sentir-se mais situado nesse ambiente, uma vez que estará manipulando com entidades que fazem parte de seu mundo real.

Pouco consta na literatura sobre ambientes computacionais baseados em modelagem e simulação voltados para a formação de funcionários no gerenciamento de uma linha de produção. Cientes da problemática e da necessidade de novos métodos de treinamento que esta área requer, este trabalho envolve o projeto de um ambiente computacional de modelagem e simulação que permita o treinamento de funcionários do chão de fábrica em novas técnicas de manufatura. Tal ambiente deve permitir que eles testem novas idéias, novos procedimentos de trabalho que possam vir a ser mudados em seus ambientes reais.

O ambiente de modelagem e simulação proposto visa possibilitar ao usuário um nível de abstração do seu mundo real, necessário ao entendimento de como o seu mundo funciona. Pensando de forma lógica e consistente nos problemas, os aprendizes podem determinar a relação de "causa" e "efeito" entre suas ações e resultados. Nesse processo, eles podem deduzir alguns princípios básicos que podem ser usados para transformar suas empresas em empresas de sucesso. Tendo o ambiente o propósito educacional, ele desloca o foco de atenção para os aspectos de interação entre usuário e sistema, possibilitando ao aprendiz construir, testar e refinar o seu modelo de forma o mais concreta possível.

Este ambiente de modelagem e simulação deverá permitir que o usuário crie modelos de seu ambiente real, construindo uma linha de produção, onde ele poderá atuar como um gerente ou como um técnico da sua "fábrica virtual", alterando processos, aplicando testes e analisando os impactos causados. Com isto o usuário pode obter uma visão mais ampla de seu trabalho, assim como implantar melhorias sobre o mesmo.

O usuário terá a sua disposição um conjunto de primitivas que representam elementos que fazem parte de seu cotidiano, para a criação de seu modelo. Estas primitivas representam elementos como fornecedores, estoques, cliente, estações e etc. E através de uma interface de manipulação direta ele poderá definir as propriedades das primitivas, interconectá-las e definir aspectos vinculados à interrelação entre as primitivas. No próximo capítulo será apresentada uma visão geral da funcionalidade do ambiente.

Capítulo 5

Ambiente Enxuto, uma visão geral de sua funcionalidade

Este capítulo, procura caracterizar o ambiente para modelagem e simulação Enxuto do ponto de vista do usuário. Apresenta o ambiente Enxuto em sua funcionalidade. Isto é, os elementos que o compõem são apresentados e a função que cada um deles exerce é explicada sem que, para isto, tenha-se que citar, por exemplo, fundamentos matemáticos e estatísticos, assim como técnicas de implementação ou de projeto, que serão explicadas nos capítulos subsequentes.

5.1 Características gerais do ambiente Enxuto

Conforme já mencionado, Enxuto é um protótipo de um sistema computacional para modelagem e simulação, que foi desenvolvido com o objetivo de permitir o treinamento de novas técnicas de produção em um contexto de manufatura. Seu desenvolvimento foi dirigido a seu usuário alvo: pessoas que trabalham diretamente no chão de fábrica e passam a maior parte de seu tempo operando máquinas e alimentando-as com matéria prima.

O ambiente Enxuto permite a modelagem e a simulação dentro do domínio de manufatura. Para tal, ele permite a representação de elementos da linha de produção em desenhos de duas dimensões, assim como o estabelecimento de seus comportamentos e descrição de aspectos correspondentes às interrelações entre estes elementos. O usuário pode "arrastar" esses elementos a partir de uma palheta de elementos, onde cada elemento possui propriedades padrão que são criadas automaticamente com ele. Por exemplo o elemento Estação possui como propriedades, tempo médio de quebra, capacidade de produção,

número de operadores, etc. Já o elemento Container possui, velocidade de transporte, capacidade de carga, etc. Estas propriedades são características de cada elemento, podendo ser referenciadas pelo usuário quando da definição do comportamento ou na interrelação entre os elementos.

Um modelo do ambiente real da fábrica pode ser definido através de uma simples composição de vários elementos. A figura 5.1 a seguir ilustra o conjunto dos elementos¹ disponíveis que permitem criar modelos dentro do domínio de atuação.

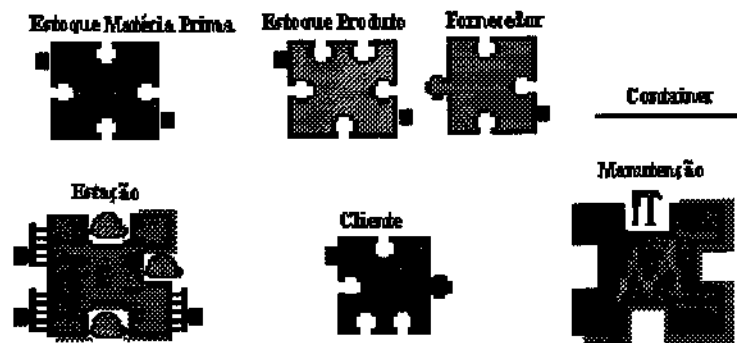


Figura 5.1: Objetos gráficos que representam os elementos do domínio de atuação do ambiente Enxuto.

Para ilustrar o uso que pode ser feito do ambiente, considere a tarefa de modelar uma fábrica com a seguinte composição: um **Fornecedor** de matéria prima, um **Cliente** que de tempos em tempos realiza alguns pedidos de produtos junto a fábrica. Internamente a fábrica é composta por **Estoques**, **Containers**, e **Estações** (máquinas). O que o usuário deve fazer? Ele deve buscar do ambiente Enxuto figuras que representam os elementos citados, podendo estabelecer na tela a posição onde melhor lhe convier para cada elemento. Isto é feito clicando e arrastando os elementos presentes na palheta de primitivas. O comportamento de cada um deles é definido, fazendo um duplo clique em cada elemento e editando o conteúdo de seus atributos. Em seguida ele deve interconectar os elementos através de containers, de tal forma que a saída de um elemento (produto) se torne a entrada do elemento subsequente (matéria prima). Pode-se observar no ambiente Enxuto, que as conexões de fornecedor com estoque de matéria prima e estoque de produto com clientes já estão estabelecidas, necessitando apenas definir suas características. Isto ocorre devido ao fato de que na maioria das fábricas todas as matérias primas que chegam do fornecedor vão para o estoque de matéria prima, o mesmo ocorrendo com o produto final da fábrica, que vai para um estoque de produto, para depois ser entregue ao cliente. A figura 5.2 representa o modelo da fábrica descrita acima.

¹Também denominados **primitivas**

Após a criação do modelo e definição dos atributos de cada elemento, o usuário deverá selecionar o método de produção que será aplicado sobre o modelo: método de produção Empurrar ou Puxar. Após esta definição o usuário poderá, iniciar a simulação do modelo, através da opção "Inicializar" do menu "Simulação". Neste ponto o ambiente então passa a validar o modelo construído pelo usuário, verificando se faltam conexões, definições de atributos dos elementos ou das interconexões entre eles. Caso o ambiente Enxuto verifique que está faltando conexão entre elementos, ele notifica o usuário da falta de conexão por meio de mensagens. Já quando ele identifica a ausência de definição de atributos relevantes de um elemento, é colocado um ponto de interrogação (?) sobre o elemento em questão, aguardando que o usuário defina os atributos que ainda não foram definidos.

Caso o modelo seja válido então o ambiente Enxuto irá pedir para o usuário o tempo total da simulação, que será realizada sobre o modelo. Neste ponto o usuário pode solicitar a execução da simulação através da opção "Executar" do menu "Simulação".

No término da simulação o usuário pode solicitar do ambiente, ferramentas que demonstrem o desempenho das estações, fornecedor, containers, demanda de produto por parte do cliente, permitindo que cada elemento possa ter seus atributos inspecionados, através de uma janela de diálogo, que é aberta toda vez que o usuário realiza um duplo clique sobre um determinado elemento do modelo. O ambiente Enxuto também informa ao usuário a porcentagem de tempo decorrido durante a simulação.

5.2 Descrição do ambiente Enxuto

O processo de produção sofre a influência de vários fatores, desde o recebimento de matéria prima do fornecedor até a entrega do produto final ao cliente. Um ambiente para modelagem e simulação para o domínio de chão de fábrica deve prover meios para representar os elementos que compõem este domínio, bem como oferecer ferramentas que permitam descrever o comportamento destes elementos e as relações entre eles. Ainda no sentido de permitir que o usuário observe os resultados da simulação e a validade do modelo criado, um ambiente de modelagem e simulação deve disponibilizar um conjunto de ferramentas adicionais que possibilitem o acompanhamento da execução da simulação do modelo. Com o objetivo de englobar as requisições citadas acima o ambiente Enxuto dispõe, portanto, de ferramentas que permitem:

- A criação/edição de modelos, que são uma abstração do domínio de atuação. Isso é feito através de desenhos de duas dimensões, representando os elementos do domínio.

- Configuração do modelo, que possibilita o ajuste de parâmetros, inicialização das variáveis dos elementos que compõem o modelo.
- Execução e acompanhamento da simulação do modelo criado. Na execução, o ambiente Enxuto mostrará os elementos envolvidos, assim como os valores de seus respectivos atributos.

A atividade de simular um modelo no ambiente Enxuto pode ser entendida, como a soma de duas atividades. A atividade de criar o modelo, sendo caracterizada como a atividade de modelagem e a atividade de simulação propriamente dita, que é formada pelas etapas de configuração, execução e acompanhamento das alterações que surgem nos elementos do modelo. Ambas as atividades têm a sua importância dentro do ambiente Enxuto, uma vez que a atividade de modelagem permite ao usuário obter um maior conhecimento do seu mundo real, pois este terá que pesquisar, estudar e levantar um maior número de informações possíveis, para que possa criar um modelo que conceitualmente imite o seu ambiente real. Na fase de simulação o usuário poderá verificar os resultados obtidos, podendo validar ou não o modelo criado e além disso o usuário poderá obter um maior conhecimento das variações de sua realidade, orientando as suas tomadas de decisões.

Antes de descrever as características do ambiente Enxuto que permitem desenvolver estas atividades, será mostrado como o ambiente Enxuto é inicializado, serão descritas as opções que ele disponibiliza e como ele é encerrado após o término do trabalho.

O ambiente Enxuto foi desenvolvido de forma que a sua interface com o usuário execute sobre o sistema operacional da Microsoft Windows 95. Logo, pode-se inicializar e encerrar a execução do ambiente Enxuto, exatamente da mesma forma como qualquer outra aplicação do Windows 95. Sendo assim deve-se realizar um duplo clique sobre o ícone do ambiente Enxuto, que aparecerá sua janela principal, como mostrado na figura 5.2. Esta janela contém os seguintes elementos:

Barra de título: Contendo o título do aplicativo, neste caso Enxuto;

Barra de menus: Mostra os nomes dos comandos disponíveis;

Ícone de controle: Quando o mouse é pressionado sobre este ícone ele apresenta os comandos para trabalhar com as janelas (minimizar, maximizar, restaurar, mover, fechar, etc).

Ícone minimização e maximização: Expande a janela ao tamanho máximo da tela ou reduz a um ícone a barra de controle da aplicação.

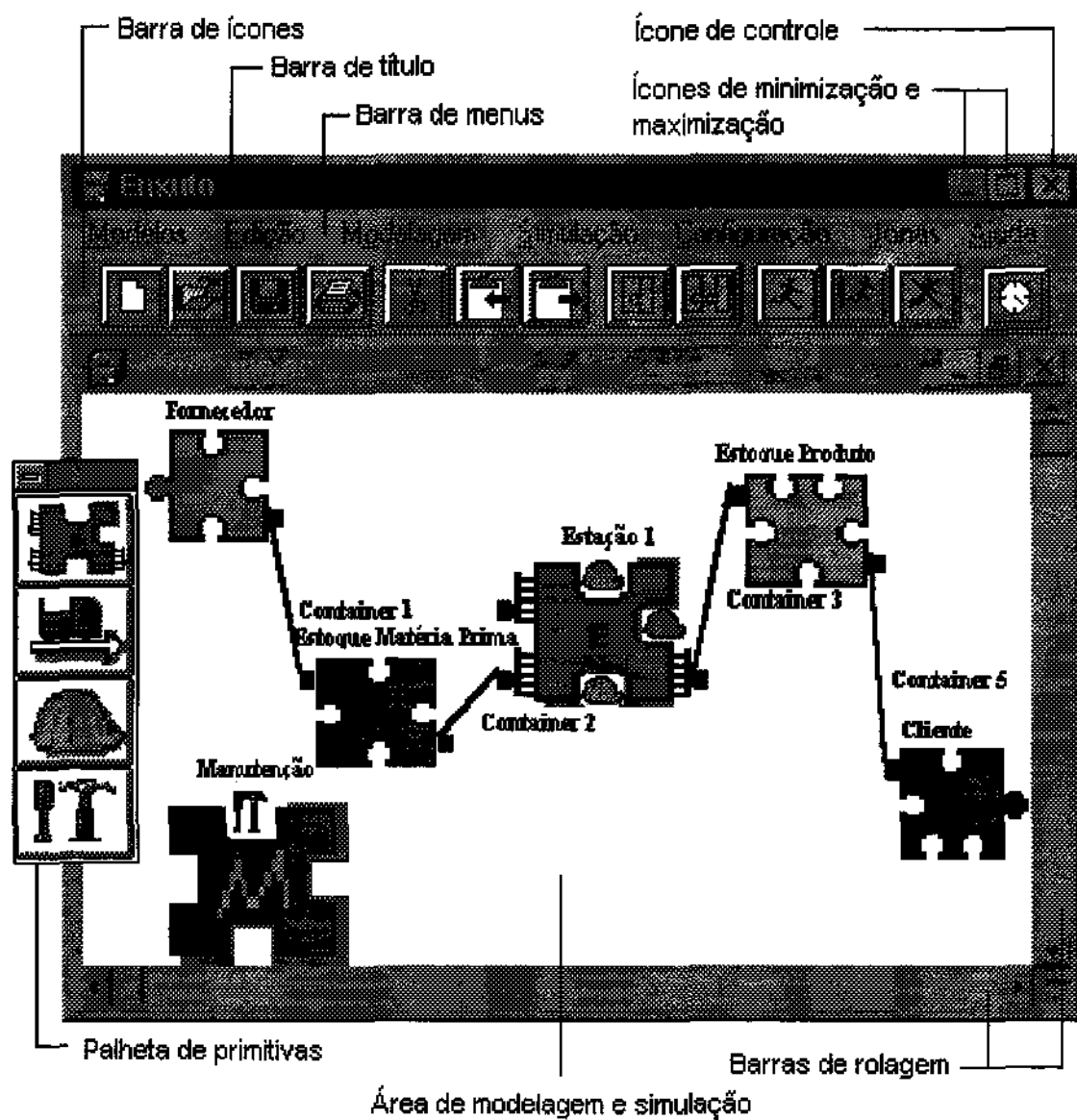


Figura 5.2: Janela principal do ambiente Enxuto.

Barra de ícones: Contém ícones para os principais comandos que podem ser selecionados pelo usuário. A maioria destes comandos pode ser acionada, também, através do uso de menus.

Barra de rolagem: Uma página de um modelo pode ser maior que a janela do mesmo. As barras de rolagem podem ser utilizadas para trazer o resto da página à vista.

Área de modelagem e simulação: É a área destinada à criação, configuração e simulação do modelo e permitirá ao usuário acompanhar a execução.

Palheta de primitivas: É formada por um conjunto de primitivas a partir das quais o usuário poderá compor seu modelo. Pode ser "arrastada" para qualquer posição da tela, que o usuário desejar.

Existem quatro formas diferentes de acionar um comando no ambiente Enxuto. Primeiro através do mouse, dando um clique no item de menu da barra de menus e escolhendo, em seguida, com um outro clique a opção desejada. Segundo, também através do mouse, dando um click em algum ícone da barra de ícones. Uma terceira forma é através do uso da tecla (Alt) em conjunto com uma das letras sublinhadas do menu, e em seguida pressionando uma das letras sublinhadas dentro dos itens do submenu escolhido. E quarto e último, é através do uso de teclas de atalho, como (Ctrl-A), que aciona o comando de abrir um modelo. Portanto, o ambiente Enxuto facilita a interação do usuário em diferentes estágios de habilidade tornando disponíveis ícones e teclas de atalho, para os comandos mais usados.

O ambiente Enxuto ainda não permite que se abra mais de um modelo, mas em seu projeto já foi especificada esta característica, podendo ser implementada em versões futuras do mesmo. Com isso permitirá ao usuário abrir mais de um modelo e realizar a execução destes modelos, comparando os resultados obtidos e o desempenho dos modelos abertos. Algumas restrições se aplicam quando se está trabalhando com mais de um modelo, por exemplo, vários comandos têm a característica de só aplicar-se a um dado modelo. Para solucionar esta restrição deve-se considerar o conceito de modelo corrente ou ativo ou selecionado. Para selecionar um modelo basta dar um clique em qualquer ponto da janela do modelo de interesse. Isto destacará a barra de título da janela, indicando que esta é a janela ativa, diferenciando o modelo selecionado dos demais modelos que por ventura, estejam abertos. As janelas dos modelos poderão ser movimentadas, redimensionadas, maximizadas ou minimizadas dentro da janela principal do ambiente Enxuto, permitindo que o usuário organize a sua área de trabalho de acordo com a sua necessidade.

As alterações feitas em um modelo podem ser gravadas, usando-se os comandos "Salvar" ou "Salvar como". Um modelo pode ser gravado com um outro nome através do comando "Salvar como". O comando "Novo" permite que seja aberto um novo modelo 'em branco', com a área de modelagem vazia permitindo que o usuário crie um novo modelo. Escolhendo o comando "Abrir", o ambiente Enxuto apresentará uma caixa de diálogo onde serão mostrados os modelos disponíveis para a abertura no diretório corrente, ou o usuário poderá pesquisar o conteúdo de outros diretórios ou drives.

Pode-se fechar o modelo através de um duplo clique em seu ícone de controle da sua janela. Ou através do comando "Sair", que fechará todos os modelos abertos e encerrará a execução do ambiente Enxuto. Caso alterações tenham sido realizadas no modelo sem terem sido gravadas, será então apresentada uma mensagem de advertência que permitirá ao usuário, gravar ou não, ou ainda, desistir da operação de "Sair".

5.3 Criação de modelos

Pode-se criar modelos do chão de fábrica no ambiente Enxuto, através de uma interface de manipulação direta de forma similar a diversos software destinados à criação de desenhos.

A atividade de construir um modelo pode ser entendida como a atividade de agregar primitivas para produzir estruturas mais complexas. O ambiente Enxuto possui uma palheta de primitivas, sendo composta por um conjunto de elementos que expressam componentes do seu domínio de atuação.

A palheta de primitivas é composta por um conjunto de quatro elementos: Estação, Container, Operador e Técnico. O desenho de um elemento contido na palheta pode ser feito da seguinte forma:

1. Seleciona-se um elemento na palheta, através de um clique no ícone do elemento;
2. Move-se o elemento para a área apropriada de modelagem, apertando e mantendo pressionado o botão esquerdo do mouse à medida em que este é arrastado em direção a posição desejada (o desenho do elemento irá aparecendo, à medida em que o mouse é deslocado);
3. Solta-se o botão esquerdo do mouse quando atingir a posição desejada do elemento.

Vários comandos do ambiente Enxuto recaem sobre os elementos selecionados. Portanto para selecionar um elemento deve-se dar um clique no elemento desejado. Para

selecionar todos os elementos desenhados na área de modelagem, pode-se usar o comando "Selecionar tudo" do menu "Edição". Os elementos selecionados aparecerão com os marcadores de seleção ao seu redor conforme figura 5.3.

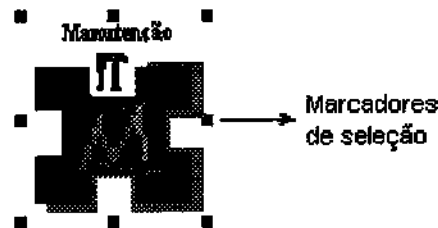


Figura 5.3: Marcadores de seleção.

Quando um elemento está selecionado, pode-se por exemplo, remover, movimentar, ou caso seja um Container realizar a sua conexão com os elementos origem e destino. Um elemento permanece selecionado até que se cancele a sua seleção. Para desativar uma seleção deve-se dar um clique em algum ponto na área de modelagem que não possua nenhum elemento desenhado.

À medida em que o usuário desenha os elementos do modelo ele pode ir realizando as conexões entre estes, de forma que defina o fluxo da produção. Quando o usuário aciona o comando "Novo" do menu "Modelo" o ambiente Enxuto disponibiliza uma nova área de modelagem e simulação, já contendo alguns elementos iniciais, "default", tais como: Fornecedor, Estoque Matéria prima, Estoque produto, Cliente e um Gerenciador de manutenção, assim como algumas conexões entre estes elementos já previamente definidas, uma vez que todos os modelos de chão de fábrica contém estes elementos. Portanto, o usuário irá precisar apenas buscar os desenhos das Estações e Containers, a partir da palheta de primitivas, como foi descrito anteriormente.

Para realizar a conexão entre um elemento e outro, deve-se primeiro selecionar o container que irá fazer a conexão entre os elementos, segundo dar um clique no marcador de seleção na extremidade direita ou esquerda do Container, mantendo o botão esquerdo do mouse pressionado. Arrasta-se até o marcador de conexão do objeto fonte (figura 5.4), caso seja a extremidade esquerda do Container que se está conectando, ou até o marcador de conexão do objeto destino, caso seja a extremidade direita. Quando o mouse estiver sobre o marcador de conexão basta liberar o botão esquerdo do mouse, que a conexão será estabelecida. O Container irá mudar de cor, passando de vermelho para preto, indicando que a conexão foi realizada, com sucesso.

Considerando que a representação no ambiente Enxuto é bidimensional a profundidade dos elementos não é representada, porém o nível no qual os elementos são desenhados

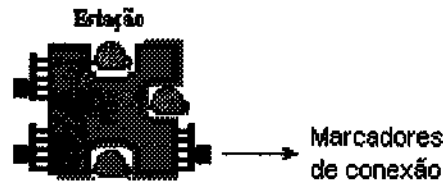


Figura 5.4: Marcadores de conexão.

afeta a percepção visual, uma vez que uns sobrepõem os outros (figura 5.5). Toda vez que um elemento é criado no ambiente Enxuto ele é colocado na frente dos demais.



Figura 5.5: Nível dos elementos criados.

A área de transferência "*Clipboard*" é um recurso oferecido pelo Windows, que permite que dados, desenhos e etc, sejam armazenados temporariamente. Isto permite, por exemplo que aplicações troquem dados entre elas ou que, uma mesma aplicação possa armazenar dados na área de transferência, com o objetivo de uma subsequente cópia. Logo pode-se ter um meio eficiente de criar múltiplas cópias de uma determinada informação. Aplicativos Windows provêm sempre um mecanismo padrão para trabalhar com a área de transferência, provendo três operações básicas. Uma operação de "cópia", na qual um aplicativo pode copiar dados para a área de transferência sem contudo removê-lo; outra operação normalmente definida como "cortar", que remove o dado do aplicativo e o armazena na área de transferência. E, por fim uma operação geralmente denominada "colar" que permite que o dado seja copiado da área de transferência para o aplicativo. O ambiente Enxuto apresenta estas três operações básicas através dos comandos do menu "Editar", denominadas de "Copiar", "Cortar" e "Colar", facilitando ao usuário a criação de um modelo.

5.4 Configuração do modelo criado

Após a criação do modelo, desenhando os elementos e a realização das conexões entre os elementos que compõem o modelo, deve-se definir as características e o comportamento desses elementos no decorrer da simulação. Esta seção descreve os atributos de cada um dos elementos, seu significado e também como é definido cada atributo.

Cada elemento do modelo possui um determinado conjunto de atributos (ou propriedades) que devem ser definidas pelo usuário. Estes atributos influem na execução da simulação, definindo um comportamento próprio para cada um dos elementos do modelo, durante a execução da simulação. Para que o usuário defina os valores destes atributos, basta que seja pressionado um duplo clique sobre o elemento específico que se deseja definir. Neste ponto o ambiente Enxuto abrirá um caixa de diálogo com cada um dos atributos do elemento e com os respectivos campos de edição de seus valores. O usuário poderá então editar os valores que cada um destes atributos deverá ter no decorrer da simulação. O usuário pode movimentar a caixa de diálogo para qualquer posição dentro da área da janela de modelagem e simulação, facilitando a edição dos campos. A seguir serão mostradas todas as caixas de diálogos para todos os elementos e também uma breve descrição de cada um dos atributos.

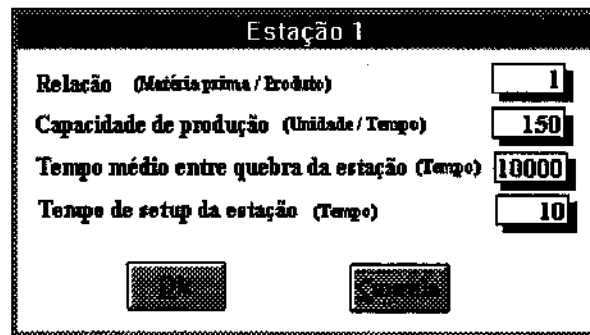
Para cada elemento Estação do modelo, o usuário deverá editar os valores dos seguintes atributos conforme ilustram as figuras 5.6 a 5.11:

- **Estação**

1. Relação (matéria prima / produto): este atributo especifica quantas unidades de matéria prima são necessárias para a produção de uma unidade de produto.
2. Capacidade produção (unidade / tempo): define quantas unidades de produto que a Estação é capaz de produzir em uma unidade de tempo.
3. Tempo médio entre quebras da Estação (tempo): este atributo define um intervalo médio entre a ocorrência de quebra na Estação.
4. Tempo setup da Estação (tempo): este atributo especifica o tempo gasto para inicialização da Estação para atender uma nova requisição. Este tempo inclui tempo para regulagem, troca de fôrmas, troca de operadores, etc, o que ocorre geralmente quando a produção da máquina irá atender uma nova requisição.

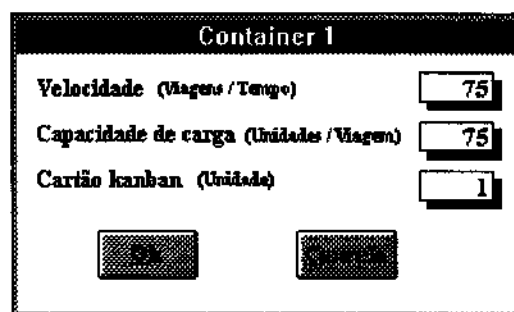
- **Container**

1. Velocidade (viagens / tempo): este atributo especifica quantas viagens que o Container é capaz de realizar em uma unidade de tempo.
2. Capacidade de carga (unidade / viagem): define quantas unidades de produto ou matéria prima que o Container é capaz de transportar por viagem.
3. Cartão kanban (unidade): define de quantas unidades é o cartão kanban no método de produção puxar.



Estação 1	
Relação (Matéria-prima / Produto)	1
Capacidade de produção (Unidade / Tempo)	150
Tempo médio entre quebra da estação (Tempo)	10000
Tempo de setup da estação (Tempo)	10
OK Cancelar	

Figura 5.6: Diálogo de edição das características de entrada da Estação.



Container 1	
Velocidade (Viagens / Tempo)	75
Capacidade de carga (Unidades / Viagem)	75
Cartão kanban (Unidade)	1
OK Cancelar	

Figura 5.7: Diálogo de edição das características de entrada do Container.

- **Cliente**

1. Quantidade média de cada pedido (unidade): representa qual a quantidade de produto que o Cliente irá pedir em cada requisição sua feita junto à fábrica.
2. Tempo médio entre cada pedido (tempo): especifica qual o intervalo entre a realização de cada requisição ou seja de quanto em quanto tempo deve ser feito um pedido.

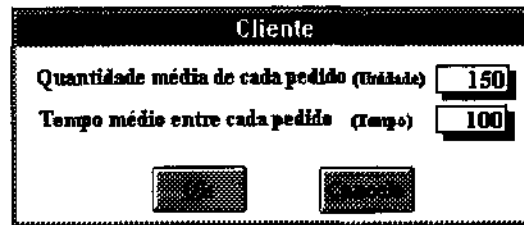


Figura 5.8: Diálogo de edição das características de entrada do Cliente.

- **Fornecedor**

1. Quantidade média (unidades / tempo): representa a quantidade de matéria prima em unidades, que o Fornecedor é capaz de fornecer em uma unidade de tempo.



Figura 5.9: Diálogo de edição das características de entrada do Fornecedor.

- **Estoque matéria prima / produto**

1. Nível crítico (unidade): especifica a quantidade mínima de matéria prima ou produto que deverá ser mantida em estoque, funcionando como uma reserva.

- **Técnico**

1. Tempo médio de conserto (tempo): define o tempo gasto para que o Técnico realize os reparos em uma determinada Estação com defeito.

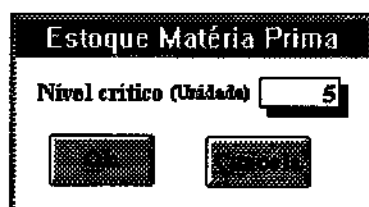


Figura 5.10: Diálogo de edição das características de entrada do Estoque matéria prima / produto.

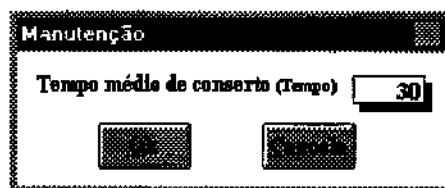


Figura 5.11: Diálogo de edição das características de entrada do Técnico.

5.5 Execução e acompanhamento do modelo criado

Terminada a fase de modelagem, onde o usuário criou o modelo, definiu os valores dos atributos que determinam o comportamento dos elementos que compõem o modelo, ele estará apto a entrar na fase de simulação. Nesta fase o usuário irá executar a simulação, assim como realizar um acompanhamento dos resultados e estatísticas retornados durante a simulação dos modelos.

Antes de iniciar a execução da simulação, deve-se definir qual o método de produção que a simulação irá seguir durante sua execução. Isto é feito através do menu "Simulação", usando a opção "Método de produção", o qual irá abrir um submenu contendo botões de escolha, como mostrado na figura 5.12, tanto para a opção método produção puxar, o que representa a produção Enxuta ou a opção método de produção empurrar, o que especificará que a simulação do modelo será realizada de acordo com a produção em Massa.

O próximo passo é inicializar os objetos da simulação, sendo realizado através da opção "Inicializar" do menu "Simulação". Acionando esta opção o ambiente Enxuto irá realizar uma vistoria em todos os elementos do modelo, para verificar se todas as características foram previamente definidas (se foram definidas corretamente) e se todas as conexões entre os elementos foram realizadas pelo usuário. Caso alguma característica de um determinado elemento não tenha sido definida, o ambiente Enxuto colocará um

sinal de interrogação (?) sobre o elemento indicando que está faltando a definição de algum de seus atributos, como mostrado na figura 5.13.

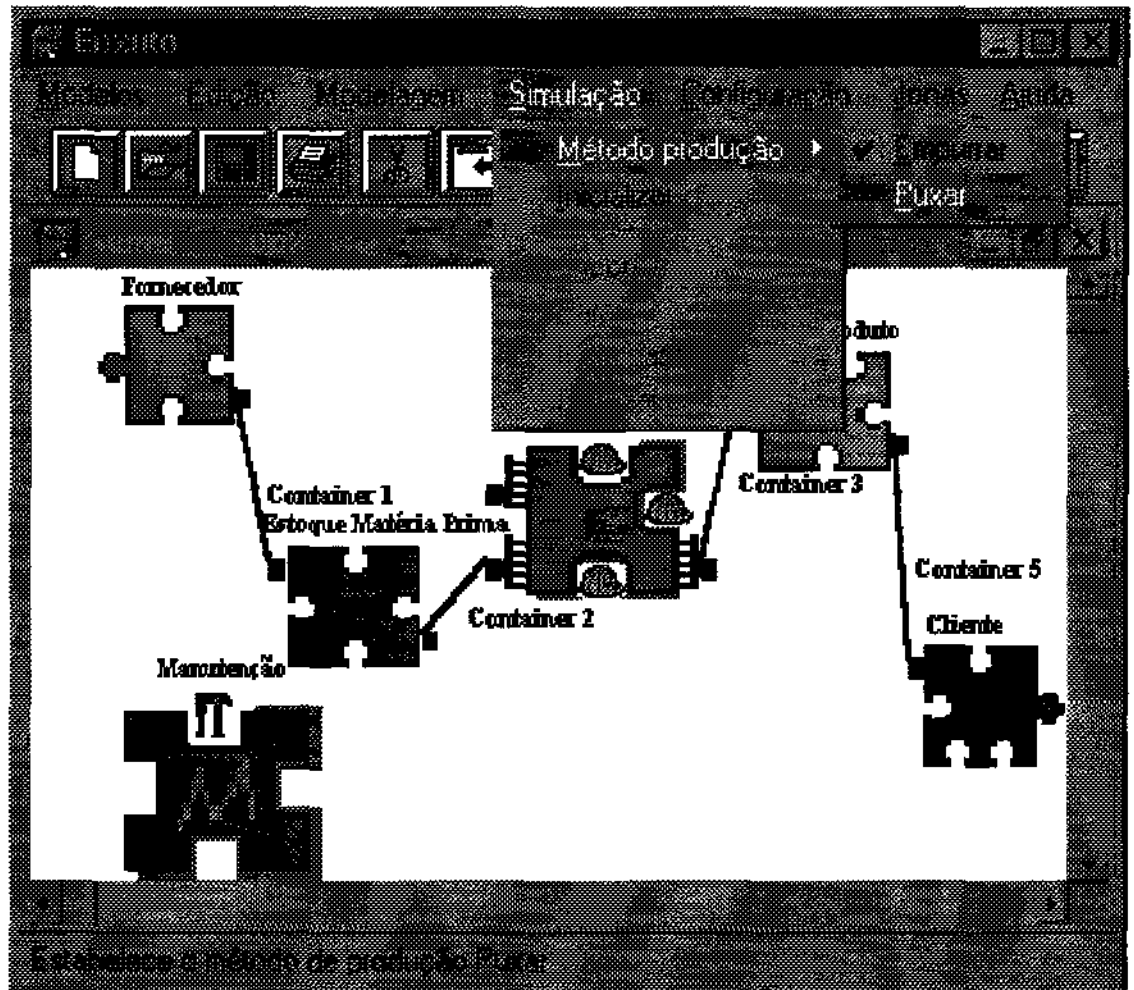


Figura 5.12: Menu de escolha do método de produção.

Caso esteja faltando alguma conexão entre dois elementos do modelo o ambiente Enxuto, exibirá uma janela contendo a mensagem "Falta conexão elemento n". Esta janela é posicionada próxima ao elemento onde está faltando conexão, como mostrado na figura 5.13.

Se o ambiente Enxuto não reportar nenhuma informação indicando a falta de definição de atributos de elementos ou sobre a falta de conexões entre elementos do modelo, então o usuário poderá acionar a execução da simulação. Isto é feito através da opção "Executar" do menu "Inicialização" como mostrado na figura 5.12. Isto acarretará a abertura de

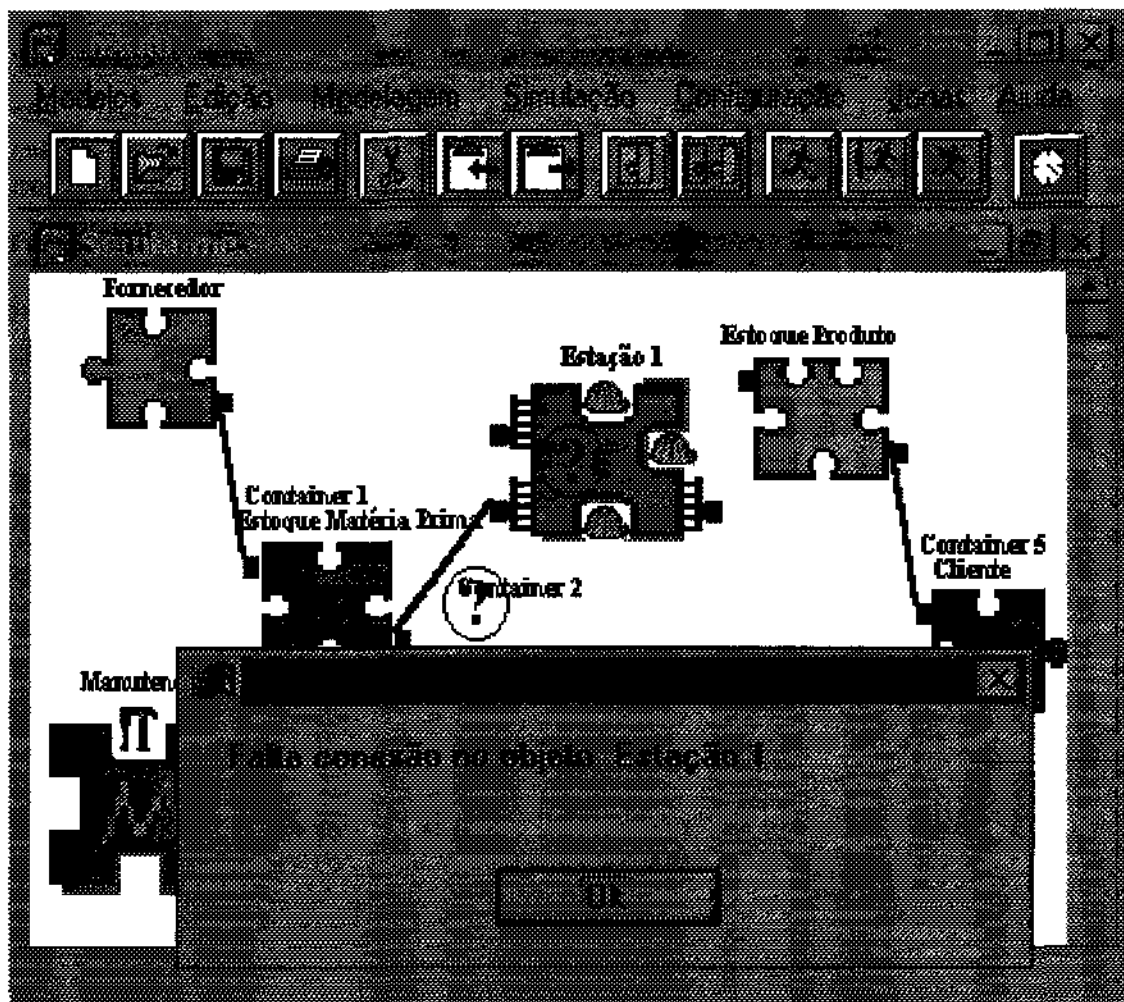


Figura 5.13: Mensagem indicando a falta de conexão.

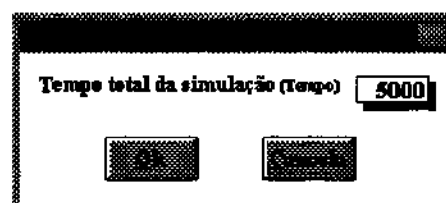


Figura 5.14: Diálogo de edição do tempo total de simulação.

uma caixa de diálogo, pedindo para o usuário informar o tempo total em que a simulação irá ser executada sobre o modelo, como mostrado na figura 5.14. Este tempo indica o número de iterações que a simulação realizará, isto é, uma unidade de tempo é uma iteração. Neste ponto todos os dados de entrada do modelo já foram definidos, portanto o ambiente Enxuto realizará a execução da simulação.

O ambiente Enxuto permite ao usuário gerenciar a execução da simulação, através das subopções do menu "Simulação". A seguir uma breve descrição de suas funções.

O usuário pode interromper a execução da simulação em um determinado ponto. Para isso o usuário deve utilizar a opção "Parar", que permitirá a visualização e análise dos valores dos atributos que cada elemento possui naquele instante. Para continuar a execução da simulação, deve-se acionar a opção "Executar", o que implicará na continuação da execução da simulação a partir do ponto em que foi interrompida anteriormente.

Caso o usuário deseje iniciar uma nova simulação, deve-se utilizar a opção "Finalizar", novamente realizar a definição e ou alteração dos dados de entrada do modelo e inicializar a nova execução da simulação como foi descrito anteriormente.

Para que o usuário visualize os valores dos atributos e as estatísticas dos elementos coletados no decorrer da simulação, deve-se realizar um duplo clique com o botão direito do mouse sobre o elemento de interesse. O ambiente Enxuto então abrirá uma janela de visualização contendo, os valores dos resultados obtidos na execução da simulação. A seguir serão descritas as informações acumuladas pelo ambiente Enxuto, para cada elemento do modelo, como ilustram as figuras 5.15 a 5.20.

• Estação

1. Quantidade matéria prima (unidade): especifica a quantidade de matéria prima que a Estação possui disponível para a realização de sua produção.
2. Quantidade produto (unidade): determina a quantidade de produto que a Estação possui disponível, para que o Container transporte até a próxima Estação da linha de produção.
3. Quantidade produzindo (unidade): é a quantidade de produto que a Estação está produzindo neste instante.
4. Quantidade média de matéria prima (unidade): é quantidade média que a Estação tende a ter disponível para a realização de sua produção.
5. Quantidade média de produto (unidade): define a quantidade média de produto que a Estação tende a ter disponível, para que o Container transporte até a próxima Estação da linha de produção.

6. Quantidade de produto requisitado (unidade): determina a quantidade total de produto que foi requisitada à Estação, desde o início da execução da simulação.
7. Quantidade produzida (unidade): é a quantidade total de produto produzido pela Estação desde o início da execução da simulação.
8. Estado: determina a porcentagem de tempo em que a Estação ficou em cada um dos estados: ociosa, produzindo, esperando matéria prima, quebrada, consertando, atendendo próxima requisição e em setup. O estado que aparecer entre os caracteres < e > , indica o estado corrente do elemento.

Estação 1	
Quantidade matéria prima (Unidade)	5
Quantidade produto (Unidade)	0
Quantidade produzindo (Unidade)	6
Quantidade média matéria prima(Unidade)	1.39986
Quantidade média produto (Unidade)	1.20367
Quantidade produto requisitado (Unidade)	6193
Quantidade produto produzida (Unidade)	6018
Estado	%

Figura 5.15: Diálogo de visualização dos resultados de saída da Estação.

• Container

1. Número de viagens (unidade): determina a quantidade de viagens realizadas pelo Container desde o início da execução da simulação.
2. Quantidade transportada (unidade): define a quantidade de matéria prima ou produto que foi transportado pelo Container desde do início da execução da simulação.
3. Quantidade transportando (unidade): especifica a quantidade atual que o Container está transportando neste instante.
4. Quantidade esperando transporte (unidade): é a quantidade de matéria prima ou produto que está esperando para ser transportada pelo Container.

5. Estado: determina a porcentagem de tempo em que o Container ficou em cada um dos estados: ocioso e transportando.

Container 1	
Número de viagens (Unidade)	83
Quantidade transportada (Unidade)	6141
Quantidade transportando (Unidade)	0
Quantidade esperando transporte (Unidade)	0
Estado	%
Ocioso	40%
Transportando	660%

Figura 5.16: Diálogo de visualização dos resultados de saída do Container.

• Cliente

1. Produto recebido (unidade): determina a quantidade de produto que o Cliente recebeu da fábrica desde o início da simulação.
2. Produto requisitado (unidade): especifica a quantidade de produto que o Cliente requisitou à fábrica desde o início da simulação.
3. Quantidade pendente (unidade): é definido pela subtração do "Produto recebido" a "Produto requisitado".
4. Quantidade média por requisição (unidade): é a quantidade que o Cliente em média pede em cada requisição.
5. Lead time (tempo): é um tempo médio que uma unidade de produto gasta para ser fornecida, desde sua requisição até sua entrega.
6. Número total de requisição (unidade): é total de requisições feitas pelo Cliente, junto à fábrica, desde o início da simulação.
7. Estado: determina a porcentagem de tempo em que o Cliente ficou em cada um dos estados: ocioso e esperando entrega.

• Fornecedor

1. Matéria prima entregue (unidade): representa a quantidade total de matéria prima entregue pelo Fornecedor desde o início da simulação.

Cliente	
Produto recebido (Unidade)	3434
Produto requisitado (Unidade)	6560
Quantidade pendente (Unidade)	3126
Quantidade média por requisição (Unidade)	138.1263
Lead time (Tempo)	1.456107
Número de requisições (Unidade)	47
Estado	%
Ociosos	100%
Produzindo	0%

Figura 5.17: Diálogo de visualização dos resultados de saída do Cliente.

2. Matéria prima requisitada (unidade): representa a quantidade de matéria prima que foi requisitada pela fábrica ao Fornecedor desde do início da simulação.
3. Matéria prima pendente (unidade): é definida pela subtração de "Matéria prima entregue" a "Matéria prima requisitada".
4. Estado: determina a porcentagem de tempo em que o Fornecedor ficou em cada um dos estados: ocioso e produzindo.

Fornecedor	
Matéria prima entregue (Unidade)	6141
Matéria prima requisitada (Unidade)	6147
Matéria prima pendente (Unidade)	6
Estado	%
Ociosos	97.92%
Produzindo	2.08%

Figura 5.18: Diálogo de visualização dos resultados de saída do Fornecedor.

- **Estoque matéria prima / produto**

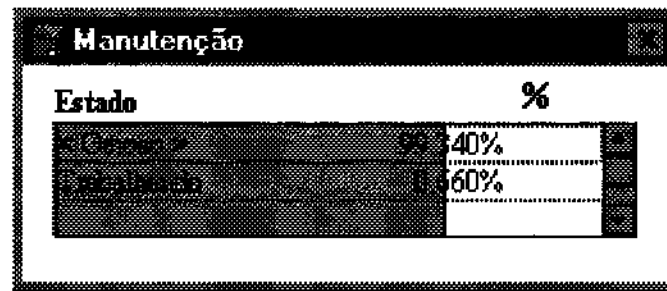
1. Quantidade em estoque (unidade): especifica qual a quantidade de matéria prima ou produto que está disponível em estoque.
2. Quantidade média em estoque (unidade): especifica qual é a quantidade média de matéria prima ou produto disponível, que o Estoque tende a ter.
3. Quantidade entregue (unidade): representa a quantidade total de matéria prima ou produto que foi entregue pelo Estoque desde o início da simulação.
4. Quantidade requisitada (unidade): representa a quantidade total de matéria prima ou produto que foi requisitada ao Estoque desde o início da simulação.
5. Quantidade pendente (unidade): é obtido a partir da subtração da "Quantidade entregue" a "Quantidade requisitada".
6. Estado: determina a porcentagem de tempo em que o Estoque ficou em cada um dos estados: ocioso, esperando completar nível crítico, esperando completar requisição e atendendo próxima requisição, no decorrer da simulação.

Estoque Matéria Prima / Estoque Produto	
Quantidade em estoque (Unidade)	6
Quantidade média em estoque (Unidade)	7 33946
Quantidade entregue (Unidade)	8616
Quantidade requisitada (Unidade)	8616
Quantidade pendente (Unidade)	0
Estado	%
Ocioso	55.13%
Esperando Completar Nível Crítico	40%
Esperando Completar Requisição	40%

Figura 5.19: Diálogo de visualização dos resultados de saída do Estoque matéria prima / Produto.

- **Técnico**

1. Estado: determina a porcentagem de tempo em que o Técnico ficou em cada um dos estados: ocioso e trabalhando, no decorrer da simulação.



Estado	%
Operação	40%
Manutenção	60%

Figura 5.20: Diálogo de visualização dos resultados de saída do Técnico.

Estão incluídos, ainda, no projeto original, outras ferramentas para facilitar a visualização dos resultados da simulação. Novas representações como gráficos que mostrem o desempenho dos elementos do modelo, bem como animação do processo de produção estão sendo desenvolvidas em um trabalho paralelo de continuidade.

Capítulo 6

Aspectos de implementação do ambiente Enxuto

Este capítulo descreve a metodologia utilizada no desenvolvimento do ambiente Enxuto. Também ilustra os módulos que compõem a sua arquitetura apresentando as características e aspectos de implementação de cada módulo.

6.1 Arquitetura do ambiente Enxuto

A arquitetura do ambiente Enxuto é composta basicamente por três módulos: modelagem, simulação e um sistema especialista, como ilustrado na figura 6.1.

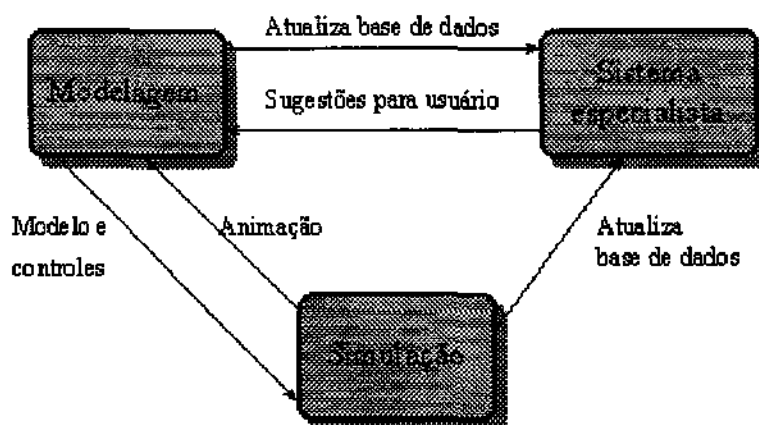


Figura 6.1: Arquitetura do ambiente Enxuto.

6.1.1 Modelagem

O módulo de modelagem é responsável pela interface com o usuário, permitindo a criação de modelos, controle sobre simulação e um acompanhamento dos resultados.

A criação de modelos representa uma interface de entrada do ambiente Enxuto, sendo constituída por um editor de modelos, onde o usuário poderá criar modelos de processos de manufatura que são uma abstração do seu mundo real. Este editor de modelos é um ambiente que possui elementos que permitem ao usuário construir modelos dentro de um domínio específico. Estes elementos são denominados elementos atômicos ou primitivas, que expressam cada um dos componentes do sub-domínio de conhecimento [Baranauskas, Oliveira, 1994]. Pode-se entender a atividade de construir modelos como a atividade de agregar primitivas para reproduzir estruturas mais complexas.

O editor de modelos possui um conjunto definido de primitivas que permitirá ao usuário criar modelos do seu cotidiano de trabalho, através de conexões entre estas primitivas, que são representadas através de desenhos em duas dimensões, conforme descrito no capítulo 05.

Uma primitiva funciona como um objeto em programação orientada a objetos, onde a palavra objeto é frequentemente usada para descrever uma entidade que possui um conjunto de dados (atributos) e de operações (comportamento) encapsuladas, que pode ser referenciada.

Este módulo também permite ao usuário definir os valores dos atributos das primitivas, para isto, o usuário deverá pressionar o botão esquerdo do *mouse* com um duplo clique sobre a primitiva que deseja realizar as definições, isto fará que seja apresentada uma caixa de diálogo para inserção dos valores dos atributos. Após as definições dos atributos das primitivas o usuário tem as opções de iniciar, executar e finalizar a simulação. A opção de iniciar, permite a criação de um novo modelo e inicialização dos atributos das primitivas. Executar, inicia a simulação, dando início às interações entre as primitivas.

E, por fim, o módulo de modelagem provê um *feedback* ao usuário, fornecendo um retorno ao usuário sobre o que está ocorrendo na simulação. Esta parte constitui a saída do ambiente Enxuto. O *feedback* é fundamental para que o usuário possa interpretar os resultados da simulação. No projeto original esta interface de saída é composta por duas fases: visualização e análise. A fase de visualização ocorrerá durante a simulação, através da animação do modelo, que permitirá um maior envolvimento do usuário com o modelo que está sendo simulado. A fase de análise ocorre após o término da simulação, onde o usuário pode analisar os resultados da simulação através de estatísticas, gráficos

e tabelas dos resultados. Parte dessa funcionalidade está em desenvolvimento em um trabalho paralelo de continuidade e não será decrita, portanto, neste documento.

6.1.2 Simulação

O módulo de simulação é responsável pelo processamento no ambiente Enxuto, tendo como entrada o modelo definido pelo usuário na fase de modelagem. Esta simulação é discreta e orientada a eventos; cada atividade inicia e termina com a ocorrência de um evento, como por exemplo falha em uma estação, término de conserto, término de produção, etc... .

Alguns objetos da simulação, primitivas que compõem o modelo, são passivos, não originando nenhuma ação por si próprios, meramente respondendo a mensagens de outros objetos da simulação. Outros objetos são ativos, realizando suas ações por iniciativa própria, sem esperar por qualquer um dos outros objetos da simulação. Tem-se ainda objetos que são classificados como objetos de controle, responsáveis pelo controle da simulação. A simulação é inteiramente realizada através de interações entre estes objetos, os objetos ativos interagem com os objetos de controle trocando mensagens, enquanto que os objetos passivos interagem com os ativos e com os de controle respondendo mensagens destes.

O módulo de simulação interage com outros módulos do ambiente Enxuto, como mostra figura 6.1. Com o módulo de modelagem, ele recebe o modelo e controles do usuário, formando sua entrada, retorna dados estatísticos obtidos e informações para a realização da animação do modelo, como o estado da simulação e resultados calculados. Com o sistema especialista ele atualiza suas bases de dados.

6.1.3 Sistema especialista

Este módulo está descrito no âmbito de outra dissertação [Borges,1997]. É um trabalho que foi desenvolvido em paralelo a este [Baranauskas et al,1997], com o objetivo de auxiliar o usuário na fase de análise e interpretação dos resultados da simulação, abordando conceitos da produção enxuta que poderiam ser utilizados na melhoria da qualidade do modelo e explicando porque uma certa alteração no modelo influenciou os resultados da simulação.

6.2 Metodologia utilizada no ambiente Enxuto

Assim como em qualquer processo de desenvolvimento de um aplicativo, há a necessidade de se definir qual ou quais as metodologias de desenvolvimento a serem seguidas. Atualmente tem se tornado cada vez mais frequente a escolha de análise, projeto e programação orientada a objetos. O seu benefício principal não está em reduzir o tempo de desenvolvimento, a metodologia orientada a objeto pode demandar mais tempo do que metodologias convencionais, mas sim em prever uma futura reutilização de código, reduzindo erros e tempo gasto em manutenções. Com esta metodologia tem-se a visão de um sistema como uma coleção de objetos que colaboram entre si para tornar possível o funcionamento do sistema.

Primeiro, um método de análise é aplicado sobre uma lista de objetos, que são localizados no vocabulário do domínio do problema. O resultado é a análise orientada a objeto, sendo usada para iniciar um projeto orientado a objeto e somente após estas fases é que pode ser realizada uma implementação com uma programação orientada a objetos. No apêndice B é mostrado os diagramas dos modelos de objeto e dinâmico da análise orientada a objetos realizada no ambiente Enxuto. Uma melhor visão do ciclo de vida da metodologia orientada a objetos, pode ser encontrada em Rumbaugh et al [1991].

6.2.1 Metodologia do módulo de modelagem

Embora a parte da simulação tenha sido desenvolvida em uma linguagem padrão a ANSI C++, podendo ser utilizada em vários sistemas operacionais, tais como OS2, UNIX, MS-DOS, Windows 95, Windows NT, etc, optou-se por utilizar o sistema operacional Windows 95, pois este dá continuidade a uma plataforma amplamente difundida - Windows 3.x sobre MS-DOS - e de custo suportável para a implantação em um chão de fábrica, além de apresentar uma interface gráfica que favorece a expressão dos elementos do domínio de chão de fábrica.

Já o ambiente de desenvolvimento utilizado na construção da parte de modelagem do ambiente Enxuto foi o **Toolbook 3.0** [Asymetrix,92]. O Toolbook é um ambiente de desenvolvimento de software composto de uma interface de manipulação direta capaz de permitir a criação de objetos tais como, objetos gráficos, botões, campos de edição, etc. Também possui uma linguagem orientada a objetos, denominada **OpenScript**, que possibilita definir o comportamento dos objetos. O Toolbook é um ambiente orientado a eventos, onde nada acontece em uma aplicação até que um evento ocorra. Muitos eventos são gerados por ações do usuário, tais como apontar para um objeto e dar um clique nele,

diferentemente do que acontece com aplicações tradicionais, que requerem que o usuário faça um conjunto específico de ações em uma certa ordem em lugar e tempo específicos. Uma aplicação Toolbook pode responder a um evento se ele ocorrer, sendo notificado que um evento ocorreu por meio de uma mensagem. As mensagens são os elos de ligação entre os eventos e as respectivas ações a serem tomadas pelo aplicativo. Cada evento gera uma ou mais mensagens que são enviadas para o objeto. Por exemplo, pressionar o botão esquerdo do *mouse* gera a mensagem `buttonDown`, liberar o mesmo botão gera a mensagem `buttonUp`, cada uma destas mensagens é enviada para o objeto que está sob o cursor do *mouse* na tela.

O Toolbook é extensível, podendo definir mensagens, funções e propriedades (atributos) para os objetos, adicionar, remover ou redefinir os comandos dos menus, trocar dados com outras aplicações, traduzir e responder mensagens de sistema e chamar funções de qualquer DLL¹. Esta última característica foi utilizada para realizar a comunicação entre o módulo Modelagem com o módulo Simulação, que é uma DLL.

O Toolbook usa a metáfora de um livro para definir uma aplicação. Assim como um livro, um aplicativo Toolbook pode conter páginas, que podem conter objetos, tais como desenhos, campos e botões. Os campos admitem textos em seu interior. Os botões são elementos gráficos que se prestam, normalmente, para fazer ações, tal como a de virar uma página de um livro quando recebe um clique do *mouse*. Dois ou mais objetos podem ser reunidos para formar um grupo, sendo que o grupo formado é considerado como um novo objeto. Uma página de um aplicativo Toolbook é composta por frente (própria página) e um fundo. Uma vez que é comum um livro ter páginas com o mesmo estilo de fundo, no Toolbook um fundo de página pode ser associado a várias páginas. Genericamente, páginas, fundos, grupos, campos, botões, janelas e desenhos são denominados de objetos. A figura 6.2 ilustra os objetos que podem ser criados no Toolbook.

Todo o objeto definido no Toolbook pode conter um *script*. Um *script* é uma coleção de um ou mais procedimentos que definem como um objeto responde às mensagens. Procedimentos podem ser escritos na linguagem *OpenScript* como *handlers*, que especificam o que deve acontecer quando uma determinada mensagem é recebida por um objeto. Sempre que uma mensagem é enviada, o Toolbook procura por um *handler* para respondê-la, começando pelo *script* do objeto que recebeu a mensagem (alvo da mensagem). Se o *script* do objeto que recebeu mensagem possui um *handler* para respondê-la o Toolbook executa este *handler* e a mensagem não mais tráfegará pela hierarquia de objetos do Toolbook,

¹DLL é abreviatura do inglês, de Dynamic Link Libraries, que são bibliotecas de funções que podem ser utilizadas dinamicamente por um aplicativo. Sendo carregadas na memória quando se deseja utilizá-las e liberadas quando não são mais necessárias.

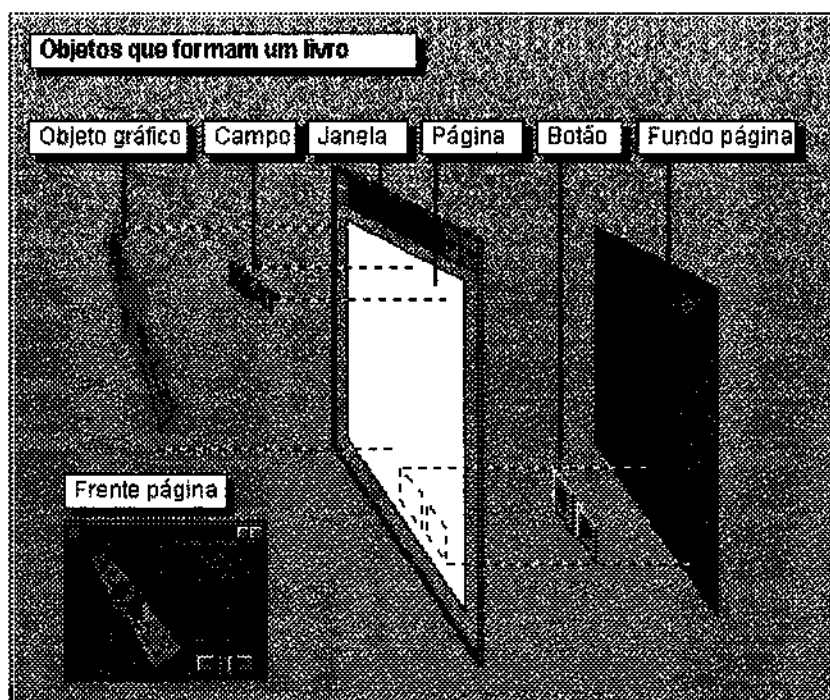


Figura 6.2: Objetos que podem ser criados no Toolbook.

ilustrada na figura 6.3, a menos que o *handler* assim especifique. Caso não exista o *handler* para responder à mensagem no *script* do objeto alvo, o Toolbook procurará por um *handler* no *script* de algum objeto, na ordem crescente da hierarquia dos objetos. O Toolbook procura por um *handler* até encontrar ou até o topo da hierarquia de objetos.

Um aplicativo Toolbook pode declarar um ou mais livros como sendo livros de sistemas. A declaração de um livro como um livro de sistema faz com que este seja adicionado na hierarquia de objetos do livro que o declarou. Livros de sistemas são ideais para definir *handlers* genéricos para serem utilizados por vários objetos de um livro ou por vários livros. Ou, ainda, eles podem ser úteis para reunir *handlers* relacionados, formando algo semelhante a uma biblioteca de funções em linguagens tradicionais.

O Toolbook constitui um meio de liberar o tempo do programador, para que este concentre-se na escrita de procedimentos que adicionem novos comportamentos a um aplicativo, uma vez que o Toolbook pode dar respostas predefinidas a diversos eventos que são irrelevantes dentro do domínio de desenvolvimento. Isto torna o Toolbook ainda mais atrativo para a sua utilização como uma ferramenta de desenvolvimento de protótipos.

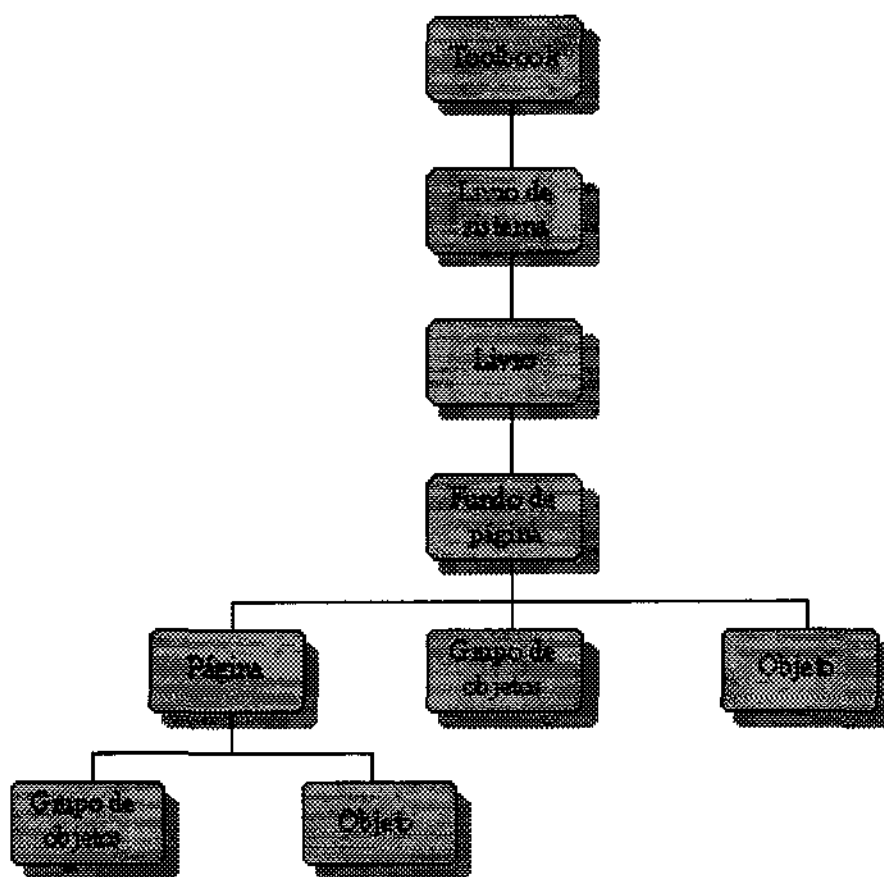


Figura 6.3: Hierarquia de objetos no ambiente Toolbook.

6.2.2 Metodologia do módulo de simulação

Um motivo que influenciou na escolha desta metodologia é que um programa de simulação imita o comportamento de objetos de sistema do mundo real, sendo portanto, natural representar estes objetos, assim como as interações entre eles, como objetos e troca de mensagens em uma programação orientada a objetos. É importante notar que as linguagens orientadas a objeto surgiram com propósitos voltados à simulação. **Simula**, foi projetada para o auxílio à programação de sistemas de simulação, mais particularmente com o intuito de simular modelos físicos, para serem usados em engenharia nuclear [Giron, 1991]. As características de orientação a objeto de **Simula** inspiraram a linguagem **Smalltalk**, que tem influenciado todas as outras linguagens orientadas a objeto.

A linguagem escolhida para implementação da simulação do ambiente Enxuto, foi C++. Diretamente influenciada pela linguagem **Simula**, foi projetada por Bjarne Stroustrup na AT&T Bell Laboratories como um super conjunto da linguagem C, provendo conceito de orientação a objeto. Um dos primeiros propósitos da linguagem C++ foi prover facilidades para implementação de sistemas de simulação [Breugnot, 1991]. Foi procurada uma linguagem eficiente, com facilidades para abstração de dados, encapsulação, herança, polimorfismo, fortemente tipada e que permitisse uma maior modularidade. C++ foi escolhida por suportar eficientemente o paradigma de orientação a objetos.

Uma outra característica desta linguagem que influenciou na sua escolha foi o conceito de funções e classes **templates**, sendo uma característica que não está disponível em outras linguagens. Este conceito permite a definição de um modelo de função ou classe, que poderá ser utilizada por vários objetos de tipos diferentes. Por exemplo:

```
template <class Tipo> class No {
protected:
    friend class Fila<Tipo>;
    Tipo          oElemento;
    No<Tipo>*      pProximo;

public:

};

template < class Tipo > class Fila {
    No<Tipo>* pPrimeiro;
```

```

    No<Tipo>* pUltimo;

protected:

public:
};

template < class Tipo > void Swap( Tipo& objetoA, Tipo& objetoB)
{
    Tipo objetoTemp = objetoA;
    objetoA          = objetoB;
    objetoB          = objetoTemp;
}

Fila<int>    oFilaInteiro;
Fila<float>  oFilaReal;

```

Esta característica facilita a implementação da simulação, pois utiliza várias estruturas de dados, tais como: filas, listas, vetores, etc, onde cada estrutura é utilizada para armazenamento de vários objetos de tipos diferentes. Portanto houve uma redução no trabalho de implementação, pois não foi necessário uma codificação das estruturas, para cada tipo de objeto a ser armazenado e assim apenas a implementação de um modelo (template) para cada estrutura, pode ser usada por objetos de qualquer tipo.

Portanto pode-se considerar que C++ provê todas as facilidades de orientação a objetos necessárias para tratar com eficiência a construção de um software complexo, sem ignorar a manutenção e os problemas de engenharia de software que possam surgir durante o desenvolvimento.

6.3 O módulo de modelagem

O módulo de modelagem é composto por:

- Uma página denominada de **modelagem**, que possibilita a criação de modelos. Através de conexões entre primitivas que formam o domínio de atuação do ambiente Enxuto.

- Uma página denominada de **primitivas**, que armazena as primitivas que compõem o domínio de modelagem do ambiente Enxuto.
- Uma palheta que possibilita ao usuário escolher alguma primitiva para ser adicionada ao modelo que está construindo.
- Um conjunto de páginas que permite o controle do usuário sobre o ambiente Enxuto, sendo composta por botões que permitem ao usuário, abrir, fechar, criar um novo modelo, assim como inicializar, executar, parar e finalizar a simulação.
- Um conjunto de páginas que formam a *interface* de entrada do ambiente Enxuto, onde o usuário pode inicializar, visualizar e alterar os valores dos atributos de cada um dos objetos que compõem o modelo por ele criado, assim como definir parâmetros da simulação, tal como o tempo total de simulação.
- Um conjunto de páginas que formam a *interface* de saída do ambiente Enxuto, mostrando os valores de todos os atributos de cada objeto do modelo que está sendo simulado, permitindo assim ao usuário ter uma visão do estado em que cada objeto se encontra durante ou no término da simulação. Também contém informações estatísticas de cada objeto, sendo coletadas no decorrer da simulação.

Este grupo de páginas descrito acima, forma um livro denominado de **Enxuto**, assimilando a metáfora de livro que o Toolbook proporciona. O *script* de cada uma de suas páginas desempenha um papel específico, relacionado à coordenação de atividades, tais como criação de modelos, edição de atributos dos objetos, controle e visualização de informações sobre a simulação.

6.4 O módulo de simulação

A simulação computacional pode ser classificada como **contínua** (analítica) ou **discreta** [Neelamkavil, 1988]. A simulação contínua utiliza o potencial da análise matemática para representar problemas que podem ser completamente entendidos ou ser analiticamente aproximados. Por exemplo, em casos onde o ambiente real que está sendo simulado, pode ser seguramente descrito por um conjunto de equações diferenciais (tal como um fluxo de calor sobre uma superfície), soluções analíticas para estas equações podem ser usadas para gerar um comportamento dependente do tempo requerido pela simulação.

Embora soluções aproximadas sejam frequentes em matemática, podem se tornar críticas e incompreensíveis. Reduzindo o ambiente real a uma abstração de relações matemáticas, elas podem tornar obscuro o entendimento deste ambiente. Existem casos em

que soluções analíticas são conhecidas, mas os possíveis métodos computacionais destas soluções não são praticáveis. Contudo a simulação contínua é indispensável em muitas situações, particularmente no caso de fenômenos físicos complexos, envolvendo um vasto número de entidades pequenas e similares, cujas interações individuais são relativamente simples e permitem um tratamento estatístico.

Porém resta uma vasta classe de problemas para os quais não existe nenhuma solução matemática formal. Usualmente estes problemas envolvem uma pequena coleção de entidades interagindo entre si. A simulação discreta é utilizada nestes tipos de problemas, onde cada entidade possui atividades, que iniciam e terminam com um evento discreto (tal como uma falha em uma estação). Dependendo de como o tempo é tratado no decorrer da simulação discreta, ela pode ser classificada como **orientada a evento**, onde o tempo é avançado discretamente de acordo com uma sucessão de estados. Transições entre estes estados são realizadas sempre que um evento relevante ocorre na interação entre as entidades. Ou como **orientada a relógio**, onde o tempo é incrementado em intervalos fixos, *ticks* de relógio, indicando a ocorrência de interações entre as entidades [Widman et al, 1989]. Para uma melhor definição do que uma simulação orientada a relógio e uma simulação orientada a evento, deve-se entender quais são os objetos que compõem a simulação discreta.

6.4.1 Objetos da simulação

Alguns objetos da simulação são **passivos**, pelo fato de que eles não originam nenhuma ação por si próprios, mas meramente respondem a mensagens de outros objetos da simulação. Há outros objetos da simulação que são **ativos**, devido ao fato que eles tomam suas ações por iniciativa própria, sem a sugestão de qualquer outro objeto. Existe ainda objetos da simulação denominados de objetos de **controle**, os quais são responsáveis pelo controle da simulação, incremento do relógio, escalonamento de objetos ativos, etc. O objeto de controle não faz parte da simulação, isto é, ele não corresponde a nenhum objeto do ambiente real que está sendo simulado. Dependendo do tipo de mensagem do objeto de controle enviada aos objetos ativos, uma simulação discreta pode ser classificada como orientada a relógio ou orientada a evento.

Em uma simulação orientada a relógio, o objeto de controle envia para todos os objetos ativos uma mensagem contendo o tempo corrente da simulação, a cada avanço de relógio. Os objetos ativos comparam o tempo corrente com o tempo no qual é suposto que ele tome alguma ação. Se o tempo corrente for igual ao tempo determinado para que alguma ação seja realizada, então a ação é executada, caso contrário ele ignora a mensagem enviada pelo objeto de controle. A figura 6.4 ilustra a interação entre os objetos na simulação

orientada a relógio. O problema com a simulação orientada a relógio é que muitas das mensagens disparadas pelo objeto de controle, contendo o tempo corrente da simulação não resultam em nenhuma ação, portanto objetos ativos desperdiçam uma boa parte do tempo computacional real determinando se o tempo da próxima ação foi alcançado.

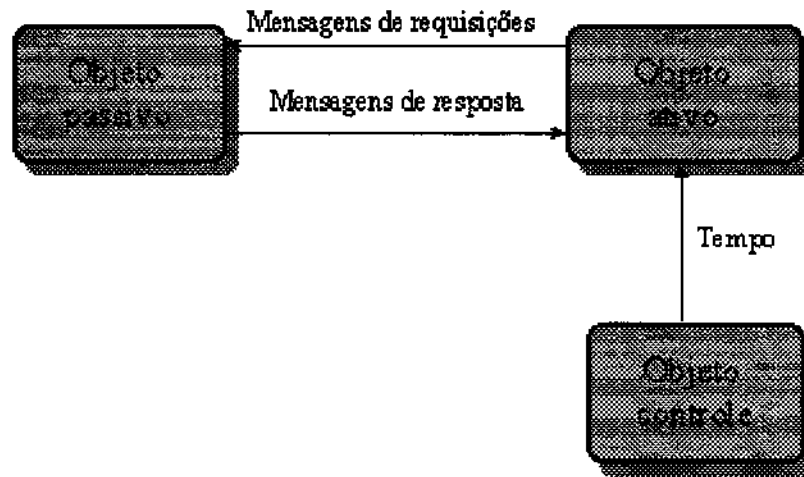


Figura 6.4: Interação entre os objetos na simulação orientada a relógio.

Uma simulação orientada a eventos, possui um objeto de controle denominado **Escala-**
donador. O objeto ativo antes de completar a sua ação corrente (evento) e ir 'dormir', ele determina o tempo em que o seu próximo evento irá ocorrer e envia uma mensagem com esta informação para o escalonador, que a recebe e escalona o evento (anota o tempo que o evento deve ocorrer) para este objeto ativo. Mais tarde, quando o tempo da simulação alcançar o tempo do evento escalonado anteriormente, o escalonador envia uma mensagem para o objeto ativo responsável por este evento, 'acordando-o' para que este possa realizar sua tarefa. Portanto um objeto ativo recebe apenas uma mensagem do objeto de controle e não uma a cada avanço do relógio. A figura 6.5 ilustra a interação entre os objetos na simulação orientada a eventos.

6.4.2 A simulação no ambiente Enxuto

A simulação no ambiente Enxuto é classificada como sendo uma simulação discreta, pois o ambiente Enxuto é formado por uma pequena coleção de objetos, onde cada objeto possui atividades que iniciam e terminam com a ocorrência de eventos discretos. Por exemplo, uma estação possui dois tipos de eventos discretos, quebra e término de produção, o técnico possui apenas um, o término de conserto, o fornecedor tem o término de fornecimento, o

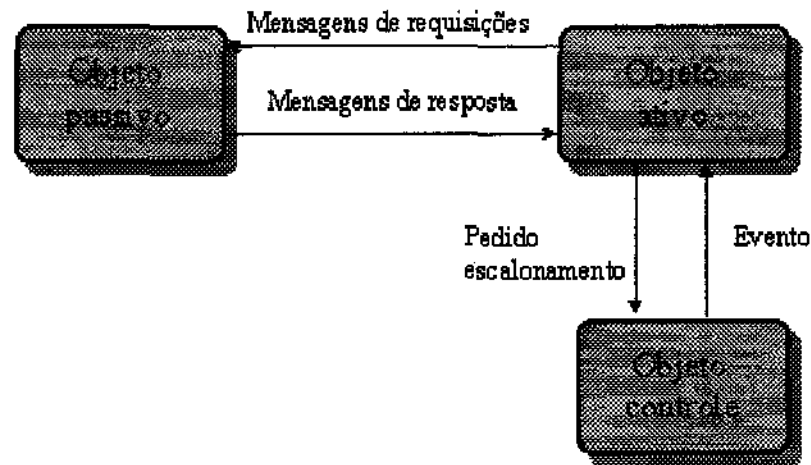


Figura 6.5: Interação entre os objetos na simulação orientada a eventos.

cliente possui o evento onde realiza o próximo pedido e por fim o conector tem o evento término de entrega.

O ambiente Enxuto utiliza a simulação discreta e orientada a eventos, uma vez que as atividades inicializadas por uma ocorrência de um evento em qualquer um dos objetos do ambiente, tem um intervalo de duração variável. Por exemplo, o término de produção em uma dada estação pode gastar um intervalo de tempo menor ou maior, dependendo da quantidade que está sendo produzida. Portanto não poderia utilizar uma simulação discreta e orientada a relógio, pois esta é indicada para a simulação de ambientes onde a duração das atividades de cada objeto ocorrem em intervalos pequenos e em uma frequência constante.

6.4.3 Objetos da simulação no ambiente Enxuto

Com o uso do paradigma orientado a objeto, o foco principal está na descrição de entidades que compõem o ambiente real. Enquanto que com o uso de paradigma procedural, tem-se o foco voltado para um procedimento global, que pode ser decomposto em várias subtarefas com o decorrer do projeto. Também enfatiza estruturas de dados, tal como filas, que são introduzidas para suportar o algoritmo a ser desenvolvido. Portanto o paradigma procedural dificulta a realização da análise do projeto, uma vez que os objetos que estão sendo manipulados não fazem parte do ambiente real, desviando os estudos do problema principal a ser solucionado, podendo acarretar erros de interpretação.

No paradigma orientado a objeto cada entidade é definida como uma abstração em termos de classe. Uma classe corresponde a definição de um tipo para uma estrutura

complexa que inclui campos para funções, assim como para dados. Entidades correntes em uma solução para o problema são representadas como instâncias destas classes abstratas.

Como descrito anteriormente os objetos que compõem uma simulação discreta são objetos ativos, passivos e de controle, conforme ilustra a figura 6.6

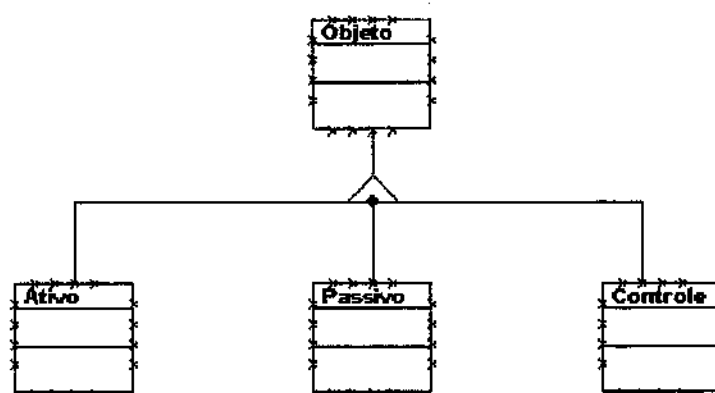


Figura 6.6: Diagrama de classe dos objetos que compõem uma simulação discreta e orientada a eventos.

Com uma análise do domínio do ambiente Enxuto, podemos identificar quais são os objetos ativos e quais são os passivos. Já que os objetos de controle são identificados de acordo com o tipo de simulação que será utilizada, estes objetos não correspondem a nenhum objeto real do domínio a ser simulado.

Para o domínio do ambiente Enxuto, foram identificados os seguintes objetos como sendo ativos: **fornecedor**, realiza a ação de entrega de matéria-prima; **conector**, faz a ação de entregar produto ou matéria-prima; **estação**, determina a ação de término de produção, assim como quando ocorrer uma quebra; **cliente** que especifica quando realizar um pedido, como também a quantidade deste pedido e por fim o **técnico** que determina quando terminar um determinado conserto em uma estação. Todas estas ações são tomadas pelos próprios objetos de forma independente dos demais. A figura 6.7 ilustra o diagrama de classe dos objetos ativos que compõem uma simulação no ambiente Enxuto.

Como objetos passivos foram identificados os seguintes objetos: **estoque**, objeto que apenas responde a mensagens que lhe são enviadas e um objeto denominado **gerenciador**, com a função de gerenciar uma fila de recursos, tais como técnicos e estações. Se existem

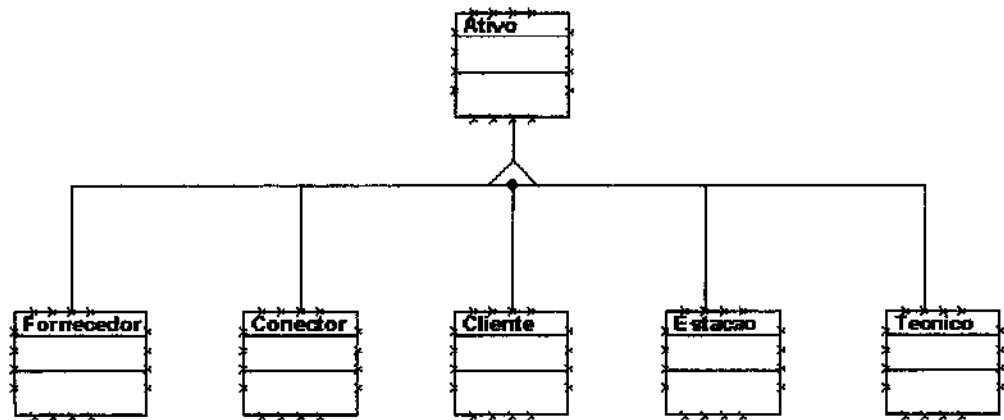


Figura 6.7: Diagrama de classe dos objetos ativos que compõem uma simulação no ambiente Enxuto.

estações esperando para serem reparadas o gerenciador mantém uma fila destas estações inoperantes e associa a estação do início da fila ao próximo técnico que se tornar disponível. Por outro lado, quando alguns técnicos não estão ocupados, o gerenciador de recursos mantém uma fila de técnicos ociosos e associa ao técnico do início da fila a próxima estação que quebrar. O gerenciador de recursos possui uma única fila que quando não está vazia ela contém apenas estações, ou apenas técnicos. A figura 6.8 ilustra o diagrama de classe dos objetos passivos que compõem uma simulação no ambiente Enxuto.

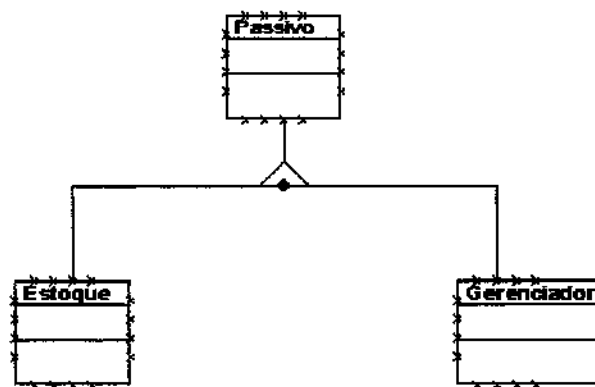


Figura 6.8: Diagrama de classe dos objetos passivos que compõem uma simulação no ambiente Enxuto.

Os objetos de controle que irão fornecer infra estrutura à simulação no ambiente Enxuto são: objeto **evento**, responsável pelo controle de eventos de um determinado objeto ativo, contendo informações sobre o identificador do evento, um apontador para o objeto ativo dono do evento e a data de ocorrência deste evento. Objeto **estado** tem a função de informar sobre um conjunto de estados que o objeto possui, o estado corrente, ou seja o estado em que o objeto se encontra, um conjunto de tempo, que acumula o tempo de cada estado em que o objeto tenha passado. Também possui um objeto denominado **requisição**, que mantém informações de uma requisição de produtos, feita pelo cliente, armazenando a data em que a requisição foi realizada, para um posterior cálculo do tempo gasto para que a requisição seja atendida.

Ainda há mais dois objetos de controle, um denominado **escalonador** que escalona e aciona estes eventos, e o objeto **distribuição**, que é responsável pelo cálculo das datas de ocorrências dos eventos dos objetos ativos. Estes objetos serão especificados a seguir.

O escalonador não simula nenhuma entidade do mundo real, mas em vez disso aciona a simulação através de mensagens **Evento**. Depois de todos os objetos ativos terem sido inicializados, todas as atividades da simulação são acionadas diretamente ou indiretamente pelo escalonador, que percorre uma lista de objetos ativos escalonados, enviando mensagens Evento a cada um.

Quando um evento é escalonado, o escalonador deve armazenar informações sobre este evento, para que posteriormente ele possa acionar este evento do objeto ativo, através da mensagem Evento na data especificada durante o escalonamento. Uma estrutura de dados faz-se necessária para a armazenagem destas informações sobre o evento. Esta estrutura de dados é denominada de **Lista de eventos** ou **Conjunto de eventos** da simulação. Um exemplo de uma lista de eventos é uma fila de prioridade, onde todo item tem uma prioridade para ser removido. O item de prioridade maior é removido primeiro, e assim por diante. Em uma lista de eventos na simulação, o tempo em que os eventos são escalonados determina suas prioridades. Eventos que devem ocorrer primeiro devem ter uma prioridade maior. Portanto eventos são removidos da lista de eventos de acordo com o seu tempo de ocorrência.

Existe um número de estruturas de dados que foram propostos para representar uma lista de eventos. Na implementação da simulação do ambiente Enxuto foi usada uma versão simplificada de uma estrutura conhecida como **Calendário de Filas** [Brown, 1998]. Essa estrutura é uma analogia de um calendário de mesa com uma página para cada dia do ano, onde os eventos podem ser anotados (escalonados) em uma sequência dentro da página apropriada - tal como agendar algum compromisso em uma data específica.

Com o passar dos dias ocorre um progresso nas páginas do calendário e em cada página encontram-se anotados todos os eventos escalonados para aquele dia.

Suponha que haja a necessidade de escalonar eventos para o próximo ano, mas o calendário do próximo ano ainda não está disponível. Se cada página for apagada, mas não rasgada, quando sua data for ultrapassada, então pode-se utilizar páginas anteriores à data corrente para registrar eventos do próximo ano. Portanto se hoje é 25 de abril e deseja-se escalonar um evento para o dia 21 de março do próximo ano, deve-se folhear as páginas calendário, de volta, até alcançar a página correspondente ao dia 21 de março e anotar o evento. Com o avanço dos dias ao longo do calendário, o seu fim será atingido e neste ponto, deve-se retornar ao seu início. Seguindo este procedimento, um mesmo calendário pode ser usado indefinidamente.

Mas há casos em que um evento irá ocorrer a mais de um ano, então cada página deve conter eventos de anos diferentes. Para tais casos terão que ser anotados na página, cada evento assim como seu respectivo ano de ocorrência e com o avanço dos dias ao longo do calendário, só serão acionados e removidos os eventos que pertencerem ao ano corrente. O código a seguir ilustra a implementação do objeto escalonador. A figura 6.9 ilustra a estrutura de dados que representa um calendário de filas de eventos.

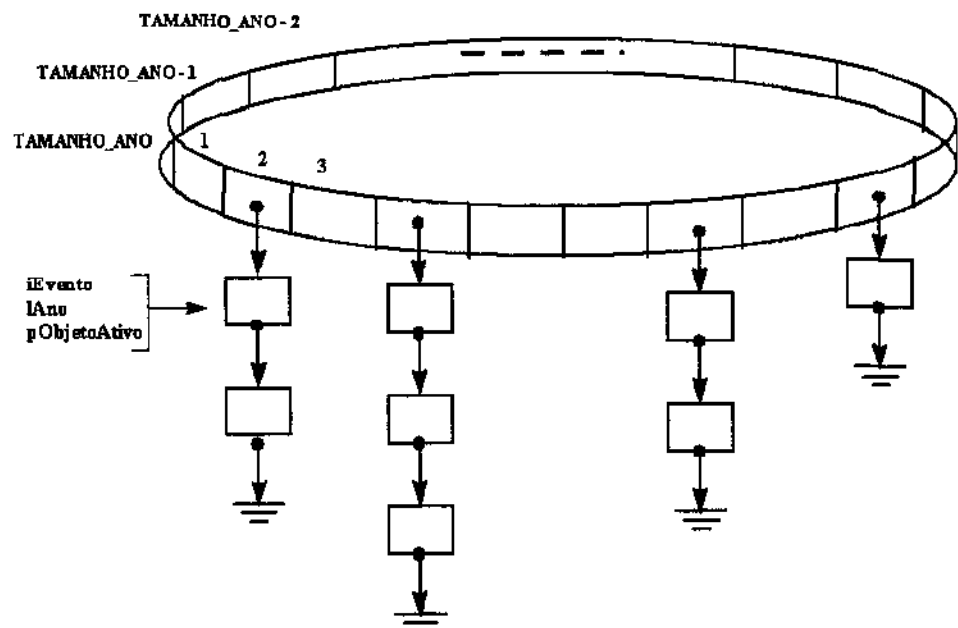


Figura 6.9: Estrutura de dados que representa um calendário de filas de eventos.

```

/*-----
Nome_____: Classe Escalonador
Definicao_: Realiza funcao gerenciar os recursos tecnico e estacao.
-----*/

class Escalonador {
    long          lRelogio;
    int           iDataCorrente;
    long          lAnoCorrente;
    int           iTamanhoAno;
    Lista<Evento>** pCalendario;

    ...

};

/*-----
Nome_____: Escalona
Efeito___: Escalona um objeto ativo para receber um evento no apos serem
           decorrido iTempo de ticks de relógio.
Modifica_: ObjetoAtivo(pCalendario,pObjetoAtivo).
Entrada__: ObjetoAtivo(pObjetoAtivo), Inteiro(iTempo, iNumeroEvento).
Saida____:
-----*/

void Escalonador::Escalona(ObjetoAtivo *pObjetoAtivo, int iDataEvento, int iNum
{
    long lAnoEvento;

    iDataEvento += iDataCorrente;
    if (iDataEvento >= iTamanhoAno) {
        lAnoEvento = lAnoCorrente + (iDataEvento / iTamanhoAno);
        iDataEvento %= iTamanhoAno;
    }
    else lAnoEvento = lAnoCorrente;
    Evento oEvento(iNumeroEvento, lAnoEvento, pObjetoAtivo);
    pCalendario[iDataEvento]->Insere(oEvento);
}

```

```

/*-----
Nome_____: ExecutaSimulacao
Efeito___: Executa a simulacao para um dado numero de passos.
Modifica_: ObjetoAtivo(pCalendario), Inteiro(lRelogio,iIndice).
Entrada__: Inteiro(iPassos).
Saida____:
-----*/

void Escalonador::ExecutaSimulacao(long lPassos)
{
for (long lCnt = 0; lCnt < lPassos; lCnt++) {
    ExaminaEventos(pCalendario[iDataCorrente]);
    lRelogio++;
    if (++iDataCorrente == iTamanhoAno) {
        iDataCorrente = 0;
        lAnoCorrente++;
    }
}
}
}

```

É claro que não há nenhum meio exato de saber quando uma estação irá quebrar, ou terminar de produzir, ou quando um cliente irá fazer um novo pedido e quantas unidades conterà este pedido. Quando os eventos dos objetos ativos devem serem simulados, usa-se números randômicos para simular o funcionamento destes eventos.

O objeto **distribuição** é que realiza o papel de uma distribuição de probabilidade de números randômicos - a probabilidade que cada número em particular ocorra - deve ser escolhida para cada problema que se está manipulando. Algumas variáveis no ambiente Enxuto que assumem valores randômicos, são geradas a partir de distribuições de probabilidades. Há variáveis de natureza quantitativa, que definem por exemplo a quantidade de tempo que uma estação gasta para produzir uma determinada quantidade de produto, ou definem a quantidade de cada pedido que o cliente faz junto à fábrica. Para estas variáveis aplica-se a distribuição exponencial, pois os números gerados por esta distribuição apresentam um comportamento quantitativo. Mas há outras variáveis que indicam quando um determinado evento deve ocorrer. Por exemplo, determinar quando uma estação irá quebrar, ou quando um cliente deverá fazer um novo pedido. A geração dos valores destas variáveis é realizada através da distribuição geométrica, que trata os resultados de evento como "sucesso" ou "insucesso". Portanto, a geração de valores randômicos no ambiente

Enxuto foi tratada através das distribuições exponencial, geométrica e uniforme, sendo esta última utilizada apenas no auxílio do cálculo da distribuição exponencial.

6.4.4 Distribuição Uniforme

Talvez a mais simples função de densidade de distribuição de probabilidades, o seu valor principal em relação às técnicas de simulação, assim como o motivo de seu uso no ambiente Enxuto, repousa na sua simplicidade e no fato de que ela pode ser usada para simular variáveis randômicas de quase todas as espécies de distribuições de probabilidade.

Para a simulação de uma distribuição uniforme em um dado domínio (a,b) tem-se a seguinte equação:

$$x = a + (b - a) * r, \text{ onde } 0 \leq r \leq 1. \quad (6.1)$$

A partir desta equação, um conjunto de números randômicos é gerado correspondendo ao domínio de probabilidade cumulativa, isto é, valores randômicos uniformes definidos no domínio $0 \leq r \leq 1$. Cada número randômico r determina unicamente um valor uniformemente distribuído x .

No ambiente Enxuto a distribuição uniforme foi utilizada apenas para auxiliar o cálculo da distribuição exponencial, onde o valor da variável r da fórmula da distribuição exponencial descrita na próxima seção é calculado através da distribuição uniforme. O código a seguir ilustra a implementação da distribuição uniforme no ambiente Enxuto.

```
/*-----
Nome_____: Uniforme
Efeito___: Escolhe um numero randomico segundo a distribuicao uniforme.
Modifica_:
Entrada__: Float(iMinimo,iMaximo).
Saida____: Float(fRandomico).
-----*/
float Distribuicao::Uniforme(float fMinimo, float fMaximo) const
{
    return (((fMaximo - fMinimo) / float(RAND_MAX + 1.0)) * float(rand()+ 1.0)) .
}
```

6.4.5 Distribuição Exponencial

Esta distribuição pode descrever um número de fenômenos físicos, tais como o tempo t que um núcleo radioativo gasta para deteriorar, ou a distância z que uma partícula viaja na atmosfera antes de sofrer uma colisão com uma molécula de água, ou o tempo x que um determinado técnico gasta para consertar um componente elétrico.

Se a probabilidade que um evento tem de ocorrer num pequeno intervalo de tempo for muito pequena, e se a ocorrência desse evento for estatisticamente independente da ocorrência de outros eventos, então o intervalo de tempo entre as suas ocorrências será exponencialmente distribuída [Neelamkavil, 1988].

Para simulação de uma distribuição exponencial tem-se a seguinte equação:

$$x = -(1/\alpha) * \log r, 0 \leq r \leq 1, \text{ onde:}$$

$$E(x) = 1/\alpha, \text{ é a esperança de } x;$$

$$V(x) = 1/\alpha^2, \text{ é a variância de } x;$$

r , é número randômico, uniformemente distribuído entre 0 e 1.

O código a seguir ilustra a implementação da distribuição exponencial no ambiente Enxuto.

```
/*-----
Nome_____: Exponencial
Efeito___: Escolhe um numero randomico segundo a distribuicao exponencial.
Modifica_:
Entrada__:
Saida____: Inteiro(iRandomico).
-----*/

int Distribuicao::Exponencial() const
{
    int iRandomico;

    for (int iCnt = 1; iCnt < MAXIMO_RANDOMICO; iCnt++) {
        iRandomico = int( -(1 / pObjetoAtivo->Alfa()) * log(Uniforme(0.0,1.0)));
        if (iRandomico > 0) return iRandomico;
    }
}
```

```
return 1;
}
```

O comportamento de um certo número de processos em alguns objetos do ambiente Enxuto, mostraram se adequar a esta distribuição. A seguir serão especificados estes objetos, assim como a definição de α para estes processos.

O objeto ativo **estação** tem a necessidade de definir quanto tempo gasta para produzir uma determinada quantidade de produtos. Este tempo é dado por:

$$T = -(1/\alpha) * \log r \quad (6.2)$$

onde α é dado como função de alguns parâmetros, tais como, quantidade a ser produzida, capacidade de produção e o número de operadores da estação. Fazendo uma proporção com estes parâmetros em relação ao tempo tem-se: uma maior quantidade para ser produzida implica em um maior tempo gasto para realizar a produção, portanto a quantidade para ser produzida é diretamente proporcional ao tempo, já a capacidade de produção e número de operadores são inversamente proporcionais ao tempo, pois quanto maior a capacidade ou o número de operadores menor será o tempo gasto para que a estação realize a produção. Sendo α inversamente proporcional ao tempo, segundo a fórmula 6.2 logo, o α pode ser definido como:

$$\alpha = (\text{capacidade de produção} * \text{número de operadores}) / \text{quantidade ser produzida}.$$

No objeto ativo **fornecedor** tem-se a necessidade de definir quanto tempo é gasto para fornecer uma determinada quantidade de matéria-prima. Também segue a distribuição exponencial só que o α é dado em função da quantidade média de matéria-prima que ele é capaz de fornecer em uma unidade de tempo e do parâmetro quantidade a ser fornecida, que é diretamente proporcional ao tempo, logo α pode ser definido como sendo:

$$\alpha = \text{média de fornecimento de matéria-prima} / \text{quantidade ser fornecida}.$$

Já em outro objeto ativo, que também segue a distribuição exponencial é o **conector**, onde o problema é definir quanto tempo ele gastará para entregar uma determinada quantidade de matéria-prima ou de produto. Considerando os parâmetros quantidade de entrega, capacidade de carga e velocidade tem-se as seguintes proporções: a quantidade de entrega é diretamente proporcional ao tempo, enquanto que a capacidade de carga e a velocidade são inversamente proporcionais ao tempo, logo o α pode ser definido como:

$$\alpha = (\text{capacidade de carga} * \text{velocidade}) / \text{quantidade de entrega}.$$

Ainda há outros dois objetos ativos que também seguem a distribuição exponencial para definição de tempos, só que estes objetos utilizam-se da esperança $E(T)$, para definir os seus α . O **técnico** tem que determinar qual será o tempo gasto no conserto de uma estação. O único parâmetro é o tempo médio de conserto, que é igual a esperança, logo o α é dado por:

$$\alpha = 1 / E(T) = (1 / \text{tempo médio de conserto}).$$

Seguindo mesmo princípio, o objeto ativo **cliente** determina qual a quantidade de cada pedido. Possuindo apenas a quantidade média de cada pedido o que é igual à esperança, o α é determinado por:

$$\alpha = 1 / E(T) = (1 / \text{quantidade média}).$$

6.4.6 Distribuição Geométrica

A distribuição geométrica trata experiências de acaso e independentes, nas quais os resultados de cada evento são expressos como "sucesso" ou "insucesso". A probabilidade de um evento ser favorável (sucesso) é geralmente denominada **p**, ($0 \leq p \leq 1$) onde **p** é considerado constante para uma sequência particular de experiências. A probabilidade de um insucesso é denominada **q**, onde $q = 1 - p$.

Os valores gerados pela contagem do número de insucessos em uma sequência de experimentos antes que ocorra o primeiro sucesso são valores de uma distribuição geométrica de probabilidades, onde sua forma é dada por:

$$f(x) = p * q^x, \text{ ou}$$

$$f(x) = p * (1 - p)^x.$$

No ambiente Enxuto há alguns objetos ativos que possuem processos cujos comportamentos seguem esta distribuição. O objeto ativo **estação** tem que definir o instante de tempo **T** que ela irá quebrar. A probabilidade que a estação quebre em um instante **T** após ser colocada em funcionamento é **p**, $T = 1$.

A probabilidade que a estação quebre no segundo instante **T** depois de ser colocada em funcionamento, é um pouco mais complicada. É a probabilidade de que a Estação quebre no segundo instante **T** e não quebre no primeiro instante **T - 1**. A probabilidade de quebrar no segundo instante é dado por **p**, e a probabilidade de não quebrar no primeiro instante

é definida por q que é igual a $1 - p$. Portanto a probabilidade de uma estação que esteja funcionando até o segundo instante e então quebre é definida por : $p * (1 - p)$, $T = 2$.

Sendo os eventos dos instantes $T = 1$ e $T = 2$ independentes, suas probabilidades são multiplicadas. Por outro lado, a probabilidade de que a estação funcione até um instante de tempo $T = t$ e então quebre é dada por: $p * (1 - p)^{(t-1)}$.

A formulação da distribuição geométrica a nível de algoritmo é ilustrada no código a seguir:

```
/*-----
Nome_____: Geometrica
Efeito___: Escolhe um numero randomico segundo a distribuicao geometrica.
Modifica_:
Entrada__:
Saida____: Inteiro(iRandomico).
-----*/
int Distribuicao::Geometrica() const
{
for (int iCnt = 1; iCnt < MAXIMO_RANDOMICO; iCnt++)
    if (rand() < pObjetoAtivo->Probabilidade()) return iCnt;
return MAXIMO_RANDOMICO;
}
```

Onde a variável `iCnt` vai contando o número de insucessos, ou seja os $t - 1$ instantes em que a estação ficou em funcionamento. E quando satisfizer a desigualdade `rand() < pObjetoAtivo->Probabilidade()` indicará o instante de tempo t em que a Estação quebra, sendo retornado o seu valor na variável `iCnt`. O cálculo da probabilidade é feito tomando todo o espaço amostral, o universo que define o intervalo de ocorrência dos números randômicos e divide este por uma média ou esperança de quebra no caso do objeto ativo estação, como é implementado abaixo.

```
/*-----
Nome_____: Probabilidade
Efeito___: Calcula a probabilidade quebra da estacao.
Modifica_:
Entrada__:
```

```

Saida____: Inteiro(fProbalidadeQuebra).
-----*/
int Estacao::Probabilidade() const
{
return int(RAND_MAX / fMmediaQuebra);
}

```

Portanto se o valor retornado por `rand()` estiver entre 1 e $RAND_MAX/fMediaQuebra$, o evento em questão (quebra) ocorre. Por outro lado, se `rand()` retornar um valor entre $RAND_MAX/fMediaQuebra$ e $RAND_MAX$ o evento não irá ocorrer. A figura 6.10 mostra a localização do intervalo de ocorrência dentro do espaço amostral.

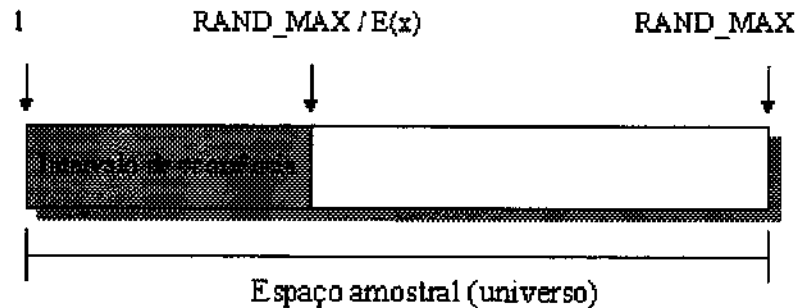


Figura 6.10: Localização do intervalo de ocorrência dentro do espaço amostral.

Um outro objeto ativo que utiliza a distribuição geométrica é o **cliente**. Ele necessita determinar quando irá realizar o próximo pedido. Segue a mesma idéia do objeto estação, só que sua probabilidade é dada em função da variável `fTempoMedioPedir` como implementado a seguir.

```

/*-----
Nome____: Probabilidade
Efeito___: Calcula a probabilidade de realizar um novo pedido.
Modifica_:
Entrada__:
Saida____: Inteiro(fProbalidade).
-----*/
int Cliente::Probabilidade() const

```

```
{  
return int(RAND_MAX / fTempoMedioPedir);  
}
```

6.5 Interação entre o módulo de modelagem e o de simulação

O módulo de modelagem interage com o de simulação, inicialmente para criar e inicializar os atributos de objetos no módulo de simulação. Estes objetos são tanto objetos que compõem o modelo criado no módulo de modelagem, que fazem parte do domínio real, como também objetos que controlam a simulação, não existindo no domínio. O módulo de modelagem também interage com o módulo de simulação para executar a simulação do modelo, por um determinado intervalo de tempo e para buscar os estados dos objetos (valores correntes dos seus atributos), dados estatísticos acumulados e os resultados finais no término da simulação. Toda esta interação ocorre por meio de chamadas de funções do módulo de simulação, pelo módulo de modelagem, como ilustra a figura 6.11.

Para executar a simulação o módulo de modelagem, executa o seguinte procedimento: de tempos em tempos chama uma função pertencente ao módulo de simulação que realiza esta tarefa. Um intervalo de tempo t entre as chamadas é definido como sendo uma parcela do tempo total T que se deseja realizar uma simulação. Decorrido o intervalo t , o módulo de modelagem atualiza os objetos, assim como pode realizar uma animação no modelo, referente às mudanças ocorridas durante este intervalo t . Depois disto o módulo de modelagem torna a chamar a função com o mesmo intervalo t , assim sucessivamente até esgotar o tempo total T . Esta função, que executa a simulação, não perde o estado da simulação, ou seja ela mantém os valores dos atributos de cada objeto do modelo referentes a última chamada da função. Quando uma nova chamada é feita ela prossegue a execução do ponto em que se encontrava anteriormente quando terminou de executar a última chamada, desde que não tenha sido realizada uma reinicialização dos objetos do módulo de simulação, o que acarretará a realização de uma nova simulação, a partir do estado inicial.

Diagramas de análise orientada a objetos do ambiente Enxuto podem ser encontrados no apêndice B.

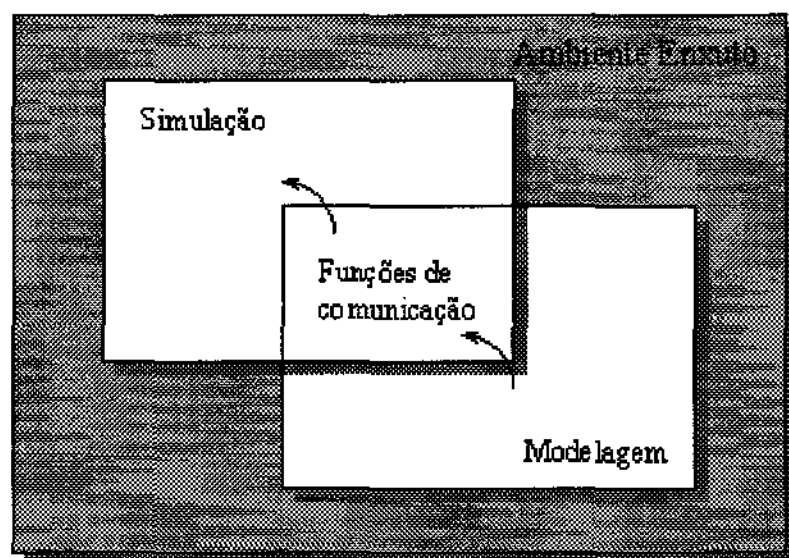


Figura 6.11: Interação entre o módulo de modelagem e o de simulação.

Capítulo 7

Um exemplo completo no ambiente Enxuto

Este capítulo descreve a simulação de um modelo no ambiente computacional de modelagem e simulação Enxuto. A simulação será ilustrada usando-se um modelo de linha de produção de uma fábrica que trabalha com o sistema de produção enxuta, embora a produção em massa também possa ser exemplificada.

7.1 Uma fábrica no sistema de produção enxuta

Este capítulo descreve algumas características que podem ser observadas pelo usuário no decorrer de uma simulação no ambiente Enxuto. Foge do escopo deste capítulo a discussão de como utilizar a interface do ambiente Enxuto para criação e a execução de modelos, visto que houve uma abordagem geral das funcionalidades do ambiente Enxuto no capítulo 05.

Utilizando uma fábrica hipotética com uma linha de produção composta por um fornecedor, um estoque de matéria-prima, duas estações de trabalho, estoque de produto e cliente, pode-se manipular determinados atributos destes elementos e verificar a respectiva influência dos mesmos no processo de produção da fábrica.

Como descrito no capítulo 05 o ambiente Enxuto suporta a execução da simulação de um modelo de uma linha de produção de uma fábrica que utiliza o sistema de produção "empurrar" ou o sistema de produção "puxar". Caso se utilize o sistema de produção empurrar o usuário poderá observar as principais características da produção em massa,

como produção em série, níveis de estoques elevados, volume de produção conforme estatísticas, etc. Já com a utilização do sistema de produção puxar, o enfoque será voltado para as características da produção enxuta, como utilização de cartões *kanban*, volume de produção *just-in-time*, níveis de estoque menores, etc. Com isto o usuário pode realizar a execução da simulação dos dois modelos de produção e observar as principais diferenças entre estes dois sistemas de produção. Neste exemplo será simulado o sistema de produção enxuta, realizando uma análise de algumas de suas principais características.

Serão usados três cenários para focar alguns tópicos que podem ser abordados no ambiente Enxuto. Todos os valores dos atributos de entrada de cada elemento do modelo e os resultados obtidos podem ser encontrados nas tabelas 7.1 a 7.4, no final do capítulo.

7.1.1 Cenário A

Este cenário tem como objetivo mostrar aos usuários do chão de fábrica que estão sendo treinados, a importância, da manutenção nas estações.

Como pode ser observado na tabela de entrada 7.1, a primeira simulação **S1** do cenário A tem inicialmente dois técnicos, cada um respectivamente com uma capacidade de realização de reparos em estações de 30 e 40 unidades de tempo por conserto. Após a execução da simulação pode-se notar em seus resultados que a Estação 1 ficou 73,55% da simulação ociosa, 1,55% produzindo, 4,85% esperando matéria prima e 20,05% consertando. Já a Estação 2, ficou 73,45% ociosa, 3,05% produzindo, 2,10% esperando matéria prima e 21,40% da simulação consertando. Pode-se notar que em toda ocorrência de quebra das estações era alocado um técnico imediatamente para realizar o reparo, pois as duas estações não ficaram no estado quebrada à espera de técnico.

Já na segunda simulação **S2** foi reduzida a quantidade de técnicos para apenas um e com uma capacidade de reparos igual a 120 unidades de tempo por conserto. Os demais atributos dos elementos foram mantidos. Com isso obteve-se resultados que mostraram que a Estação 1 ficou 18,90% ociosa, 0,50% produzindo, 1,65% esperando matéria prima, 39,30% consertando e 39,65% quebrada. Na Estação 2 teve-se os seguintes resultados 16,25% ociosa, 0,70% produzindo, 14,80% esperando matéria prima, 49,50% consertando e 18,75% quebrada. Neste caso quando uma estação quebrava, na maioria das vezes ela tinha que ir para uma fila à espera de um técnico para consertá-la.

Pode ser observado na tabela 7.4 que na primeira simulação **S1**, que o Cliente requisitou 101 unidades de produto e foi completamente atendido. Mas na segunda simulação **S2** ele requisitou 125 unidades e só foram entregues 24 pela fábrica, isto devido ao fato

das estações, no caso da segunda simulação, terem ficado quebradas a maior parte do tempo, sem ter uma manutenção adequada.

7.1.2 Cenário B

Este cenário tenta ilustrar uma fábrica atendendo as requisições de seus clientes em tempo hábil, sem atingir níveis de inventários em processos elevados, sendo uma fábrica balanceada.

Como mostrado na tabela de entrada 7.1, a primeira simulação **S1** do cenário B, mostra os containers da fábrica com os seguintes valores dos cartões *kanban*: 120, 100, 80, 60 e 2 unidades cada. Os resultados obtidos com a execução da simulação mostraram que as requisições do Cliente foram atendidas, quantidade requisitada pelo Cliente é igual a quantidade recebida. Entretanto os níveis dos inventários em processos ficaram altos. O Estoque de matéria prima ficou com 14 unidades armazenadas, a Estação 1 com 50 unidades de matéria prima e 80 de produto, a Estação 2 com 120 unidades de matéria prima e 60 de produto e o Estoque de produto ficou com 77 unidades de produtos acabados armazenados, sem ter para quem entregar. Isto faz com que haja um aumento no custo desse produto, pois haverá um gasto com a armazenagem e manutenção destes estoques na fábrica. Além disso, todo o capital agregado a estes estoques está parado.

Alterando os valores dos cartões *kanban* para 20, 20, 20, 40 e 2 unidades respectivamente na segunda simulação **S2** e observando os seus resultados nota-se que todas as requisições do Cliente também foram atendidas e que os níveis de inventários em processo diminuíram substancialmente em relação à primeira simulação. O Estoque de matéria prima passou para 10 unidades armazenadas, a Estação 1 com 20 unidades de matéria prima e 20 de produto, a Estação 2 com 20 unidades de matéria prima e 40 de produto e o Estoque de produto ficou com 26 unidades de produtos acabados.

O usuário pode notar que deve-se buscar um equilíbrio entre o tempo gasto para atender uma determinada requisição e os níveis de inventários. Pois na primeira simulação **S1** a fábrica conseguiu atender as requisições em um tempo hábil (Lead time = 122,33), mas para isto os seus níveis de inventários ficaram altos, acarretando custos em armazenagem e capital parado. Já na segunda simulação **S2** a fábrica também conseguiu atender todas as requisições com um tempo razoável (Lead time = 148,70) e os níveis de inventários ficaram baixos, diminuindo os custos de armazenagem e a quantia de capital parado.

<i>Primitiva</i>	<i>Cenário A</i>		<i>Cenário B</i>		<i>Cenário C</i>	
	S1	S2	S1	S2	S1	S2
Fornecedor						
Qtdade média (unidade/tempo)	100	100	1	1	1	1
Container 1						
Velocidade (viagens/tempo)	1	1	1	1	1	1
Capacidade carga (unidades/viagem)	100	100	1	1	1	1
Cartão kanban (unidade)	5	5	120	20	10	10
Estoque matéria prima						
Nível crítico (unidade)	1	1	1	1	1	1
Container 2						
Velocidade (viagens/tempo)	1	1	1	1	1	1
Capacidade carga (unidades/viagem)	100	100	1	1	1	1
Cartão kanban (unidade)	5	5	100	20	10	10
Estação 1						
Relação (matéria prima/produto)	1	1	1	1	1	1
Capacidade produção (unidade/tempo)	10	10	1	1	1	1
Tempo médio entre quebra (tempo)	100	100	100	100	100	100
Tempo setup (tempo)	10	10	10	10	10	10
Container 3						
Velocidade (viagens/tempo)	1	1	1	1	1	1
Capacidade carga (unidades/viagem)	100	100	1	1	1	1
Cartão kanban (unidade)	10	10	80	20	10	10
Estação 2						
Relação (matéria prima/produto)	1	1	1	1	1	1
Capacidade produção (unidade/tempo)	1	1	1	1	1	1
Tempo médio entre quebra (tempo)	100	100	100	100	100	100
Tempo setup (tempo)	10	10	10	10	10	10
Container 4						
Velocidade (viagens/tempo)	1	1	1	1	1	1
Capacidade carga (unidades/viagem)	100	100	1	1	1	1
Cartão kanban (unidade)	5	5	60	40	10	40
Estoque produto						
Nível crítico (unidade)	1	1	1	1	1	1
Container 5						
Velocidade (viagens/tempo)	1	1	1	1	1	1
Capacidade carga (unidades/viagem)	100	100	1	1	1	1
Cartão kanban (unidade)	5	5	2	2	2	2
Cliente						
Qtdade média pedido (unidade)	10	10	2	2	2	2
Tempo médio entre pedido (tempo)	200	200	400	400	400	400
Técnico 1						
Tempo médio conserto (tempo)	30	120	30	30	30	30
Técnico 2						
Tempo médio conserto (tempo)	40					

Tabela 7.1: Informações de entrada do ambiente Enxuto.

<i>Primitiva</i>	<i>Cenário A</i>		<i>Cenário B</i>		<i>Cenário C</i>	
	S1	S2	S1	S2	S1	S2
Fornecedor						
Matéria prima entregue (unidade)	167	57	714	330	72	156
Matéria prima requisitada (unidade)	167	47	714	330	72	156
Matéria prima pendente (unidade)	0	0	0	0	0	0
Ociosos (%)	98,300	99,400	97,745	99,310	99,715	99,400
Produzindo (%)	1,700	0,600	2,255	0,690	0,285	0,600
Container 1						
Número de viagens (unidade)	34	12	714	330	72	156
Qtdade transportada (unidade)	167	57	714	330	72	156
Qtdade transportando (unidade)	0	0	0	0	0	0
Qtdade esperando transporte (unidade)	0	0	0	0	0	0
Ociosos (%)	98,300	99,400	95,175	97,860	99,545	98,960
Transportando (%)	1,700	0,600	4,825	2,140	0,455	1,040
Estoque matéria prima						
Qtdade em estoque (unidade)	2	2	14	10	12	6
Qtdade média estoque (unidade)	2,11	2,03	13,10	17,00	11,90	5,97
Qtdade entregue (unidade)	165	55	700	320	60	150
Qtdade requisitada (unidade)	165	55	700	320	60	150
Qtdade pendente (unidade)	0	0	0	0	0	0
Ociosos (%)	96,600	98,800	93,345	98,245	99,535	99,02
Esperando completar nível crítico (%)	0,100	0,100	0,055	0,140	0,090	0,115
Esperando completar requisição (%)	3,300	1,100	6,700	1,615	0,375	0,865
Container 2						
Número de viagens (unidade)	33	11	700	320	60	150
Qtdade transportada (unidade)	165	55	700	320	60	150
Qtdade transportando (unidade)	0	0	0	0	0	0
Qtdade esperando transporte (unidade)	0	0	0	0	0	0
Ociosos (%)	98,350	99,450	98,630	98,960	99,815	99,280
Transportando (%)	1,650	0,550	1,370	1,040	0,185	0,720
Estação 1						
Qtdade matéria prima (unidade)	5	5	50	20	10	10
Qtdade produto (unidade)	10	10	80	20	10	10
Qtdade produzindo (unidade)	0	0	0	0	0	0
Qtdade média matéria prima (unidade)	2,99	2,99	3,04	1,04	5,49	5,49
Qtdade média produto (unidade)	9,62	9,86	74,90	19,46	9,94	9,82
Qtdade produto requisitado (unidade)	165	55	700	320	60	150
Qtdade produto produzida (unidade)	160	50	640	300	50	140
Ociosos (%)	73,550	18,900	68,730	70,835	73,370	0,400
Produzindo (%)	1,550	0,500	2,080	0,320	0,055	0,290
Esperando matéria prima (%)	4,850	1,650	4,635	2,230	0,510	70,400
Consertando (%)	20,05	39,30	20,485	22,895	21,820	22,155
Quebrada (%)	0	39,65	4,070	3,720	0,245	6,755

Tabela 7.2: Informações de saída do ambiente Enxuto.

<i>Primitiva</i>	<i>Cenário A</i>		<i>Cenário B</i>		<i>Cenário C</i>	
	S1	S2	S1	S2	S1	S2
Container 3						
Número de viagens (unidade)	15	4	480	240	40	130
Qtdade transportada (unidade)	150	40	480	240	40	130
Qtdade transportando (unidade)	0	0	0	0	0	0
Qtdade esperando transporte (unidade)	0	0	0	0	0	0
Ociosos (%)	99,250	99,800	99,000	99,220	99,950	99,640
Transportando (%)	0,750	0,200	1,000	0,780	0,050	0,360
Estação 2						
Qtdade matéria prima (unidade)	0	0	120	20	10	0
Qtdade produto (unidade)	5	0	80	40	10	10
Qtdade produzindo (unidade)	0	0	0	0	0	0
Qtdade média matéria prima (unidade)	4,99	4,99	6,04	1,04	5,49	4,49
Qtdade média produto (unidade)	4,82	3,60	57,94	39,46	9,95	32,21
Qtdade produto requisitado (unidade)	110	30	240	260	30	130
Qtdade produto produzida (unidade)	145	40	360	200	30	130
Ociosos (%)	73,450	16,250	71,335	73,550	0,170	48,965
Produzindo (%)	3,050	0,700	0,510	0,185	0,035	0,380
Esperando matéria prima (%)	2,100	14,800	1,975	1,805	73,050	19,060
Consertando (%)	21,400	49,500	21,515	19,860	20,830	26,235
Quebrada (%)	0	18,75	4,665	4,600	5,915	5,360
Container 4						
Número de viagens (unidade)	22	5	240	160	20	120
Qtdade transportada (unidade)	110	25	240	160	20	120
Qtdade transportando (unidade)	0	0	0	0	0	0
Qtdade esperando transporte (unidade)	0	0	0	0	0	0
Ociosos (%)	98,950	99,800	99,280	99,710	99,885	99,250
Transportando (%)	1,050	0,200	0,720	0,290	0,115	0,375
Estoque produto						
Qtdade em estoque (unidade)	9	1	77	26	1	17
Qtdade média estoque (unidade)	4,90	6,09	64,80	36,49	2,06	28,00
Qtdade entregue (unidade)	101	24	163	134	19	103
Qtdade requisitada (unidade)	101	125	163	134	198	103
Qtdade pendente (unidade)	0	101	0	0	179	0
Ociosos (%)	97,650	28,200	97,760	99,190	16,650	99,030
Esperando completar nível crítico (%)	0,050	0	0	0	0	0
Esperando completar requisição (%)	2,300	71,80	2,240	0,810	83,350	0,970
Container 5						
Número de viagens (unidade)	14	3	163	134	19	103
Qtdade transportada (unidade)	101	24	163	134	19	103
Qtdade transportando (unidade)	0	0	0	0	0	0
Qtdade esperando transporte (unidade)	0	0	0	0	0	0
Ociosos (%)	99,300	99,850	98,970	99,075	99,890	99,305
Transportando (%)	0,700	0,150	1,030	0,925	0,110	0,695

Tabela 7.3: Informações de saída do ambiente Enxuto (continuação).

<i>Primitiva</i>	<i>Cenário A</i>		<i>Cenário B</i>		<i>Cenário C</i>	
	S1	S2	S1	S2	S1	S2
Cliente						
Produto recebido (unidade)	101	24	163	134	19	103
Produto requisitado (unidade)	101	125	163	134	198	103
Qtdade pendente (unidade)	0	101	0	0	179	0
Qtdade média requisição (unidade)	7,03	13,26	3,12	3,03	3,18	3,10
Lead time (tempo)	14,71	81,67	122,33	148,70	1025,69	193,24
Número requisições (unidade)	14	8	52	44	62	33
Ocioso (%)	97,000	28,150	97,425	98,890	16,605	98,820
Esperando entrega (%)	3,000	71,850	2,575	1,110	83,395	1,180
Técnico 1						
Ocioso (%)	82,700					
Trabalhando (%)	17,300					
Técnico 2						
Ocioso (%)	88,800					
Trabalhando (%)	11,200					

Tabela 7.4: Informações de saída do ambiente Enxuto (continuação).

7.1.3 Cenário C

Este cenário aborda a identificação de gargalos em uma linha de produção, ou seja a identificação de um elemento que não está sendo capaz de atender suas requisições e não permitindo a vazão do fluxo de produção.

Realizando a primeira simulação **S1** deste cenário com os valores dos atributos de entrada de cada elemento mostrados na tabela 7.1 e analisando os resultados obtidos nota-se que quase todos os elementos da linha de produção estão atendendo as requisições que lhes são feitas, exceto o Estoque de produto, que teve uma quantidade de 198 unidades de produto requisitadas e só entregou 19 unidades. Portanto este elemento não está sendo capaz de puxar a produção de acordo com as requisições feitas pelo Cliente.

Mas na segunda simulação **S2** foi alterado o valor do cartão *kanban* do container anterior ao Estoque de produto para 40 unidades ao invés de 10, conforme ilustra a tabela 7.1, fazendo com que cada requisição do Estoque à Estação 2 seja quadruplicada. Os resultados obtidos com a execução desta simulação demonstraram que todos elementos atenderam a todas as requisições, não possuindo nenhum gargalo na linha de produção.

7.2 Interação com o sistema especialista Jonas

Durante a execução da simulação ocorre uma interação entre o ambiente Enxuto e o sistema especialista Jonas [Borges,1997], com objetivo de auxiliar os usuários a avaliar os resultados e entender os conceitos dos sistemas de manufatura embutidos na simulação. O Jonas atua como um consultor despertando o aprendiz para a análise dos cenários que estão ocorrendo durante a simulação, enfocando os principais conceitos dos sistemas de produção em massa e enxuta que surgem nesses cenários.

No ambiente Enxuto a análise feita nos cenários da seção anterior pode ser auxiliada pelo Jonas, que analisa os resultados obtidos da simulação e orienta o usuário na melhoria dos processos envolvidos na produção de seu modelo. Jonas também permite que o usuário questione sobre determinados conceitos ou situações que ocorrem durante a simulação.

7.3 Avaliação informal de usabilidade

A realização de uma avaliação formal da usabilidade do ambiente Enxuto foge do escopo deste trabalho, pois esta avaliação formal requer um consumo de tempo maior e o envolvimento de profissionais de outras áreas, tais como a área de educação, de psicologia, etc. No escopo deste trabalho foi usado um método informal de avaliação de usabilidade denominado de avaliação heurística, sendo um método fácil, podendo ser ensinado em um seminário de poucas horas de duração e rápido, a maioria das avaliações gasta menos de um dia [Nielsen, 1993A].

A avaliação heurística é um método usado para encontrar problemas de usabilidade no *design* de uma interface de usuário, podendo ser tratada como parte de um processo de *design* iterativo. Como a maioria dos métodos de usabilidade, pode contribuir substancialmente na construção de interfaces.

A avaliação heurística envolve um pequeno grupo de avaliadores que examinam a interface e a julgam de acordo com reconhecidos princípios de usabilidade (as heurísticas). Em geral, a avaliação heurística é difícil de ser realizada por um único indivíduo, pois uma única pessoa pode não ser capaz de encontrar todos os problemas de usabilidade em uma interface. Usando múltiplos avaliadores é possível melhorar a eficácia do método significativamente; normalmente é recomendável usar grupos de três, quatro ou até cinco avaliadores, pois com o uso de grupos maiores o ganho de informações adicionais é muito pequeno.

Cada avaliador realiza sozinho a inspeção na interface, assegurando a independência da avaliação, sem influências de outros avaliadores. Quando todos os avaliadores tiverem completado suas respectivas inspeções, então é realizada uma reunião para exposição e discussão de cada problema encontrado na interface.

Na avaliação heurística o avaliador possui um conhecimento mínimo sobre técnicas de *design* de interfaces, ficando sobre sua responsabilidade durante a avaliação, o levantamento de características da interface que não estão de acordo com as heurísticas. Em contraste, situações onde os testes são realizados com o usuário final, que não possui nenhum conhecimento sobre técnicas de *design* de interfaces, a avaliação fica sobre responsabilidade de um *designer*, que terá que interpretar as ações do usuário e deduzir aquelas que estão relacionadas com algum problema de usabilidade.

Em geral, durante uma sessão de avaliação é recomendável que o avaliador inspecione a interface e os seus elementos de diálogo várias vezes e os compare com uma lista de princípios reconhecidos de usabilidade. Estas heurísticas são regras gerais que descrevem propriedades comuns de interfaces de grande usabilidade. É interessante que a interface seja inspecionada no mínimo duas vezes, a primeira tem o objetivo de obter informações do fluxo de interação e um escopo geral do sistema. Enquanto que a segunda permitirá ao avaliador focar especificamente cada elemento da interface e sua importância dentro do sistema.

Abaixo é apresentada a lista original de heurísticas que foi desenvolvida por Molich e Nielsen (1993B).

- Diálogo simples e natural;
- Usa a linguagem do usuário;
- Minimiza a memorização por parte do usuário;
- Consistência;
- *Feedback*;
- Clareza nas opções de saídas do sistema;
- *Shortcuts* (atalhos);
- Mensagens de erro precisas e construtivas;
- Evita erros;

- Provê documentação e *help*.

O conjunto de heurísticas descritos abaixo foi derivado de uma análise de vários problemas de usabilidade feitas por Nielsen (1993B).

1. Estética e *design* mínimo: Diálogos não devem conter informações irrelevantes ou que são raramente necessárias. Toda unidade de informação extra em um diálogo compete com unidades relevantes de informação e diminui sua visibilidade.
2. Coerência entre o sistema e o mundo real: O sistema deve falar a língua do usuário, com palavras, frases e conceitos familiares ao usuário, ao invés de termos orientado ao sistema. Segue convenções do ambiente real, faz as informações aparecerem em uma ordem lógica e natural;
3. Reconhecer é melhor do que relembrar: O sistema faz objetos, ações e opções visíveis. O usuário não precisa relembrar de informações de uma parte de um diálogo em outro. Instruções para usar o sistema devem ser visíveis ou fáceis de serem recuperadas sempre que forem necessárias;
4. Consistência e padronização: Usuários não devem ter surpresas se diferentes palavras, situações ou ações significam a mesma coisa. O sistema segue uma plataforma de convenções;
5. Visibilidade do *status* do sistema: O sistema deve sempre manter os usuários informados sobre o que esta sendo feito, através de um *feedback* apropriado dentro de um tempo razoável;
6. Permite ao usuário um controle e liberdade de opções: Usuários frequentemente escolhem funções erradas e necessitam de clareza nas opções de "saídas de emergência" sem ter que atravessar um extenso diálogo. Suporta *undo* e *redo*;
7. Flexibilidade e eficiência de uso: Aceleradores, não conhecidos por usuários novatos, podem agilizar a interação com usuários experientes. O sistema pode atender as necessidades de ambos usuários inexperientes e experientes. Permite ao usuário ajustar suas ações frequentes;
8. Ajuda os usuários a reconhecer, diagnosticar e recuperar erros: Mensagens de erros devem ser expressas em linguagem plana, sem códigos, indicando o problema precisamente e sugerindo uma solução construtivamente;
9. Previne erros: a presença de mensagens explicativas é um cuidado de *design* que previne a ocorrência de problemas em primeiro lugar;

10. Documentação e *help*: Sempre é melhor que o sistema possa ser usado sem documentação, mas ela pode ser necessária. Qualquer informação deve ser facilmente encontrada e enfatizar a tarefa do usuário; uma lista de passos concretos deve ser disponibilizada e não deve ser extensa.

7.3.1 Avaliação do ambiente Enxuto

A avaliação da usabilidade da interface do ambiente Enxuto foi realizada com a participação de três avaliadores, onde somente um possui um conhecimento real do domínio em questão e os demais avaliadores atuam no desenvolvimento de interfaces. A avaliação do ambiente Enxuto consistiu de quatro fases: uma pré-avaliação, a avaliação, uma discussão dos resultados da avaliação e uma fase de verificação da veracidade dos problemas de usabilidade encontrados na fase de avaliação.

Na fase de pré-avaliação os avaliadores percorreram, independentemente, a interface do ambiente Enxuto seguindo um cenário pré-definido com o intuito de obterem um conhecimento da interface e da importância de cada item dentro do todo. Uma vez que os avaliadores sabem muito pouco sobre o domínio é necessário um cenário descrevendo os principais tópicos que o sistema se propõe a abordar. A seguir está descrito o cenário que foi apresentado aos avaliadores do ambiente Enxuto.

- Criar um modelo de uma fábrica contendo no mínimo duas primitivas de Estação;
- Interconectar as primitivas do modelo;
- Definir os atributos de entrada de cada primitiva do modelo;
- Definir o tempo total da simulação;
- Definir o método de produção que deverá ser simulado;
- Inicializar a simulação;
- Realizar um acompanhamento do *status* da simulação;
- Realizar uma análise dos resultados obtidos na simulação.

Na fase de avaliação os avaliadores tornaram a inspecionar a interface do ambiente Enxuto, compararam-na com a lista de heurísticas descrita anteriormente e anotaram os itens da interface que apresentavam problemas de usabilidade. No apêndice A é apresentado os resultados da avaliação de cada avaliador.

Posteriormente foi realizada uma reunião para discussão entre os avaliadores com o intuito de reunir os problemas de usabilidade levantados e discutir a veracidade de cada problema. A seguir são comentados sucintamente os resultados obtidos da avaliação, mostrando os pontos positivos e negativos levantados:

1. Estética e *design* mínimo

- (+) Não existe "Poluição visual" na interface, os elementos presentes são relevantes e não são extremamente chamativos.
- (+) Diálogos de entrada dos atributos das primitivas são conciso e claro.

2. Coerência entre o sistema e o mundo real

- (+) A semelhança do modelo construído com uma linha de montagem também aproxima a interface da linguagem do usuário.
- (-) O tempo total de execução da simulação representa o número de iterações, ao invés de ser expresso em minutos, hora, etc, não expressando a linguagem do usuário e sim um termo orientado ao sistema.
- (-/+) Surgiram respostas divergentes a respeito da representação gráfica das primitivas:

O uso de primitivas com funcionalidade/aparência iguais a de uma linha de produção contribui para o fácil entendimento do modelo construído.

As figuras das primitivas que representam os elementos da fábrica são pouco representativas no contexto, não seguindo convenções do ambiente real.

Estes problemas estão sendo tratados em trabalhos paralelo a este, onde estão sendo criados novos desenhos para as primitivas com uma representação mais próxima do mundo real. E a definição de uma unidade para representar o tempo total da simulação, uma vez que os cálculos são expressos em porcentagem, permitindo a definição de uma unidade sem grandes alterações.

3. Reconhecer é melhor do que lembrar

- (+) Boas opções de menu, fácil utilização das palhetas de primitivas e de seus componentes.
- (-) Os botões da barra de tarefa principal do ambiente Enxuto são pouco intuitivos.

- (-) Para execução de simulação o usuário tem que inicializar, executar e finalizar. Isto poderia ser agrupado em uma única ação, executar simulação, ficando transparente para o usuário.

4. Consistência e padronização

- (+) Segue uma plataforma de convenções, mas há alguns problemas de padronização entre a barra de ícones e as opções de menu.

5. Visibilidade do *status* do sistema

- (+) O *feedback* visual durante a simulação mantém o usuário informado do que está sendo feito, sendo um fator positivo nesse critério.
- (-) Para obtenção dos resultados finais da simulação não parece intuitivo o uso do botão direito do *mouse*.
- (-) Apresenta pouca informação sobre o que está sendo feito, não possuindo relógios e nem animação.
- (-) Não apresenta um *feedback* apropriado, na apresentação dos resultados da simulação a indicação do estado atual de cada primitiva entre os sinais < e > não é clara. E quando o usuário pede para inicializar a simulação e existem objetos com atributos entrada não preenchidos, o sistema coloca sobre esses objetos o sinal (?) de significado pouco explicativo.

Estes problemas podem ser solucionados com a criação de uma animação das primitivas, mostrando sinais que expressam o estado corrente de cada uma e a apresentação de mensagens explicativas ao invés de símbolos, sendo uma proposta de continuidade do trabalho.

6. Permite ao usuário um controle e liberdade de opções

- (+) Possui a opção "Desfazer" no menu "Editar", desfazendo a última ação realizada no editor de modelagem.
- (-) Não possui opção de "Refazer".

7. Flexibilidade e eficiência de uso

- (-) Possui poucas teclas de atalhos ou outros tipos de aceleradores.
- (-) A parte de conexão entre os objetos torna-se quase impossível dependendo da resolução gráfica utilizada. Torna-se difícil para usuários inexperientes.

8. Ajuda os usuários a reconhecer, diagnosticar e recuperar erros

- (+) Os sinais (?) sobre as primitivas que possuem atributos não preenchidos ajudam a prevenção e reconhecimento de erros.
- (-) Algumas mensagens de erros indicam em que primitivas o erro ocorreu, mas não exatamente onde ele está. Facilita o reconhecimento porém não o diagnóstico e correção do mesmo.

9. Previne erros

- (+) Os sinais (?) sobre as primitivas indicando que algo está faltando na simulação ajudam a fácil visualização de onde consertar o erro.
- (+) A entrada de dados do ambiente Enxuto mostra-se bem robusta, não deixando que o usuário entre com valores indesejados.

10. Documentação e *help*

- (-) Não possui *help on-line*.

Mas esta observação pode ser discutida, uma vez que a literatura defende que um *design* de interface deve ser refinado de forma que não seja necessário *help* para seu uso.

Na avaliação pode ser destacado que a interface do ambiente Enxuto está de acordo com a heurística 1, "estética e *design* mínimo", onde seus diálogos não contém informações irrelevantes, heurística 4, "consistência e padornização" seguindo um plataforma de convenção e heurística 9, "previne erros", apresentando mensagens explicativas e uma entrada de dados robusta.

Capítulo 8

Discussão, conclusão e proposta de continuidade do trabalho

Este capítulo discute tópicos conclusivos, de caracter geral, relacionados à experiência obtida no *design* do ambiente Enxuto.

8.1 Síntese

Este trabalho abordou a modelagem e simulação computacional como ferramentas voltadas ao treinamento de recursos humanos na indústria. Foi desenvolvido um ambiente computacional de modelagem e simulação denominado Enxuto, voltado para o domínio de uma linha de produção de uma fábrica, que permite que seus trabalhadores vivenciem novas filosofias de trabalho deste domínio.

A literatura tem mostrado consideráveis evidências dos benefícios advindos com a aplicação de ambientes de modelagem e simulação como ferramentas de treinamento. Isso porque as atividades de modelar e simular formam um rico ambiente no qual o usuário pode desenvolver habilidades na resolução de problemas, apresentação e entendimento sobre um determinado domínio. Mas a literatura tem mostrado, também, algumas deficiências dos ambientes de modelagem e simulação computacional. Um dos principais problemas é que estes ambientes são destinados a usuários no nível gerencial, ou com algum conhecimento formal em matemática. Um outro problema é a generalidade e a consequente ausência de suporte a um domínio específico de trabalho, que pode levar o usuário a sentir-se perdido com a manipulação aleatória dos vários recursos do sistema, que não fazem parte de seu cotidiano.

O ambiente apresentado é um ambiente de modelagem e simulação computacional voltado para um domínio específico. Provê ao usuário deste domínio um conjunto bem definido de fronteiras, permitindo sentir-se mais situado, uma vez que estará manipulando com entidades que fazem parte diretamente do seu trabalho [Baranauskas, et al, 1997]. A seguir são destacados algumas características do ambiente Enxuto e de outros ambientes:

Outros ambientes computacionais de modelagem e simulação:

- Permitem a criação do modelo, execução e análise dos resultados;
- Dirigidos a profissionais experientes, com conhecimento formal em matemática;
- Destinados a usuários de nível gerencial;
- Permite a modelagem e simulação em diferentes domínios do conhecimento;
- A generalidade pode levar o usuário a se sentir perdido.

Ambiente Enxuto:

- Também permite a criação do modelo, execução e análise dos resultados;
- Dirigidos aprendizes inexperientes, sem conhecimento formal em matemática;
- Destinados a funcionários do chão de fábrica;
- Permite a modelagem e simulação em um domínio específico;
- É composto por um conjunto de primitivas que fazem parte do domínio.

Este trabalho pode ser destacado pela interação entre indústria e universidade, propiciando várias vantagens para ambas. Para a universidade esta interação possibilita um maior conhecimento do ambiente industrial, de suas necessidades, o perfil de seus profissionais e suas tendências, possibilitando um direcionamento de suas atividades com o intuito de melhorar uma de suas funções sociais que é a formação de profissionais. O ambiente industrial também representa um vasto campo de pesquisa pouco explorado pelas universidades. Já para a indústria a universidade representa uma fonte de projetos estratégicos que poderão torna-lá mais competitiva e dinâmica na absorção de novas tecnologias.

8.2 O processo de design do ambiente Enxuto

Na fase de *design* houve uma constante interação com a fábrica buscando uma maior colaboração por parte do usuário, com o objetivo de identificar as características relevantes que deveriam fazer parte do ambiente Enxuto. Através de reuniões realizadas com os usuários, foram apresentados protótipos contendo características do domínio para serem discutidas e analisadas. O resultado destas reuniões foi um conjunto de características e requisitos importantes que foram incorporadas no ambiente computacional de modelagem e simulação para treinamento de usuários do chão de fábrica.

No desenvolvimento do ambiente Enxuto foi utilizada a metodologia orientada a objetos, tendo sido de fundamental importância no desenvolvimento do ambiente Enxuto. O uso desta metodologia no desenvolvimento de um software provê uma base sólida para a sua especificação, projeto e implementação, uma vez que toda análise é realizada sobre objetos que fazem parte do mundo real. O *designer* mantém a sua atenção direcionada sobre o problema, o resultado é um software mais compreensível, de fácil evolução e manutenção.

A análise orientada a objetos (AOO) permite o entendimento e a especificação da aplicação dentro do domínio na qual ela atua. No desenvolvimento de um software para modelagem e simulação de um domínio específico tem-se a necessidade de definição de um conjunto de primitivas que permita criar modelos dentro deste domínio. A implementação desta definição é bastante facilitada com a criação do modelo de objetos da AOO, que permitiu definir com precisão todos os objetos do domínio em questão.

O ambiente Enxuto é um ambiente de simulação discreta e orientada a eventos, onde há uma interação constante entre os seus elementos através da ocorrência de eventos. A especificação destes eventos tornou-se bastante prática com a definição do modelo dinâmico da AOO, que mostra as possíveis sequências de eventos para cada objeto através do seu diagrama de estado.

Uma parte que pode ser destacada no desenvolvimento do ambiente Enxuto foi mapear todos os requisitos e características levantadas, para os diagramas do modelo de objeto e dinâmico da análise orientada a objetos, envolvendo um estudo detalhado dos objetos envolvidos. Estes diagramas proporcionaram uma grande facilidade na fase de implementação do ambiente Enxuto. Tendo apenas um trabalho de tradução dos diagramas do modelo de objeto em criação de estruturas de dados e classes de objetos, assim como os diagramas do modelo dinâmico em linhas de código da linguagem de programação orientada a objetos usada.

8.3 Trabalhos futuros

O ambiente Enxuto, permite a inclusão de novas características, algumas das quais já estão sendo realizadas por outros trabalhos ou podem vir a serem realizadas em trabalhos futuros.

Uma das principais características que poderá ser adicionada ao ambiente Enxuto é a realização de animação sobre os elementos do modelo. Por exemplo é interessante haver uma animação dos *containers* quando os mesmos estão fazendo o transporte de matéria-prima ou produto entre os demais elementos do modelo. Esta animação possibilitaria uma visualização da diferença de dinâmica entre os sistemas de produção empurrar e puxar. A animação dos *containers* no sistema de produção empurrar mostrará um fluxo de matéria-prima ou produto se deslocando do fornecedor em direção ao cliente. Já no sistema de produção puxar seria mostrado um fluxo de matéria-prima ou produto sendo puxado pelo cliente, que busca os produtos do estoque de matéria-prima, o qual busca da última estação da linha de produção e assim sucessivamente até atingir o fornecedor.

Um outro aspecto de animação que poderia ser implementada no ambiente Enxuto seria a utilização de "caixas tanques" que representariam a variação dos inventários em processos dos elementos da linha da produção. Um elemento que estivesse com o nível de inventário em processo elevado, sua caixa tanque estaria com um conteúdo cheio, caso contrário, níveis menores, sua caixa tanque teria um conteúdo menor. Esta animação permitiria a identificação visual de gargalos no decorrer da linha de produção. Isso porque se um determinado elemento da linha de produção apresentasse um nível de inventário em processo alto, com os elementos subsequentes possuindo níveis de inventários baixos e um grande número de requisições pendentes, identificaria que este elemento não está sendo capaz de dar vazão ao fluxo de produção. Esta animação também permitiria a visualização de como se comportariam os níveis de estoques (estoques de matéria-prima, inventários em processo e estoques de produto) da linha de produção no decorrer da simulação.

Uma outra característica de animação que poderia ser incluída no ambiente Enxuto seria alguma representação gráfica que mostrasse em que estado o elemento está no decorrer da simulação. Por exemplo, uma estação no estado "produzindo" poderia ser representada por um sinal gráfico verde, no estado "quebrada" poderia ser representada por um sinal gráfico vermelho, em analogia ao sinal de trânsito por exemplo.

Existem outros trabalhos em paralelo que estão dando continuidade ao projeto inicial do ambiente Enxuto adicionando novas características, como por exemplo, permitir a abertura de mais de um modelo dentro do ambiente Enxuto. Com isto o usuário poderá

realizar a execução da simulação destes modelos, comparando os resultados obtidos e o desempenho de cada modelo aberto. Estão sendo realizados alguns ajustes na interface de manipulação direta do ambiente Enxuto com o intuito de facilitar a realização de conexões entre as primitivas do modelo. Além disso está sendo adicionada uma nova primitiva ao conjunto de primitivas do ambiente Enxuto, que permitirá modelar e simular o espaço ocupado pelos inventários em processo dos elementos da linha de produção no chão de fábrica. Isto dará ao usuário uma noção de quanto da área da fábrica fica ocupado com os inventários em processo.

Com a realização de uma avaliação informal, como a avaliação heurística, não se pode esperar que ela trate todos os problemas de usabilidade do sistema avaliado, uma vez que os avaliadores não possuem o conhecimento dos usuários reais e nem de suas tarefas. Estes dois aspectos são elementos críticos para a avaliação do sistema. Entretanto, a avaliação heurística serviu para detectar muitos problemas de usabilidade, que estão sendo tratados nas extensões em andamento.

A avaliação heurística pode ser complementada com a realização de observação do uso no ambiente real. Deve-se enfatizar a importância da realização de uma avaliação formal, ampla, que seja conduzida no próprio ambiente da fábrica, com a participação de um grupo de usuários do sistema. Os novos problemas de usabilidade apontados por esta avaliação formal estarão relacionados mais especificamente com o domínio de atuação. A realização desta avaliação formal no ambiente Enxuto, prevista como continuidade a este trabalho pode refina-lo e tornar sua interface ainda mais próxima das necessidades do usuário.

Apêndice A

Análise heurística

A.1 Observações do avaliador A

1. Estética e *design* mínimo

(+) A interface está de acordo com esta heurística.

2. Coerência entre o sistema e o mundo real

(-) O tempo total de execução da simulação representa o número de iterações, ao invés de ser expresso em minutos, hora, etc, não expressando a linguagem do usuário e sim um termo orientado ao sistema.

(-) As figuras das primitivas que representam os elementos da fábrica são pouco representativas no contexto, não seguindo convenções do ambiente real.

3. Reconhecer é melhor do que relembrar

(-) Para execução de simulação o usuário tem que inicializar, executar e finalizar. Isto poderia ser agrupado em uma única ação, executar simulação, ficando transparente para o usuário.

4. Consistência e padronização

(+) A interface está de acordo com esta heurística.

5. Visibilidade do *status* do sistema

(-) Apresenta pouca informação sobre o que está sendo feito, não possuindo relógios e nem animação.

- (-) Não apresenta um *feedback* apropriado, na apresentação dos resultados da simulação a indicação do estado atual de cada primitiva entre os sinais < e > não é clara. E quando o usuário pede para inicializar a simulação e existem objetos com atributos entrada não preenchidos, o sistema coloca sobre esses objetos o sinal (?) de significado pouco explicativo.
- 6. Permite ao usuário um controle e liberdade de opções
 - (-) A maioria das ações não suportam *undo* e *redo*.
- 7. Flexibilidade e eficiência de uso
 - (-) Possui poucas teclas de atalhos ou outros tipos de aceleradores.
- 8. Ajuda os usuários a reconhecer, diagnosticar e recuperar erros
 - (+) A interface está de acordo com esta heurística.
- 9. Previne erros
 - (+) A interface está de acordo com esta heurística.
- 10. Documentação e *help*
 - (-) Não possui *help on-line*.

A.2 Observações do avaliador B

1. Estética e *design* mínimo
 - (+) Não existe "Poluição visual" na interface, os elementos presentes são relevantes e não são extremamente chamativos.
 - (+) Diálogos de entrada dos atributos das primitivas são concisos e claros.
2. Coerência entre o sistema e o mundo real
 - (+) A semelhança do modelo construído com uma linha de montagem também aproxima a interface da linguagem do usuário.
 - (+) O uso de primitivas com funcionalidade/aparência iguais a de uma linha de produção contribui para o fácil entendimento do modelo construído.

3. Reconhecer é melhor do que relembrar

- (+) Boas opções de menu, fácil utilização da palheta de primitivas e de seus componentes.
- (-) Os botões da barra de tarefa principal do ambiente Enxuto são pouco intuitivos.

4. Consistência e padronização

- (+) A interface está de acordo com esta heurística.

5. Visibilidade do *status* do sistema

- (+) O *feedback* visual durante a simulação mantém o usuário informado do que está sendo feito, sendo um fator positivo nesse critério.
- (-) Para obtenção dos resultados finais da simulação não parece intuitivo o uso do botão direito do *mouse*.

6. Permite ao usuário um controle e liberdade de opções

- (+) Possui a opção "Desfazer" no menu "Editar", desfazendo a última ação realizada no editor de modelagem.
- (-) Não possui opção de "Refazer".

7. Flexibilidade e eficiência de uso

- (-) Possui poucas *hotkeys*.

8. Ajuda os usuários a reconhecer, diagnosticar e recuperar erros

- (+) Os sinais (?) sobre as primitivas que possuem atributos não preenchidos ajudam a prevenção e reconhecimento de erros.
- (-) Algumas mensagens de erros indicam em que primitivas o erro ocorreu, mas não exatamente onde ele está. Facilita o reconhecimento porém não o diagnóstico e correção do mesmo.

9. Previne erros

- (+) Os sinais (?) sobre as primitivas indicando que algo está faltando na simulação ajudam a fácil visualização de onde consertar o erro.

10. Documentação e *help*

- (+) Apesar de não estar implementado o projeto preve um *help* e outros componentes de auxílio a utilização do sistema.

A.3 Observações do avaliador C

1. Estética e *design* mínimo

(+) A interface está de acordo com esta heurística.

2. Coerência entre o sistema e o mundo real

(-) As figuras que representam as primitivas devem ser trocadas por outras que tenham uma melhor representatividade do mundo real.

3. Reconhecer é melhor do que relembrar

(-) Algumas ações que devem ser tomadas para executar comandos no ambiente não são usuais.

4. Consistência e padronização

(+) Segue uma plataforma de convenções, mas há alguns problemas de padronização entre a barra de ícones e as opções de menu.

5. Visibilidade do *status* do sistema

(-) É difícil para o usuário saber se a simulação começou, está acontecendo ou já terminou. A inclusão de um relógio seria válida.

6. Permite ao usuário um controle e liberdade de opções

(+) A interface está de acordo com esta heurística.

7. Flexibilidade e eficiência de uso

(-) *Shortcuts* conhecidas deveriam ser inseridas no ambiente.

(-) A parte de conexão entre os objetos torna-se quase impossível dependendo da resolução gráfica utilizada. Torna-se difícil para usuários inexperientes.

8. Ajuda os usuários a reconhecer, diagnosticar e recuperar erros

(+) A interface está de acordo com esta heurística.

9. Previne erros

- (+) A entrada de dados do ambiente Enxuto mostra-se bem robusta, não deixando que o usuário entre com valores indesejados.

10. Documentação e *help*

- (-) Não possui *help on-line*, devendo ser inserido preferencialmente na forma de hipertexto.

Apêndice B

Diagramas da análise orientada a objetos

B.1 Modelo de objeto e Modelo dinâmico

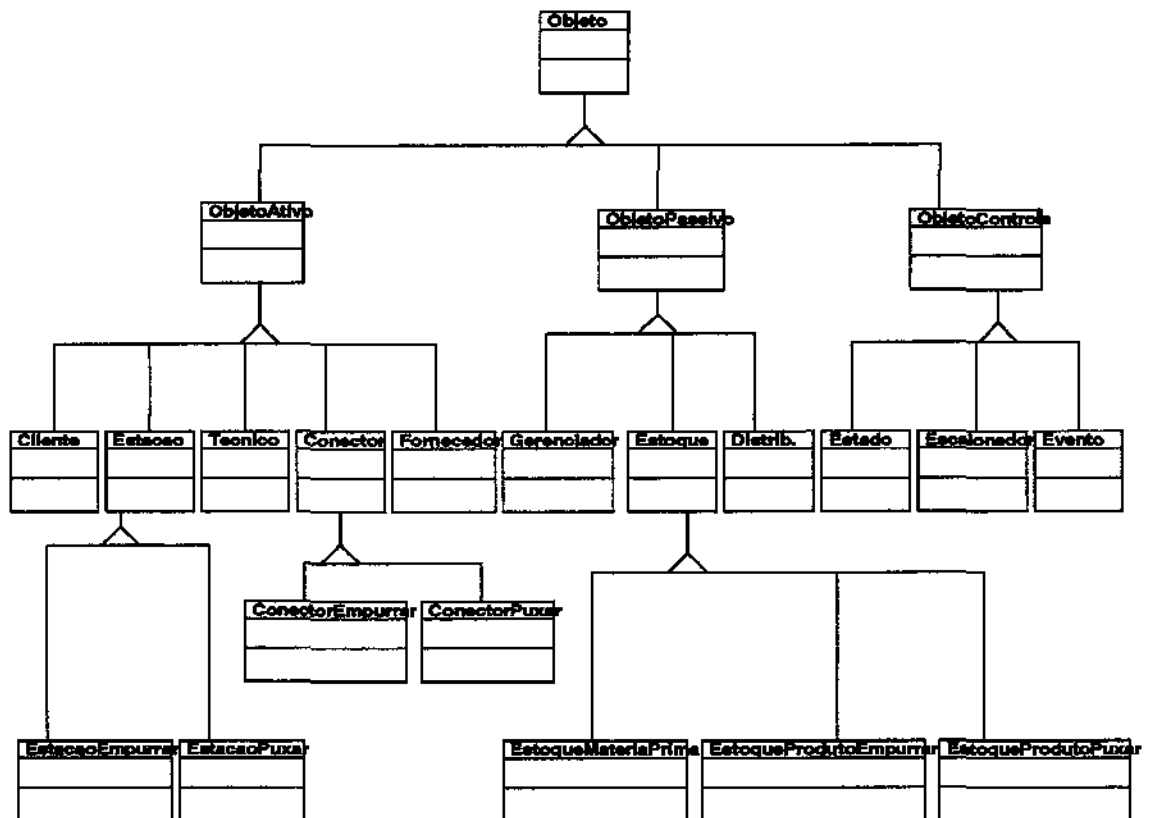


Figura B.1: Modelo de objeto do ambiente Enxuto.

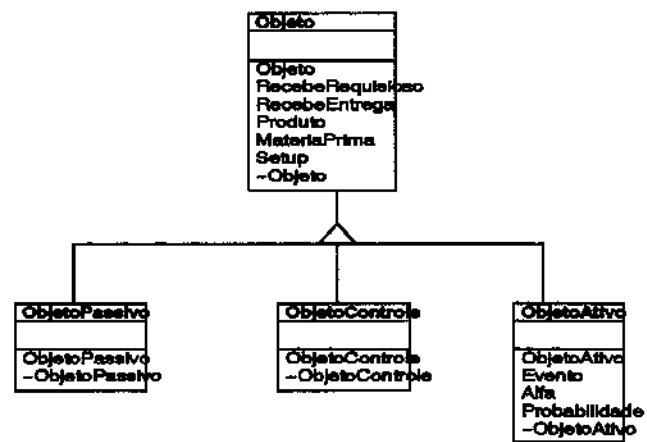


Figura B.2: Modelo de objeto da classe base.

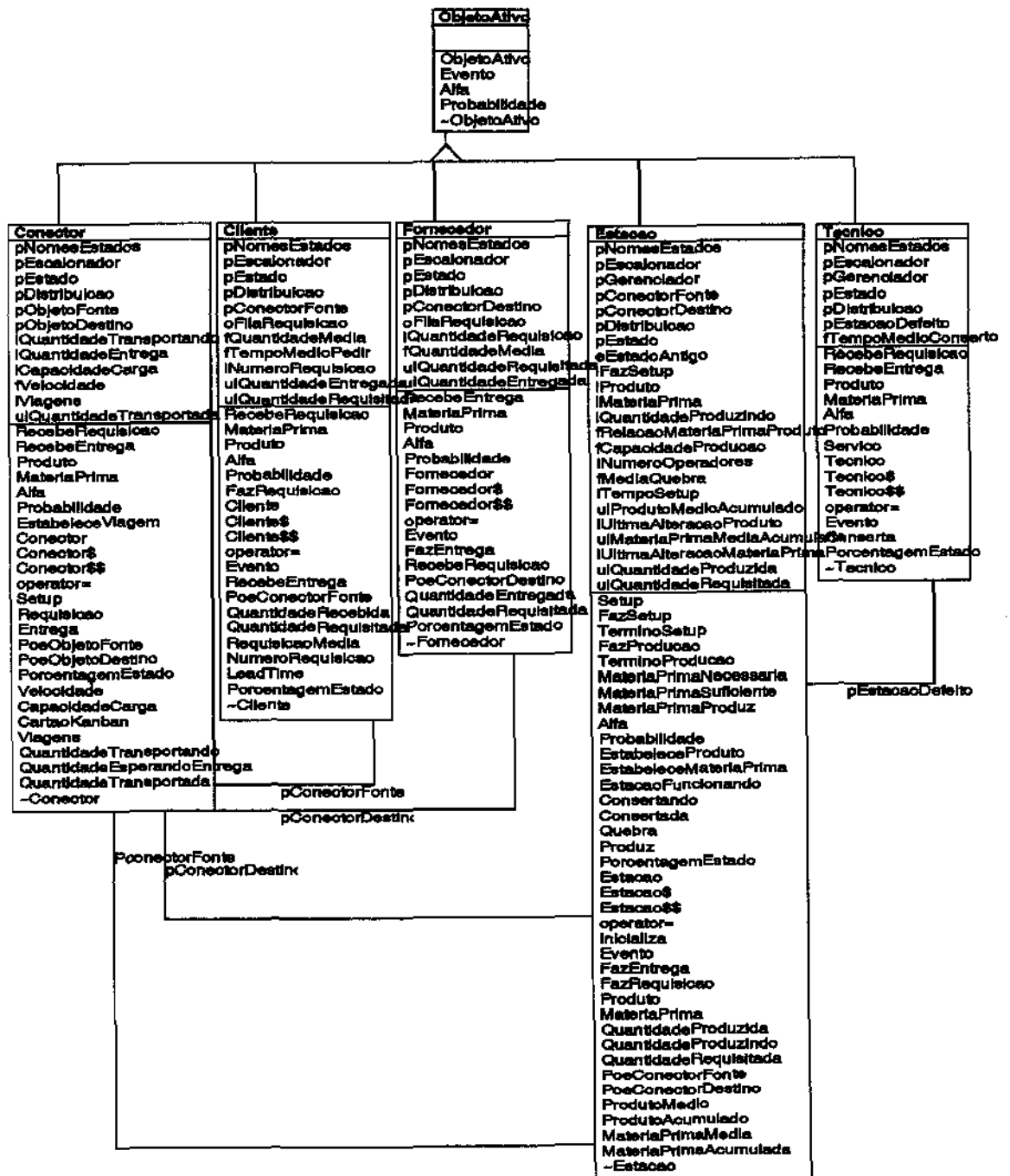


Figura B.3: Modelo de objeto da classe objeto ativo.

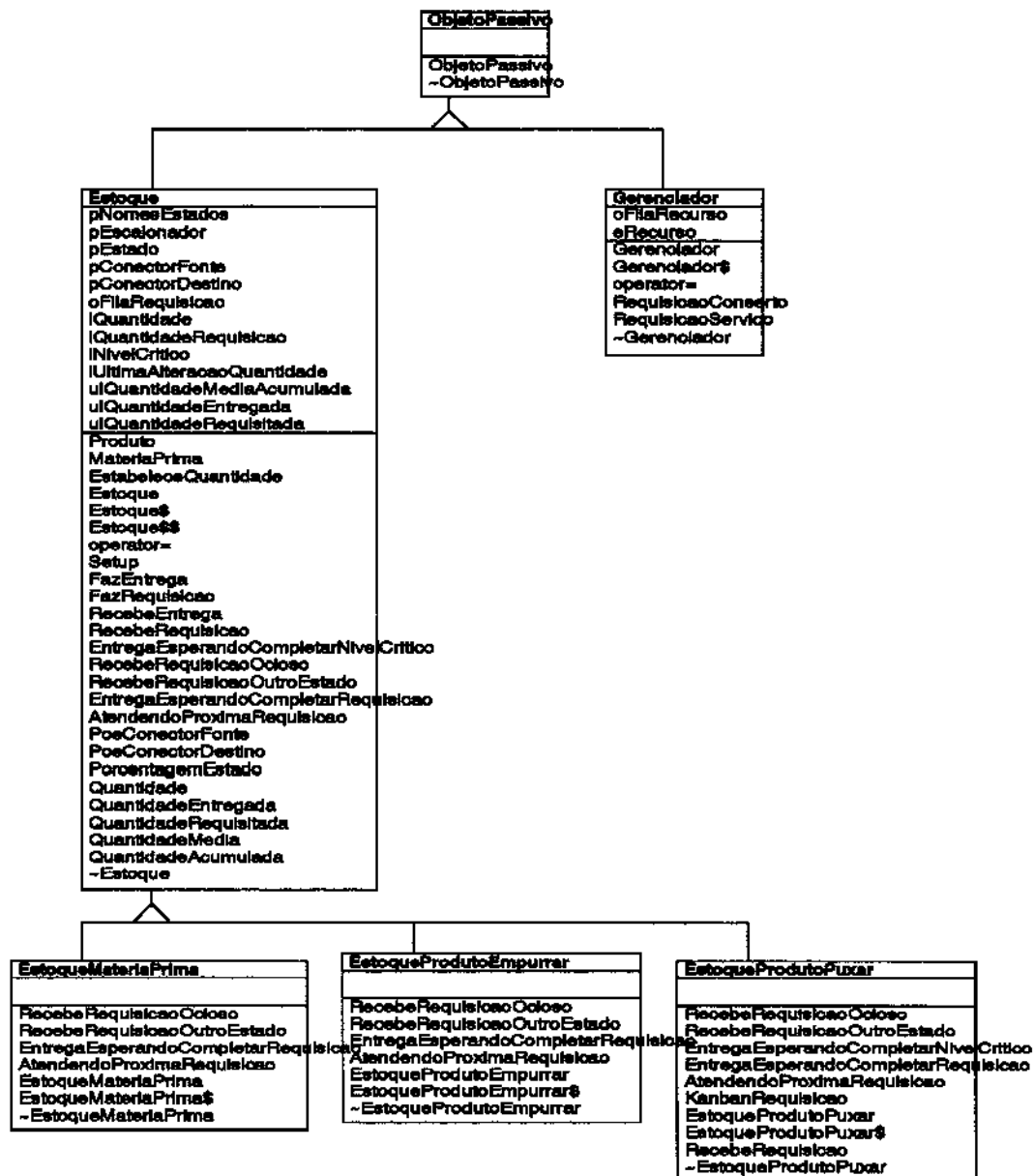


Figura B.4: Modelo de objeto da classe objeto passivo.

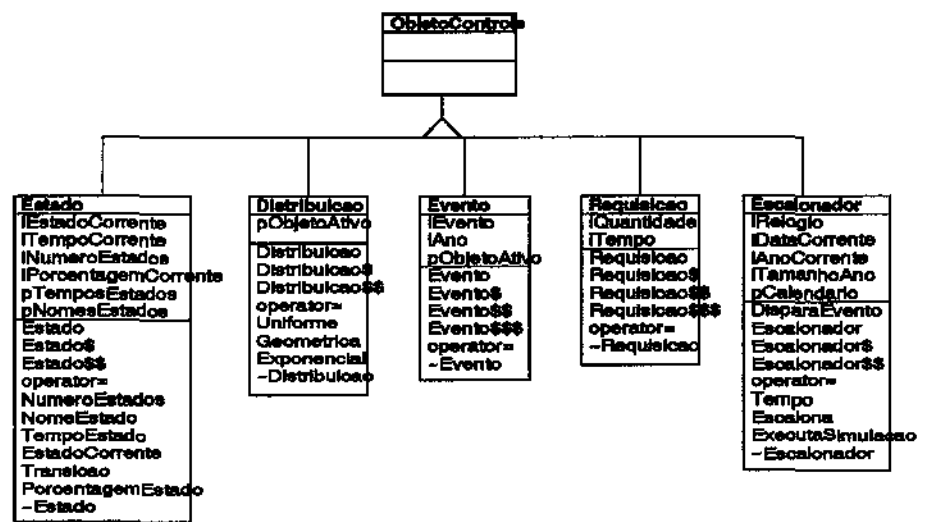


Figura B.5: Modelo de objeto da classe objeto controle.

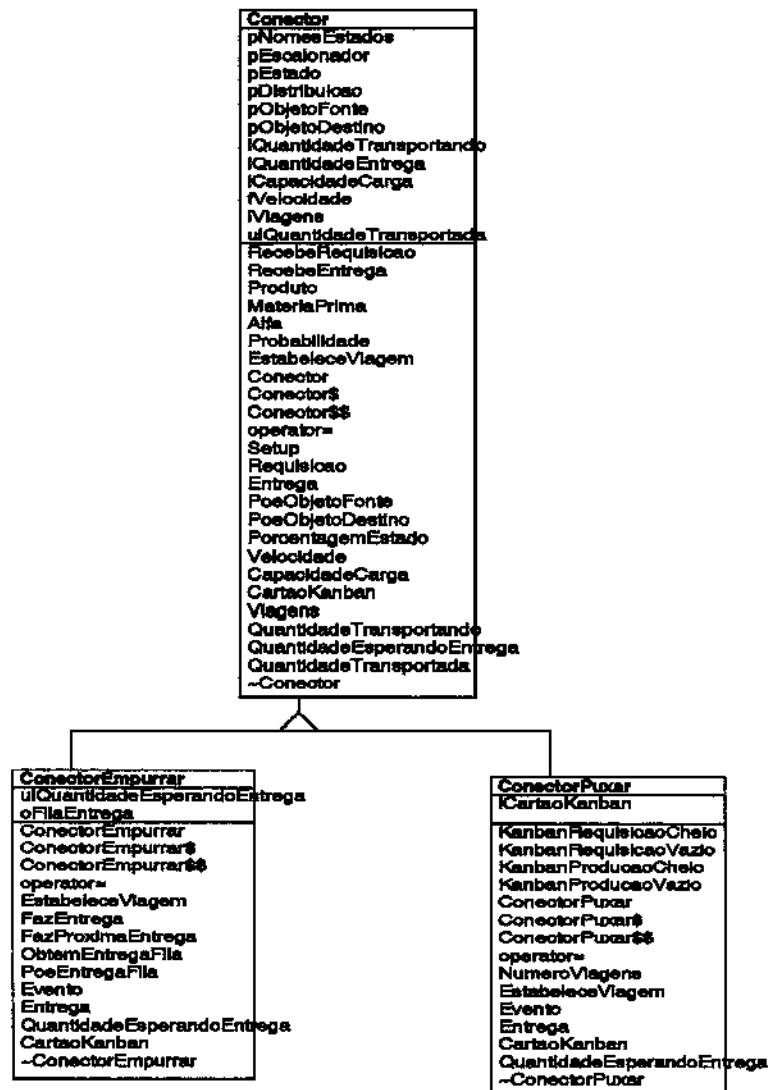


Figura B.6: Modelo de objeto da classe objeto conector.

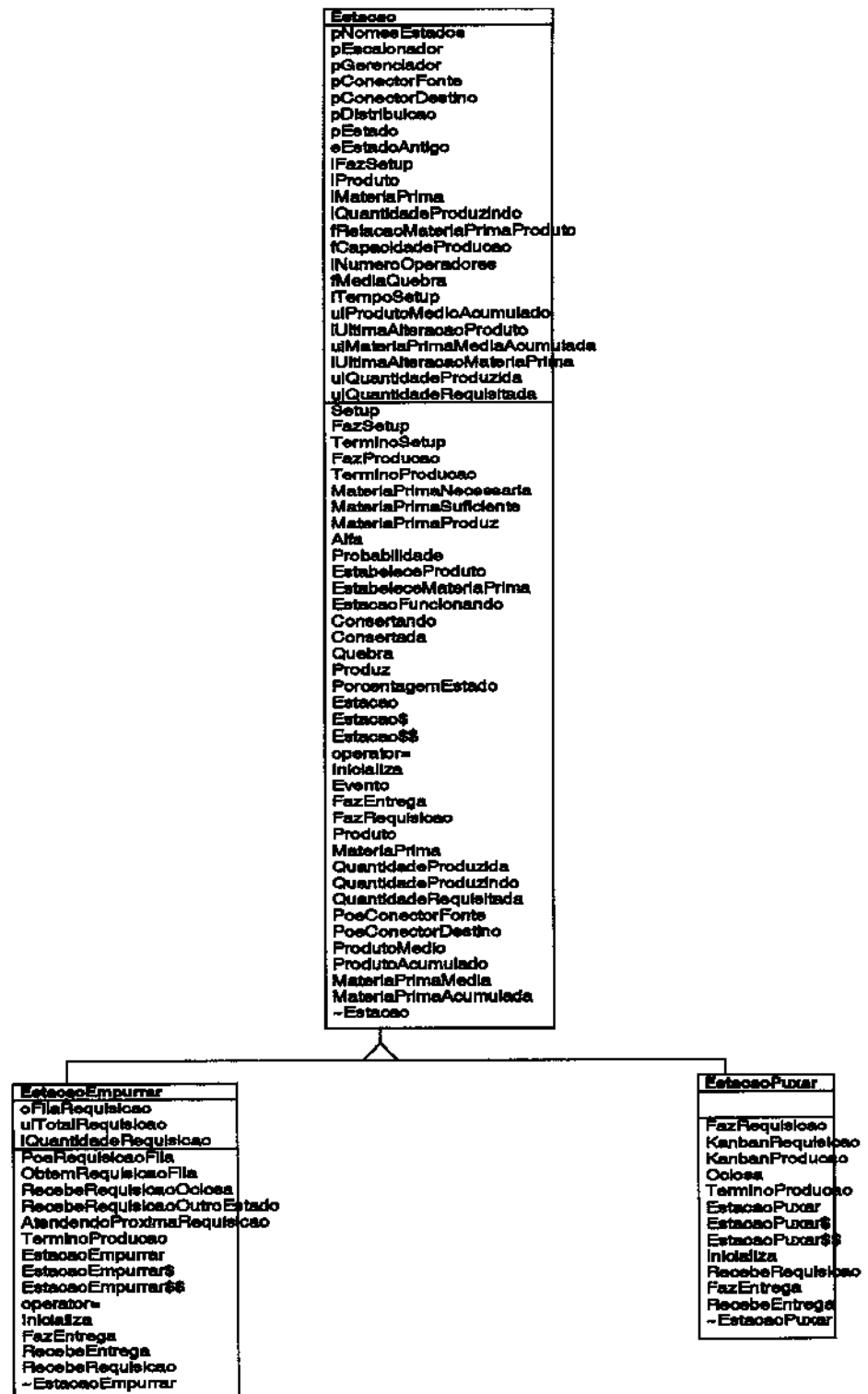


Figura B.7: Modelo de objeto da classe objeto estação.

Figura B.8: Modelo de objeto da classe Fábrica.

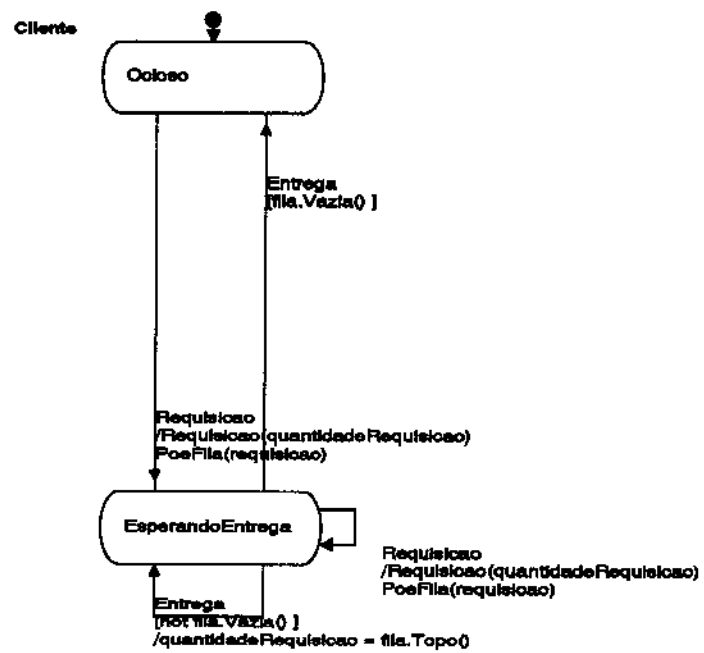


Figura B.9: Modelo dinâmico do objeto Cliente.

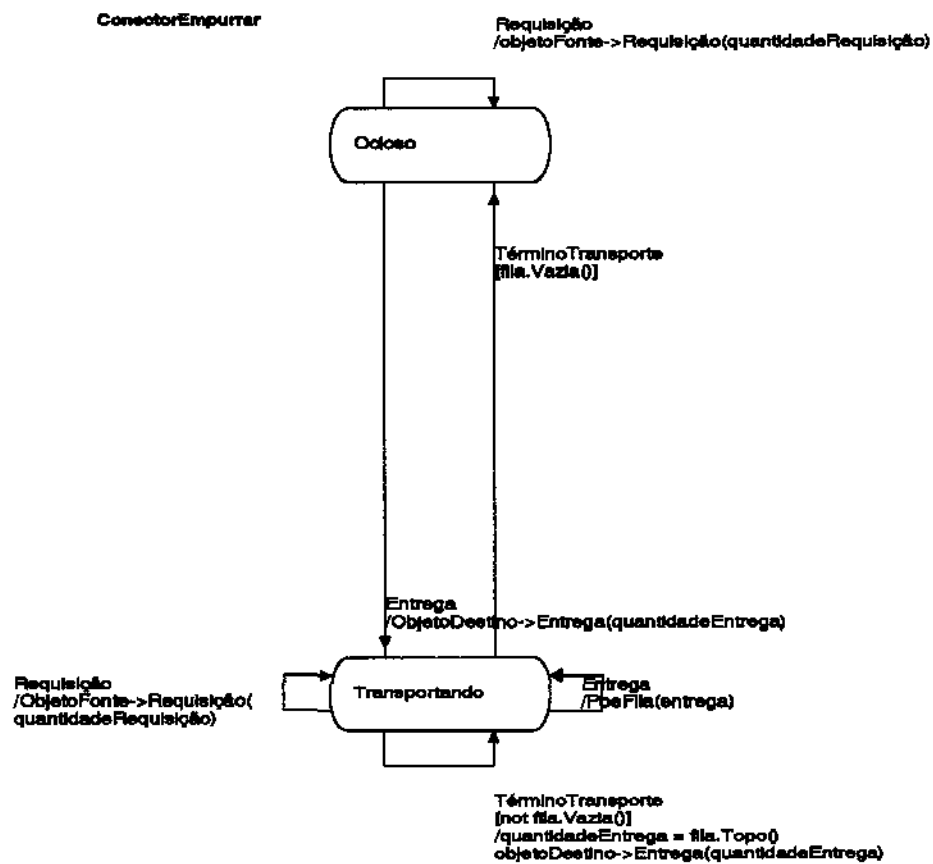


Figura B.10: Modelo dinâmico do objeto Conector Empurrar.

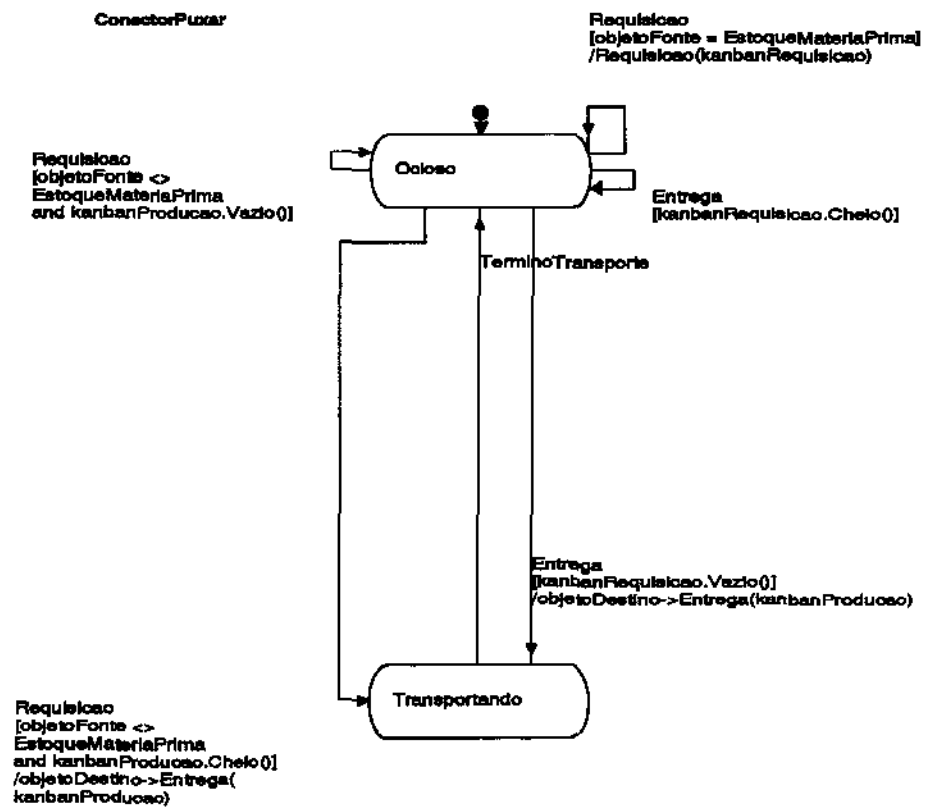


Figura B.11: Modelo dinâmico do objeto Conector Puxar.

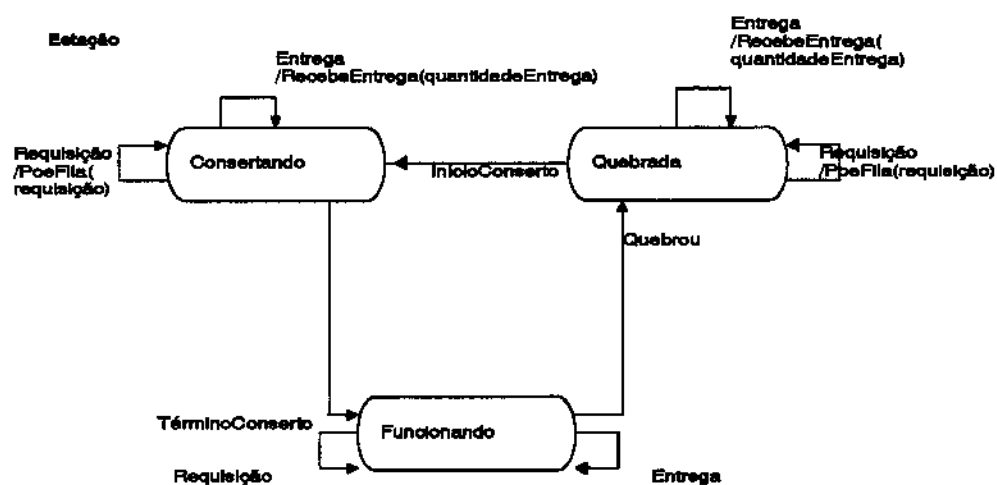


Figura B.12: Modelo dinâmico do objeto Estação.

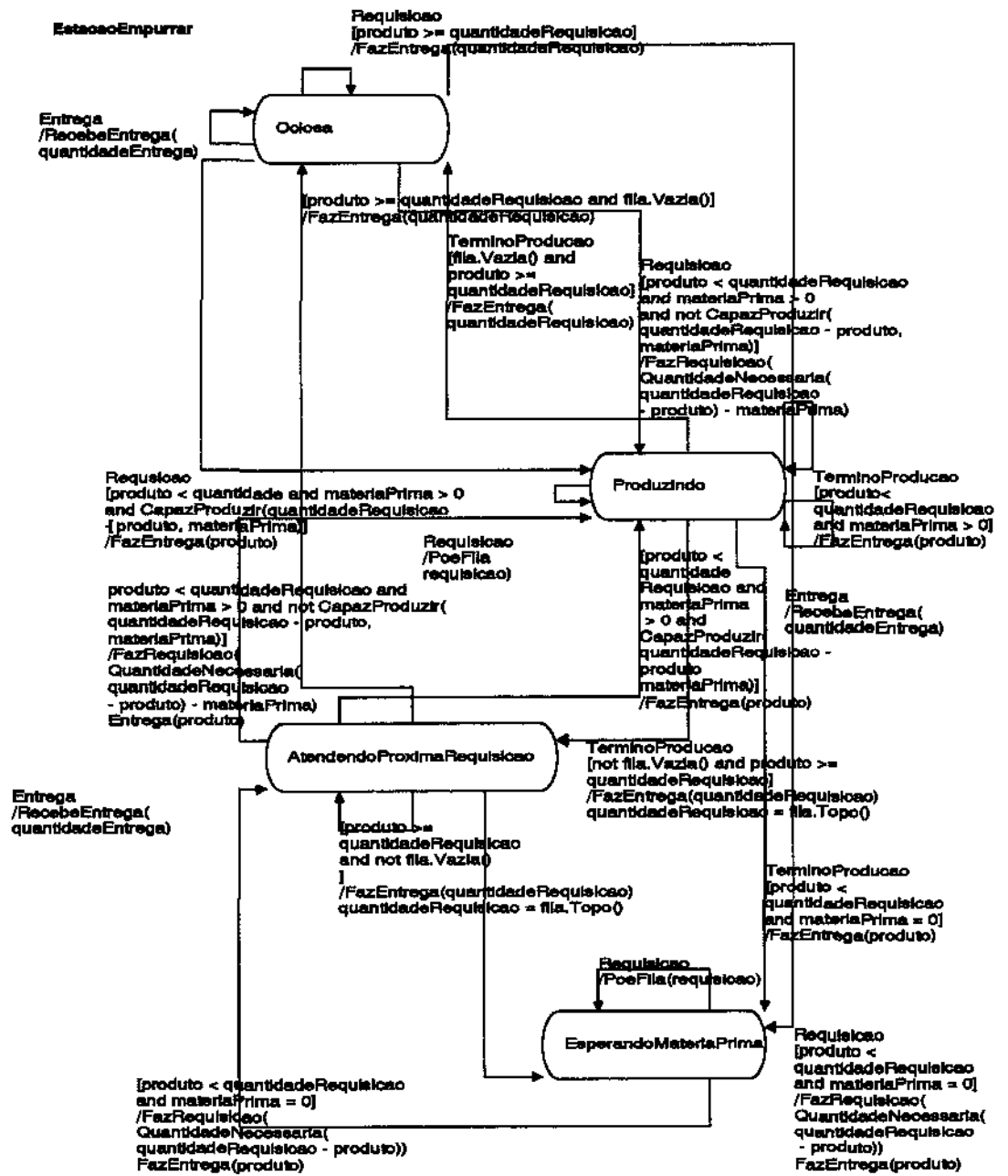


Figura B.13: Modelo dinâmico do objeto Estação Empurrar.

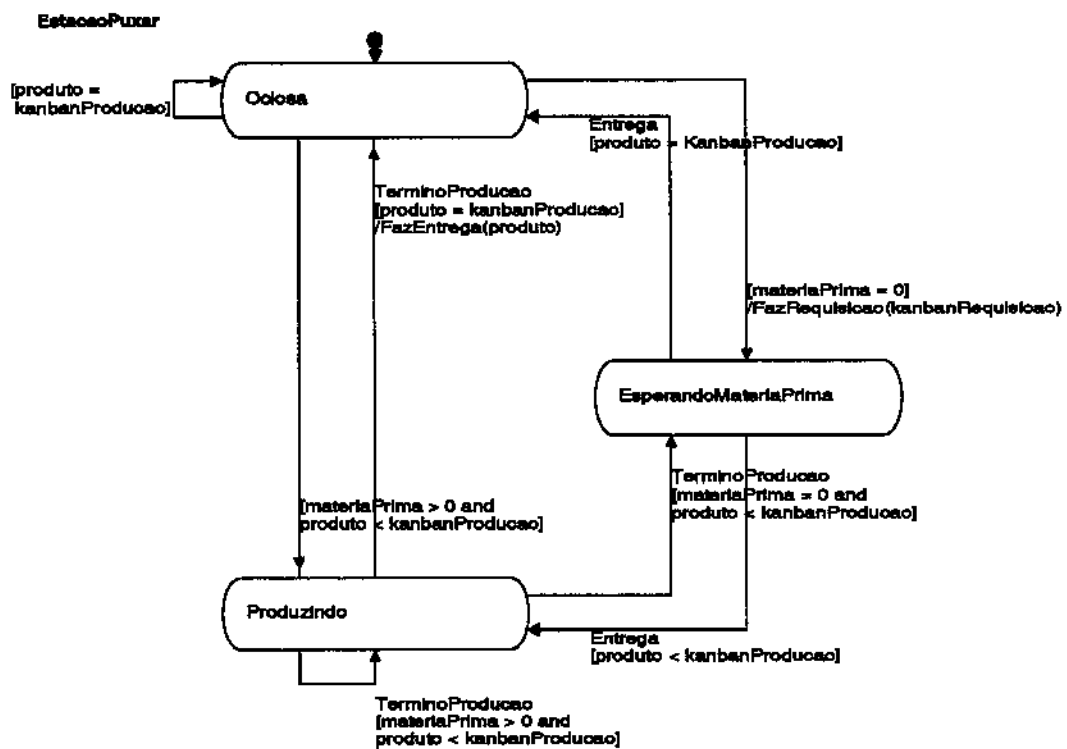


Figura B.14: Modelo dinâmico do objeto Estação Puxar.

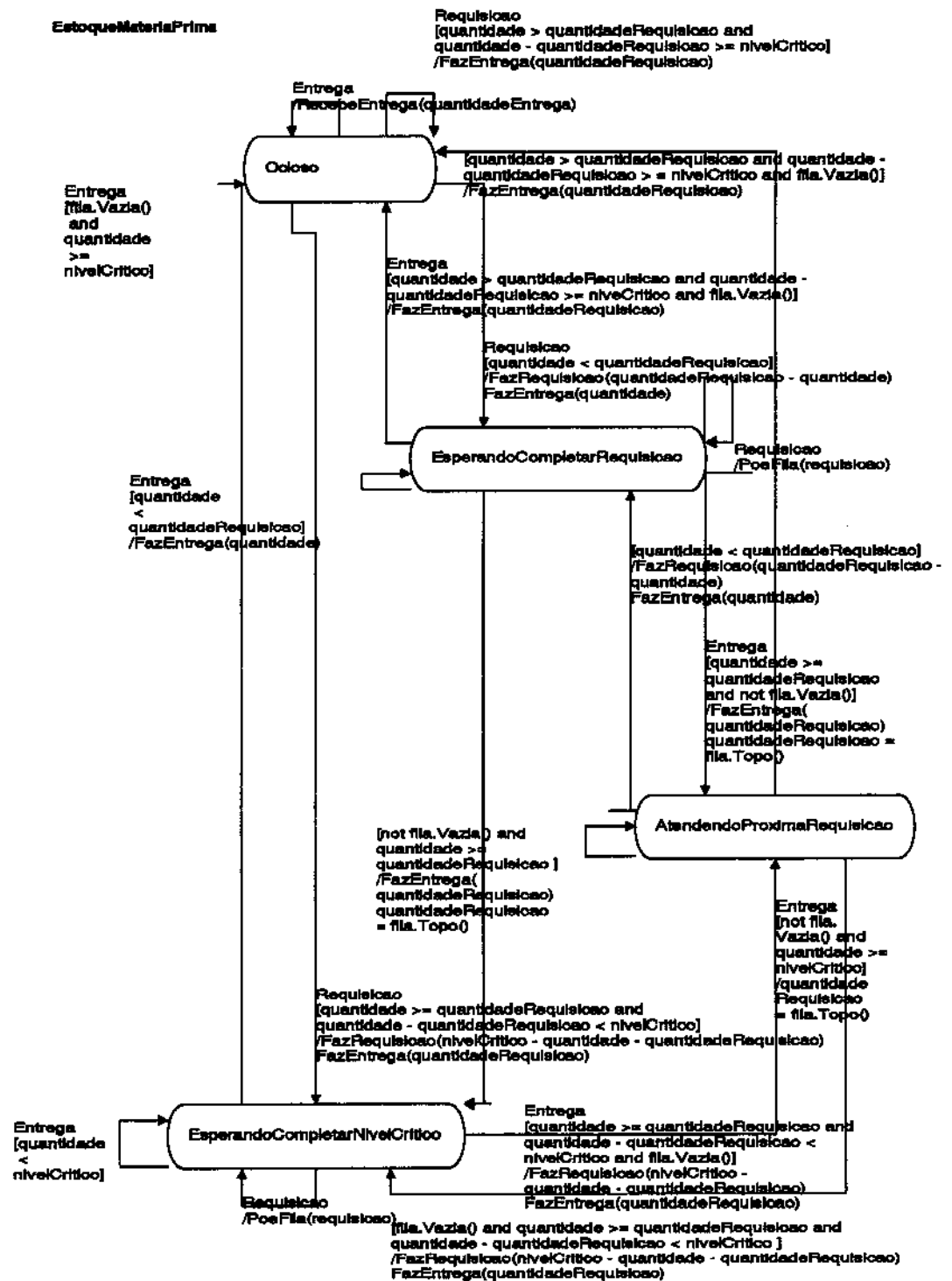
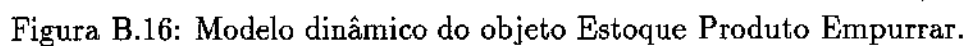


Figura B.15: Modelo dinâmico do objeto Estoque de Matéria Prima.



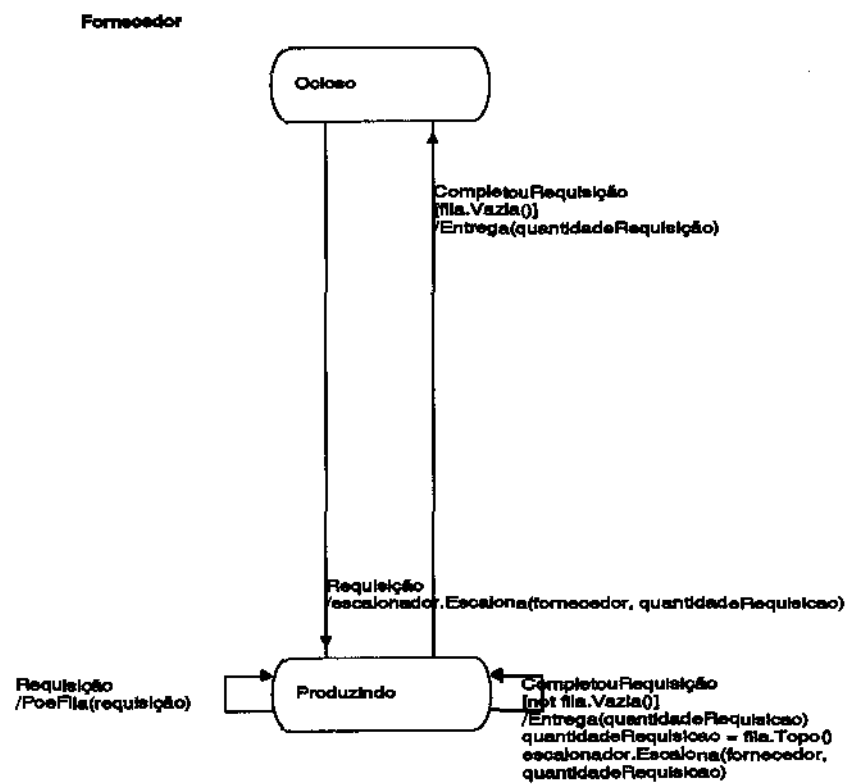


Figura B.18: Modelo dinâmico do objeto Fornecedor.

Bibliografia

- [Asymetrix, 1992] Asymetrix Corporation. *Toolbook Ideas: An authors introduction programming in Toolbook*. Washington, USA.
- [Baranauskas et al, 1997] Baranauskas, Maria C. C., Borges, Marcos Augusto F., Borges, Edmar Lourenço. *Learning by creating models: a computer-based environment for industrial application, 1997*. Proceedings in Education, AI-ED97. Edited by B. du Boulay and R. Mizoguchi. IOS Press, Tokyo Japan. Pág. 426-433.
- [Baranauskas, Oliveira, 1994] Baranauskas, Maria C. C., Oliveira, Osvaldo L. *Estratégia para design de ambientes computacionais para modelagem, 1994*. III SBIE Simpósio Brasileiro de Informática na Educação; Pág. 25-37, Porto Alegre-RS.
- [Baranauskas, Oliveira, 1995] Baranauskas, Maria C. C., Oliveira, Osvaldo L. *Domain oriented modelling: a balance between simulation an programming, 1995*. Proceedings of the sixth IPFIP World Conference on Computers in Education, 1995. Edited by J. David Tinsley an Tom J. Von Weert. Published by Chapman and Hall, Paris France. Pág. 118-126.
- [Borges, 1997] Borges, Marcos Augusto F. *O design centrado no aprendiz no sistema Jonas: Uma experiência de desenvolvimento de um sistema para formação na empresa, 1997*. Tese Msc. Universidade Estadual de Campinas, Instituto de Computação.
- [Borges et al, 1995] Borges, Edmar Lourenço, Borges, Marcos Augusto F., Baranauskas, Maria C. C. *Da simulação à criação de modelos - um contexto para aprendizagem na empresa, 1995*. VI SBIE - Simpósio Brasileiro de informática na educação, Pág. 154-165. Florianópolis SC.
- [Brown, 1998] Brown, Randy. *Calendar Queues: A fast $O(1)$ priority queue implementation for the simulation event set problem, 1988*. Communications of the ACM, October 1998, Volume 31, Number 10, Pág. 1220-1226.

- [Breugnot et al, 1991] Breugnout, D. Gourgand, M. Hill, D. Kellert P. *Game: An object-oriented approach to computer animation in flexible manufacturing system modelling, 1991*. Record of Proceedings. Edited by Alan H. Rutan. The 24th annual Simulation symposium, New Orleans, Louisiana, april 1991.
- [De Jong, 1991] De Jong, T. *Learning and Instrucion with computer simulation, 1991*. Education e computing, 6, pág. 217-229.
- [Deming, 1986] W. Edwards Deming. *Out of the Crisis, 1986*. Cambridge: Massachusetts Institute of Techonology, Center for Advanced Engineering Study.
- [Exame, 1995] Exame, Informática, *O software reinventa a fábrica, 1995*. Revista Informática Exame, setembro 1995.
- [Folha de S. Paulo, Sebrae, 1994] Folha de S. Paulo, Sebrae. *Qualidade total: o que todos procuram, 1994*. Fascículo 1, março 1994.
- [Giron, 1991] Giron-Sierra, Jose M., Gomez-Pulido, Juan A. *Doing object oriented simulations: Advantages, new development tools, 1991*. Record of Proceedings. Edited by Alan H. Rutan. The 24th annual Simulation symposium, New Orleans, Louisiana, april 1991.
- [Goldratt, 1989] Goldratt, Eliyahu M. e Fox, Robert E. *A Meta, 1989*. Instituto de Movimentação e Armazenagem de Materiais, IMAM, São Paulo.
- [Graszy, 1994] Graszy, Barbara. *The Americam Association for Artificial Intelligence, 1994*. A report to ARPA on twenty first centry intelligent system. AI Magazine.
- [Hebenstreit, 1991] Hebenstreit, J. *Simulation as on Education tool, 1991*. Proceedings of international conference on technology and education, Toronto, Canadá.
- [High Performance Systems, 1993] High Performance Systems. *IThink Demo Package, 1994*. High Performance Systems Inc.
- [Imagine That Inc., 1993] Imagine That Inc. *Demonstration Version Extend 2.0, 1993*. Imagine That Inc.
- [Laurel, 1990] Laurel, B. K. *The art of human-computer interface design, 1990* Massachusetts, Addison Wesley, 1990. Pág. 91-93. Users and Context Introduction.
- [Mazzone, 1993] Mazzone, Juares Salvatore. *O sistema "enzuto" e a educação no Brasil, 1993*. Computadores e conhecimento: repensando a educação organizador por José Armando Valente, Campinas-SP Gráfica Central da Unicamp, pág. 274-312.

- [Mazzone, 1995] Mazzone, Juarez Salvatore. *Educação na sociedade enxuta, 1995*. NIED-MEMO n.33, 2012. Núcleo de informática Aplicada à Educação, Universidade Estadual de Campinas.
- [Moura, 1989] Moura, Reinaldo. *Kanban, a simplicidade do controle da produção, 1994*. Instituto de Movimentação e Armazenagem de Materiais, IMAM, São Paulo.
- [Neelamkavil, 1995] Neelamkavil, Francis. *Computer Simulation and modelling, 1988*. John Wiley e Sons Ltd., 1988. ISN 0 471 911291.
- [Nied, 1995] *Projeto de dinamização da formação e aprendizagem nas empresas, 1995*. Documento interno do Núcleo de Informática Aplicada a Educação, Universidade Estadual de Campinas, 1995.
- [Nielsen, 1993A] Nielsen J. *Usability evaluation and inspection methods. In bridges between worlds, 1993*. INTERCHI'93. Tutorial Notes 22. Reading, MA: Addison Wesley. 1993.
- [Nielsen, 1993B] Nielsen J. *Usability Engineering, 1993*. Boston: Academic Press, 1993.
- [Oliveira, 1995] Oliveira, Osvaldo Luiz. *Design de ambientes computacionais para modelagem e um contexto educacional, 1995*. Tese Msc. Universidade Estadual Campinas, DCC/IMECC.
- [Pagano, 1992] Pagano, R. *Computer Simulation as an education tool, 1992*. PhD. Thesis. Université Catholique de Louvain, Belgium.
- [Papert, 1980] Papert. *Mind-Storms children, computers and powerful ideas, 1980*. Basic Book Inc.
- [Papert, 1986] Papert. *Construcionism: A new opportunity for elementary science education, 1986*. A proposal to the national science foundation, Massachusetts Institute of Technology. Media Laboratory, Epistemology and Learning Group, Cambridge, MA.
- [Pollacia, 1989] Pollacia, Lissa F. *A survey of discrete event simulation and state-of-the-art discrete event languages, 1989*. Simulation Digest, Volume 20, Número 3, Pág. 8-25, fall 1989.
- [Reinhardt, 1995] Reinhardt, Andy. *As novas formas de aprender, 1995*. Byte março, 1995.

- [Rumbaugh et al, 1991] Rumbaugh, J., Blaha, M., Premerelani, W., Eddy, F., Lorenzen, W. *Object-oriented modeling and design, 1991*. New Jersey Prentice Hall, Englewood Cliffs, 1991.
- [Shubik, 1960] Shubik, M. *Simulation of the industry and the firm, 1960*. American Economic Review, L, n.5, 908-919.
- [Valente, 1993] Valente, J. A. *Porque o computador na educação, 1993*. Computadores e conhecimento: repensando a educação organizado por José Armando Valente, Campinas-SP Gráfica Central da Unicamp, pág. 24-44.
- [Widman et al, 1989] Widman, Lawrence E. Laporo, Kenneth A. Nielsen, Norman R. *Artificial Intelligence, simulation and modelling, 1989*. A Willy Interscience Publication 1989.
- [Womack, 1992] Womack, J. P., Jones, D. T., Roos, D., Ferro, J. R. *A máquina que mudou o mundo, 1992*. Editora Campus, tradução do inglês.
- [Zuk, 1990] Zuk, John S. *Effective cost modeling on the factory floor: Taking simulation to botton line, 1990*. Winter Simulation Conference Proceedings. Edited by Osman Balci, Randall P. Sadowski e Richard E. Nance. December 1990, New Orleans, Louisiana, USA. Pág. 590-594.