

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE ENGENHARIA MECÂNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

DESENVOLVIMENTO DE PROTÓTIPO DE SISTEMA ESPECIALISTA PARA
PROJETO PNEUMÁTICO

DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL DE SANTA CATARINA
PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA MECÂNICA

JOEL BRASIL BORGES

FLORIANÓPOLIS, MARÇO DE 2002

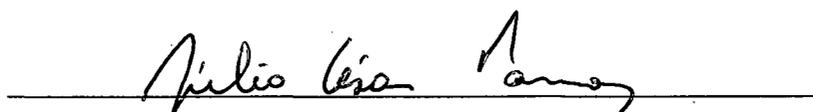
DESENVOLVIMENTO DE PROTÓTIPO DE SISTEMA ESPECIALISTA PARA
PROJETO PNEUMÁTICO

JOEL BRASIL BORGES

ESTA DISSERTAÇÃO FOI JULGADA PARA OBTENÇÃO DO TÍTULO DE MESTRE
EM ENGENHARIA MECÂNICA

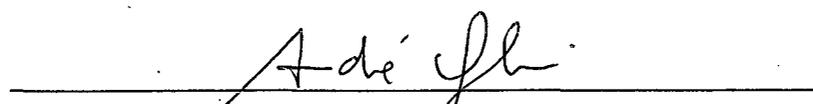
ESPECIALIDADE ENGENHARIA MECÂNICA E APROVADA EM SUA FORMA
FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

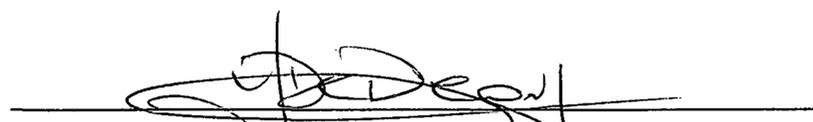

Prof. Jonny Carlos da Silva, Dr. Eng. – Orientador


Prof. Júlio César Passos, PhD. – Coordenador da Pós-Graduação

BANCA EXAMINADORA


Prof. Edison da Rosa, Dr. Eng.


Prof. André Oglíari, Dr. Eng.


Prof. Victor Juliano de Negri, Dr. Eng.

A lei da vida é
mudar.

Simone de Beauvoir

Aos esforços dos meus pais Joel e Sirley,
a minha irmã e sobrinho Luciane e João,
e à Suzy pelo companheirismo.

AGRADECIMENTOS

Ao LASHIP e aos seus membros pela possibilidade de desenvolvimento deste trabalho;

Ao prof. Jonny pela orientação e participação durante todas as fases deste trabalho;

Ao prof. Victor pela colaboração prestada no teste do protótipo desenvolvido;

Aos membros da banca examinadora desta dissertação pela colaboração no aprimoramento do conteúdo redigido;

Aos colegas, Guilherme e Fernando, pela amizade e pelos bons momentos. Também aos colegas Antônio, Edivaldo, Fred e Rodolfo;

Ao POSMEC pela possibilidade de integração ao seu programa de pós-graduação;

À SATC pelo apoio laboratorial, com as bancadas didáticas em hidráulica e pneumáticas e o *software Automation Studio*. Aos alunos de 2001/2 da disciplina de Projetos hidráulicos e pneumáticos, dos cursos de Eletromecânica e Automação Industrial pelo teste do protótipo;

À CAPES pelo apoio financeiro nas etapas iniciais deste trabalho.

SUMÁRIO

Capítulo 1 – Introdução	1
1.1 – A pneumática no cotidiano	1
1.2 – Perfil dos projetistas de sistemas pneumáticos	2
1.3 – Demanda por conhecimento em projetos de sistemas pneumáticos	3
1.4 – Objetivo do trabalho	4
1.5 – Objetivos finais esperados	4
1.6 – Tópicos do trabalho	5
Capítulo 2 – Metodologia de projeto	7
2.1 – Origem dos projetos	7
2.2 – Etapas das metodologias de projetos	10
2.3 – Projeto de sistemas	12
2.4 – Engenharia simultânea	13
2.5 – Computadores no auxílio ao projeto	19
Capítulo 3 – Sistemas especialistas	21
3.1 – Abordagens de IA e SE	21
3.1.1 – Origem da abordagem de SE	23
3.1.2 – Diferenças entre as abordagens algorítmicas e heurísticas	25
3.1.3 – Aplicação da técnica de SE	26
3.1.4 – Campos de atuação dos SE	29
3.1.5 – Estrutura computacional dos SE	31
3.1.6 – Sistemas de produção ou <i>regras</i>	32
3.1.7 – Tipos de encadeamentos entre <i>regras</i>	33
3.1.8 – Busca por solução	35
3.1.9 – Sistemas <i>shell</i>	38
3.2 – Metodologia de projeto de SE	40
3.2.1 – Aquisição do conhecimento	44
3.2.2 – Representação do conhecimento	46
3.2.3 – Implementação do conhecimento	49
3.2.4 – Teste do SE	49
Capítulo 4 – Projeto de sistemas pneumáticos	50
4.1 – Aplicações da pneumática	50
4.2 – Classificação dos circuitos de <i>comando</i> pneumático	51
4.3 – Estrutura de sistemas pneumáticos	53
4.4 – Elementos de sistemas pneumáticos	54

4.5 – Requisitos de projeto em pneumática	56
4.5.1 – Modelagem de cargas mecânicas	57
4.5.2 – Requisitos qualitativos complementares	60
4.5.3 – Requisitos para a interface de projetos pneumáticos sequenciais	61
4.6 – Abordagens aplicadas aos projetos de sistemas pneumáticos	63
4.6.1 – Abordagens <i>adaptativas</i> na construção de esquemas pneumáticos	64
4.6.2 – Abordagens <i>originais</i> na construção de esquemas pneumáticos	64
4.6.3 – Abordagem de <i>variante</i> na construção de esquemas pneumáticos	65
4.6.4 – Princípios de solução para projetos de sistemas pneumáticos	65
4.6.5 – Combinação de princípios de solução	73
4.7 – Ferramentas computacionais nos projetos pneumáticos	74
Capítulo 5 – Etapas de desenvolvimento do protótipo	76
5.1 – Metodologia de projeto de sistema especialista adotada no protótipo	76
5.1.1 – Análise da viabilidade dos SE no projeto de sistemas pneumáticos	77
5.1.2 – Especificação do protótipo	78
5.1.3 – Projeto preliminar	81
5.1.3.1 – Técnica de representação do conhecimento	81
5.1.3.2 – Método de inferência	83
5.1.3.3 – Ferramentas	84
5.1.3.4 – Time de desenvolvimento	86
5.1.4 – Prototipagem inicial	87
5.1.5 – Projeto detalhado	87
5.2 – Estrutura funcional do protótipo	88
5.3 – Domínio de conhecimento do protótipo	90
5.3.1 – Circuitos de gerenciamento	92
5.3.2 – Modularidade nos circuitos	93
5.3.3 – Nomenclatura dos componentes	94
5.4 – Estrutura de classes	97
Capítulo 6 – Teste do protótipo	102
6.1 – Importância da etapa de teste em SE	102
6.2 – Diferenças no teste de programas computacionais convencionais e SE	103
6.3 – Estágio do teste de SE	104
6.3.1 – Verificação em SE	104
6.3.1.1 – Erros de sintaxe	105
6.3.1.2 – Erros de semântica	106
6.3.2 – Validação em SE	106
6.4 – VV do protótipo	107
6.4.1 – Verificação do protótipo com <i>teste de caso</i>	107
6.4.1.1 – Definição dos dados do <i>teste de caso</i>	108
6.4.1.2 – Montagem do <i>teste de caso</i> no <i>Automation Studio</i>	111
6.4.1.3 – Montagem dos esquemas em bancada experimental	113
6.4.2 – Validação do protótipo	114
6.4.2.1 – Usuários não especialistas	117

6.4.2.2 – Usuários especialistas _____	119
Capítulo 7 – Conclusões e recomendações futuras _____	123
7.1 – Conclusões sobre o trabalho _____	123
7.2 – Recomendações futuras _____	126
Referências bibliográficas _____	129
Bibliografia _____	133
Apêndice A – Telas do protótipo no ambiente CLIPS padrão _____	134
Apêndice B – Telas do protótipo no ambiente orientado a eventos _____	136
Apêndice C – Considerações computacionais sobre o protótipo _____	140
Apêndice D – Comparação entre as versões do protótipo _____	142

LISTA DE TABELAS

Tabela 2.1 – Diferenças na condução do projeto (BULLINGER; WARSCHAT, 1996).	16
Tabela 2.2 – Custo das etapas de projeto (MILLER, 1993).	17
Tabela 2.3 – Custo típico decorrente de mudanças (MILLER, 1993).	18
Tabela 3.1 – Definições de IA.	22
Tabela 3.2 – Diferenças entre programas computacionais convencionais e SE (DYM; LEVITT, 1991, DURKIN, 1994, WATERMAN, 1986).	26
Tabela 3.3 – Comparação entre EH e SE (DURKIN, 1994, WATERMAN, 1986).	27
Tabela 3.4 – Critérios para a seleção dos SE (WATERMAN, 1986).	28
Tabela 3.5 – Características das técnicas de busca (DURKIN, 1994).	37
Tabela 4.1 – Componentes de sistemas pneumáticos (BOLLMANN, 1997).	56
Tabela 4.2 – Circuitos de <i>simples comando</i> utilizados segundo critérios de <i>adaptação</i> .	66
Tabela 4.3 – Principais métodos de projeto pneumáticos (BOLLMANN, 1997).	66
Tabela 4.4 – Limitações dos princípios de solução (BOLLMANN, 1997).	74
Tabela 5.1 – Viabilidade da técnica de SE em projetos pneumáticos.	77
Tabela 5.2 – Opções de sistemas <i>shell</i> para o protótipo (DURKIN, 1994).	85
Tabela 5.3 – Dedicção sobre os módulos de SE (DYM; LEVITT, 1991).	88
Tabela 5.4 – Designação do tipo de componente.	94
Tabela 6.1 – Validação do protótipo com interface orientada a eventos por estudantes.	118
Tabela A.1 – Validação da interface de entrada por não especialistas.	142
Tabela A.2 – Validação da interface de saída por não especialistas.	143
Tabela A.3 – Validação da BC por não especialistas.	145

LISTA DE FIGURAS

Figura 2.1 – Etapas dos projetos de produto (FRENCH, 1985).	11
Figura 2.2 – Comparação do tempo de lançamento de produtos (HARTLEY, 1998).	15
Figura 2.3 – Período de lançamento dos produtos (CHAPMAN et al, 1992).	16
Figura 2.4 – Queda de competitividade devido às mudanças (HARTLEY, 1998).	17
Figura 3.1 – Interações na construção dos SE (WATERMAN, 1986).	23
Figura 3.2 – Abordagens algorítmicas versus heurísticas (WATERMAN, 1986).	25
Figura 3.3 – Número de SE desenvolvidos por ano (DURKIN, 1994).	29
Figura 3.4 – Número de SE desenvolvidos por tipo de aplicação (DURKIN, 1994).	30
Figura 3.5 – Funções exercidas pelos SE (DURKIN, 1994).	30
Figura 3.6 – Elementos computacionais dos SE (DYM;LEVITT, 1991).	32
Figura 3.7 – Estrutura padrão das regras (BUCHANAN; SHORTLIFFE, 1984).	32
Figura 3.8 – Classificação dos tipos de DC (DYM; LEVITT, 1991).	33
Figura 3.9 – Percursos dos encadeamentos <i>direto e reverso</i> (CHORAFAS, 1990).	34
Figura 3.10 – Exemplo de <i>rede de inferência de regras</i> (DURKIN, 1994).	36
Figura 3.11 – Busca em profundidade.	36
Figura 3.12 – Busca em amplitude.	36
Figura 3.13 – Categorias de <i>software</i> para SE (DURKIN, 1994).	39
Figura 3.14 – Etapas de elaboração de <i>software</i> (GONZALEZ; DANKEL, 1993).	41
Figura 3.15 – <i>Todo-parte e generalização-especialização</i> (COAD; YOURDON, 1997).	48
Figura 4.1 – Esquema de aplicações de comando (BOLLMANN, 1997).	51
Figura 4.2 – Principais tipos de comandos binários (BOLLMANN, 1997).	52
Figura 4.3 – Estrutura orientada a objeto para sistemas hidráulicos (SILVA, 1998).	54
Figura 4.4 – Elementos de sistemas pneumáticos (HASEBRINK; KLOBER, 1988).	55
Figura 4.5 – Elementos de sistema de projeto seqüenciais (FESTO DIDATIC, 1992).	61
Figura 4.6 – Diagrama das memórias com intertravamento.	68
Figura 4.7 – Diagrama expandido das memórias com intertravamento.	69
Figura 4.8 – Memória pneumática pura utilizando componentes com memorização.	70
Figura 4.9 – Memória puramente pneumática utilizando componentes sem memorização.	71
Figura 4.10 – Memória elétrica utilizando relé simples.	72
Figura 5.1 – Plataformas computacionais utilizadas na construção de SE (DURKIN, 1994).	85
Figura 5.2 – Programas computacionais utilizados na construção de SE (DURKIN, 1994).	85
Figura 5.3 – Estrutura global do protótipo.	89

Figura 5.4 – Campos do código dos componentes. _____	94
Figura 5.5 – Exemplo de esquema de circuito fornecido pelo protótipo. _____	96
Figura 5.6 – Detalhes sobre componentes dos circuitos. _____	96
Figura 5.7 – Estrutura refinada das classes do protótipo. _____	98
Figura 6.1 – <i>Diagrama trajeto-passo</i> do teste de caso. _____	108
Figura 6.2 – Esquema do método passo a passo para o teste de caso. _____	109
Figura 6.3 - Esquema do método seqüência máxima para o teste de caso. _____	110
Figura 6.4 – <i>Teste de caso</i> do método passo-a-passo montado no AS. _____	111
Figura 6.5 – <i>Teste de caso</i> do método seqüência máxima montado no AS. _____	112
Figura 6.6 – <i>Teste de caso</i> do método passo-a-passo montado na bancada didática. _____	114
Figura 6.7 – <i>Ficha de avaliação</i> do protótipo no formato impresso. _____	115
Figura 6.8 – <i>Ficha de avaliação</i> inserida no protótipo. _____	116
Figura A.1 – Tela do protótipo na interface padrão do CLIPS. _____	134
Figura A.2 – Tela do protótipo na interface padrão do CLIPS. _____	134
Figura A.3 – Tela de entrada do protótipo. _____	136
Figura A.4 – Seleção do sistema operacional. _____	136
Figura A.5 – Menu de opções do protótipo. _____	137
Figura A.6 – Seleção do tipo de projeto que deseja-se realizar. _____	137
Figura A.7 – Seleção do sistema operacional. _____	138
Figura A.8 – <i>Browser</i> do protótipo. _____	139
Figura A.9 – Tela de entrada de dados do projetista. _____	139
Figura A.10 – Tela de entrada de dados qualitativos sobre o projeto. _____	139

LISTA DE EQUAÇÕES

Equação 3.1 – Conhecimento dos especialistas (DURKIN, 1994).	22
Equação 5.1 – Somatório para a descrição da <i>função critério</i> .	101

LISTA DE SÍMBOLOS

- AS – Automation Studio
- BC – Base de conhecimento
- CLP – Controlador lógico programável
- DC – Domínio do conhecimento
- EC – Engenheiro do conhecimento
- EH – Especialista humano
- EP – Eletropneumático
- ES – Engenharia simultânea
- H&P – Hidráulica e pneumática
- IA – Inteligência artificial
- LASHIP – Laboratório de Sistemas Hidráulicos e Pneumáticos
- NA – Normalmente aberto
- NF – Normalmente fechado
- OO – Orientação a objetos
- PC (*personal computer*) – Computador pessoal
- PDM – Projeto de dissertação de mestrado
- POSMEC – Programa de Pós-Graduação em Engenharia Mecânica / UFSC
- PN – Pneurônico
- PP – Pneumático puro
- SATC – Sociedade de Assistência aos Trabalhadores do Carvão / Criciúma – SC
- SE – Sistemas especialistas
- UFSC – Universidade Federal de Santa Catarina
- VB – Visual Basic
- VD – Válvula direcional
- VV – Verificação e validação

RESUMO

Com o contínuo incremento dos processos industriais e a crescente demanda por formas alternativas de energia que satisfaçam critérios e necessidades específicas, a pneumática apresenta-se como uma fonte de energia para a obtenção de trabalho mecânico, concorrendo diretamente com os princípios de solução que utilizam o domínio hidráulico e eletromecânico, sendo aplicada nos mais diversos contextos de automação de processos industriais, principalmente em situações que requeiram procedimentos seqüenciais.

Neste contexto, este trabalho propõe-se a desenvolver um protótipo computacional fundamentado nos conceitos da técnica de sistemas especialistas, sendo este formato adequado em condições em que prevaleça o conhecimento heurístico na tomada de decisão. Nestas condições os projetos de sistemas pneumáticos podem beneficiar-se do conhecimento especializado para atingir o nível de qualidade esperado, com a detecção ainda nas fases iniciais das possíveis falhas, seguindo os conceitos da engenharia simultânea.

Na construção desta ferramenta computacional, são estudados conceitos sobre o processo de evolução do mesmo, através da prévia análise de viabilidade e posterior desenvolvimento através de processos incrementais sobre os ciclos de aquisição do conhecimento, representação do conhecimento, implementação do conhecimento e teste.

Dentre as áreas de aplicação da pneumática, neste sistema especialista a ser construído busca-se atingir predominantemente o domínio de conhecimento utilizado em projetos de sistemas pneumáticos destinados à operarem em condições seqüenciais de movimentação de cargas mecânicas, segundo metodologias adotadas na pneumática e eletropneumática como o passo-a-passo e seqüência máxima.

ABSTRACT

Due to the continuous application of industrial processes and increasing demand for alternative forms of energy that satisfy criteria and specific needs, pneumatics presents itself as source of energy to obtain mechanical power, competing with other forms such as hydraulic and electro-mechanics, in some power ranges, applied to different automation processes, mainly in those requiring sequential procedures.

In this context, this work aims to develop a computational system prototype based on the concept of expert systems, more applicable in conditions where a great deal of knowledge is of heuristic nature to support the decision making process. With this condition the design of pneumatic systems benefits from expertise to achieve the expected level of quality, detecting since the early stages some possible failures, according to the concepts of concurrent engineering.

In the development of this computational tool, concepts related to its expansion are studied, through the feasibility analysis and incremental model application via continuous cycles of knowledge acquisition, representation, implementation and test.

Among the areas this prototype aims to cover is the domain used in the design of pneumatic systems to operate in sequential steps, using the methodologies of pneumatics and electro-pneumatics, previously known as step-by-step and maximum sequence methods.

Capítulo 1 – Introdução

1.1 – A pneumática no cotidiano

A utilização industrial da pneumática como fonte de trabalho mecânico, nos últimos anos vem conquistando espaço. A pneumática destaca-se em aplicações que demandam elevada performance, em ambientes que exijam condições de higiene controlada, para atividades com grande repetitividade e velocidade (SCHNEIDER; HITCHCOX, 1998).

Haja vista estas vantagens, muitas são as aplicações de automação industrial em que se prioriza a pneumática em detrimento de outros princípios de solução, como a mecânica, hidráulica e elétrica. Mesmo assim, a pneumática demonstra-se suficientemente flexível para operar em conjunto com estes princípios de solução, através da disponibilização de inúmeros componentes, que oferecem alta modularidade na construção de circuitos, viabilizando soluções personalizadas para cada demanda, com o oferecimento de circuitos hidropneumáticos, eletropneumáticos e, mais recentemente, a pneumática que agrupa a pneumática e a eletrônica dos *controladores lógicos programáveis* (CLP), com destaque para as redes de comunicação, como o padrão ASi¹.

Isto propicia a redução no tempo de desenvolvimento, além de construções mais compactas, resultando em custos menores à instalação quando concluída (SCHNEIDER; HITCHCOX, 1998).

Como todo princípio de solução, a pneumática também apresenta algumas desvantagens, dentre as quais cita-se (ABHP, 1995):

- Para o bom funcionamento, os componentes pneumáticos requerem níveis de impureza e umidade do ar estritamente controlados;
- A força e torque possíveis de se obter são significativamente menores que os obtidos na hidráulica;
- Os escapes de ar existentes para a atmosfera são uma grande fonte geradora de ruído;

¹ ASi – Actuators sensors interface (ASI, 2001).

- Em situações de rápida expansão do ar, pode haver congelamento dos componentes.

Apesar destas desvantagens, a pneumática ainda apresenta destaque. É na exploração de seus benefícios, que os projetistas optam por selecioná-la. Dentre os setores que aplicam a pneumática de forma intensiva para automação, tem destaque (ABHP, 1995):

- Setor de embalagem de produtos;
- Indústria de processamento de alimentos;
- Setor agrícola e agropecuário;
- Exploração mineral;
- Indústria química e petroquímica;
- Indústria de polímeros, com destaque para a termoformagem por sopro, onde o ar comprimido exerce a função de ferramenta de trabalho;
- Indústria metalúrgica;
- Setores de móveis e madeira, papel, têxtil, couro, construção civil e transporte.

Com este amplo campo de aplicação da pneumática, toda pesquisa sobre ela é de grande valia, por ser mais uma contribuição para o aprimoramento nas suas aplicações.

1.2 – Perfil dos projetistas de sistemas pneumáticos

No ciclo de desenvolvimento de um produto, o qual pode ser um sistema pneumático, inevitavelmente tem-se a passagem pela etapa inicial de projeto, por mais simples que este seja. A etapa de projeto requer o empenho do projetista, que deve ser um profissional capacitado para exercer esta função.

Em estudo feito por Heney (1998) para a revista *Hydraulics & Pneumatics*, vários consultores opinaram sobre temas diversos, dentre eles a *desqualificação dos profissionais* que atuam em *hidráulica e pneumática* (H&P). Dentre os motivos levantados, cita-se os seguintes fatores:

- Deficiência na formação destes profissionais por parte das instituições de ensino;
- Existência de uma prioridade por parte das instituições de ensino em oferecer conteúdos sobre mecânica e elétrica em detrimento dos temas correlacionados à H&P.

Isto é observado nos dados levantados por Bud Trinkel, um dos consultores questionados por Heney (1998). Ele informa que 90% dos projetos de H&P nos Estados

Unidos são desenvolvidos por: distribuidores de componentes de H&P, engenheiros dos fabricantes de componentes de H&P ou consultores em H&P. Pela relevância dos Estados Unidos no mercado de H&P, é esperado um cenário similar no restante dos países.

Segundo Trinkel (HENEY, 1998), isto deve-se ao desconhecimento por parte dos projetistas sobre H&P, que preferem adotar soluções em domínios que possuam maior afinidade, segundo a sua formação acadêmica de origem como mecânica ou elétrica, apesar de muitas vezes serem estas soluções tecnologicamente deficientes para a aplicação a que se destinam.

A mudança deste cenário, segundo consultores questionados por Heney (1998), passa pela adoção de medidas que priorizem a formação de profissionais em H&P.

1.3 – Demanda por conhecimento em projetos de sistemas pneumáticos

O cenário traçado quanto à habilitação e qualificação dos profissionais técnicos em H&P demonstra uma lacuna existente entre a demanda por conhecimento sobre H&P e a oferta por profissionais treinados e devidamente capacitados.

A deficiência identificada na obtenção destes profissionais qualificados em H&P devido à falta de conhecimento destes em pneumática, indica a existência de uma demanda reprimida por fontes que possam ser usadas como *base de conhecimento* (BC) em pneumática.

Para completar esta lacuna existente nos projetos, as abordagens que utilizam a técnica de *sistemas especialistas* (SE), uma das áreas de aplicação da *inteligência artificial* (IA), demonstra a sua eficiência através de sistemas de busca de informações específicas sobre o projeto, auxiliando na análise, otimização e combinação de soluções. Com a abordagem de SE, a demanda por *especialistas humanos* (EH) pode ser parcialmente suprida pelo conhecimento inserido na BC gerada. Assim, o projetista pode dedicar-se a outras tarefas, como a coleta de necessidades dos consumidores, deixando as tarefas exaustivas e repetitivas para serem feitas pelo computador (PAHL; BEITZ, 1995).

Aplicações diversas têm utilizado os benefícios da técnica de SE, como os setores de projeto e planejamento, que juntos compõem aproximadamente 20% do total de SE elaborados (DURKIN, 1994).

Em estudos anteriores realizados por Silva (1998), demonstra-se a adequação e viabilidade de construção de SE para projetos em hidráulica, sendo esperado resultado similar para aplicações em pneumática, devido à semelhança entre estes domínios tecnológicos.

1.4 – Objetivo do trabalho

Dentre as diversas áreas de atuação do *Laboratório de Sistemas Hidráulicos e Pneumáticos* (LASHIP) da *Universidade Federal de Santa Catarina* (UFSC), encontra-se em desenvolvimento a linha de pesquisa de SE aplicados em projetos H&P, que teve início com o trabalho desenvolvido por Silva (1998) em seu doutorado.

Dentro desta linha de pesquisa, o presente trabalho destina-se à construção de um protótipo computacional fundamentado na técnica de SE, capaz de realizar projetos pneumáticos durante a fase conceitual.

Este trabalho pretende envolver também as seguintes áreas de pesquisa:

- *Engenharia do conhecimento* em projetos de sistemas pneumáticos, que é o *domínio de conhecimento* (DC) a ser explorado;
- Abordagens metodológicas e de *engenharia simultânea* (ES) tanto sobre o desenvolvimento de programas computacionais em SE, quanto no processamento de informações em projetos de sistemas pneumáticos;
- Modelagem *orientada a objeto* (OO);
- Ergonomia de interfaces com o usuário;
- Teste de programas computacionais em SE.

1.5 – Objetivos finais esperados

Este protótipo computacional em SE a ser desenvolvido é destinado à construção de sistemas pneumáticos, em algumas aplicações específicas, como os projetos pneumáticos seqüenciais que possuem uma grande aplicação em processos de automação industrial, devido à grande amplitude de aplicações da pneumática.

Com a conclusão do SE protótipo, espera-se ao fim do trabalho oferecer uma BC capaz de:

- Perpetuar o conhecimento no domínio de projetos de sistemas pneumáticos;
- Ser uma base de consulta em pneumática, prontamente disponível;
- Ser uma BC auxiliar que possa ser utilizada como complemento as propostas de princípios de solução em pneumática, em conjunto com o trabalho dos projetistas.

Das etapas de desenvolvimento deste SE, pretende-se demonstrar os benefícios obtidos com:

- A modelagem OO de sistemas pneumáticos, quanto à modularidade e simplificação no desenvolvimento de BC;
- A abordagem de SE na etapa conceitual de projetos de sistemas pneumáticos;
- A abordagem metodológica de desenvolvimento de *software* em SE;

Das etapas de desenvolvimento de projetos de sistemas pneumáticos, pretende-se demonstrar a otimização obtida no trabalho realizado pelos projetistas, através dos seguintes pontos:

- Agilização da atividade de proposta de soluções alternativas durante a fase conceitual;
- Realização de combinações de princípios de soluções;
- Minimização do tempo despendido em etapas de construção de esquemas em ambiente computacional de auxílio ao projeto;
- Oferecer treinamento em pneumática, exercendo a função didático-pedagógica;

Dentre as características gerais esperadas deste SE protótipo, destacam-se:

- Facilidade na entrada de dados do projeto, de forma interativa, sem haver necessidade de um conhecimento prévio em pneumática por parte do usuário, bastando possuir a descrição das cargas mecânicas que deseja-se movimentar;
- Facilidade na obtenção dos dados de saída, também de forma interativa, com boa velocidade, através de diagramas de sistemas pneumáticos ou eletropneumáticos, conforme simbologia normatizada para estes componentes;
- Listagem de soluções preferenciais e soluções alternativas, todas consistentes segundo dados de entrada do projeto fornecido pelo usuário.

1.6 – Tópicos do trabalho

O restante do texto desta dissertação encontra-se dividido em um total de mais 5 capítulos, dos quais, citam-se os seguintes conteúdos por capítulo:

Capítulo 2 – Metodologia de projeto: abordagens metodológicas em projetos de modo geral, ferramentas de auxílio ao projeto, abordagem de ES e SE no auxílio ao projeto;

Capítulo 3 – SE: origem, diferenças com outros programas computacionais, metodologias de elaboração de SE, aquisição do conhecimento, representação do conhecimento com modelagem OO e *regras*;

Capítulo 4 – Projetos de sistemas pneumáticos: tipos de projetos pneumáticos, fontes de conhecimento em pneumática, etapas de um projeto pneumático, metodologias para projetos seqüenciais pneumáticos e eletropneumáticos, ferramentas computacionais no auxílio ao projeto;

Capítulo 5 – Descrição do protótipo: identificação de requisitos de projeto para *software*, fases de desenvolvimento do protótipo, interfaces de entrada e saída de dados do SE protótipo, estrutura OO do protótipo;

Capítulo 6 – Teste do protótipo: diferenças no teste de SE e programas computacionais convencionais, divisão da etapa de teste nas fases de *verificação e validação* (VV), VV do protótipo, resultados da VV do protótipo;

Capítulo 7 – Conclusões e recomendações para trabalhos futuros.

Capítulo 2 – Metodologia de projeto

Neste capítulo é abordada a aplicação de metodologias de projeto nas fases de desenvolvimento de produtos, sua importância e ferramentas computacionais de auxílio ao projeto. Por fim, é discutida a relevância da *engenharia simultânea* (ES) em projetos de origens diversas, como em sistemas pneumáticos e em *programas computacionais*.

2.1 – Origem dos projetos

O ponto de partida para as grandes soluções ou idéias passa pela concepção inicial dos projetos dos produtos. Pahl e Beitz (1995, p. 1, grifo nosso) caracterizam os projetos como sendo “[...] uma atividade criativa que solicita uma fundamentação sólida em matemática, física, química [...] assim como *conhecimento e experiência* em determinados domínios”.

A demanda por realizar um projeto tem origens diversas e pode estar fundamentada tanto nas reais necessidades do mercado, que tem a noção de suas carências, ou pode surgir de idéias inovadoras que os potenciais consumidores inicialmente não expressam por um determinado produto que, portanto, evidencia a possibilidade de exploração (PAHL; BEITZ, 1995). Nesta mesma linha de classificação das necessidades que dão origem aos projetos, Juran (apud BACK; FORCELLINI, 199-?) considera o seguinte arranjo:

- Necessidades manifestas;
- Necessidades reais;
- Necessidades latentes;
- Necessidades culturais;
- Necessidades atribuíveis a usos inesperados;
- Necessidades dos clientes relativas à satisfação do produto.

O projeto de qualquer produto, seja na forma concreta ou virtual, como o caso de *software*, tem início com a total identificação destas necessidades dos *clientes*. Para a perfeita condução das etapas de projeto, as necessidades dos *clientes* devem ser do conhecimento de todos os envolvidos com o projeto, sendo a função da equipe de projetistas oferecer a solução otimizada (CHAPMAN; BAHILL; WYMORE, 1992).

Como *clientes do projeto*, Back e Forcellini (199-?) classificam-os em três grupos: *externos* formados pelos *clientes* que são os usuários do produto final, *internos* que são os *clientes* que projetam e produzem o produto e os *intermediários* que fazem a conexão entre os *clientes internos* e os *externos* como os encarregados pelo armazenamento, transporte, vendas entre outros.

No ciclo de vida de *software*, alguns paralelos podem ser traçados com o ciclo de vida de produtos que constituem bens materiais, em que o conceito de *clientes do projeto* também pode ser aplicado, diversificando-se apenas na forma abstrata que o *software* possui e como é construído. Segundo Lucena (1987, p. 13-14) “[...] os custos se concentram no desenvolvimento e não na produção [...] sendo o modelo de ciclo de vida do software [...] importado da engenharia convencional, onde a noção de ciclo de vida de um produto é utilizada há muito tempo no planejamento de produtos”.

O fator humano responsável pela criação do projeto é personificado pelo projetista ou por um grupo de projetistas. O gerenciamento de todos os esforços dos participantes deve ser valorizado e devidamente orientado para obter o resultado final às custas do menor esforço, tempo e custo.

No percurso entre o início do projeto, com a identificação dos *clientes* existentes e as suas necessidades a serem atendidas, diversos caminhos podem ser percorridos e, é claro, diversas saídas ou soluções podem ser propostas.

O que define a saída a ser adotada é o conhecimento e experiência sobre o processo produtivo. Isto pode estar correlacionado à pressão do mercado por melhor desempenho, melhores preços e redução do *time-to-market*¹, planejamento do produto e vendas. Para tanto, é exigido conhecimento especializado de engenharia que, pode ter origem tanto na formação teórica do projetista, como na experiência anterior sobre outros produtos (PAHL; BEITZ, 1995).

Para auxiliar nos processos evolutivos de desenvolvimento de produtos, a formalização estruturada das etapas evolutivas dos projetos fundamenta o trabalho do projetista. Historicamente, a evolução dos projetos que seguem estruturas sistemáticas tem origem com Leonardo da Vinci, que utilizava variações dentre os princípios de soluções possíveis; após evoluções na forma de abordar os problemas de projeto, as etapas percorridas em um projeto tornaram-se metodológicas, o que não desvaloriza a intuição e a experiência dos projetistas talentosos e sim destaca a criatividade dos mesmos, racionalizando o projeto e os processos

¹ *Time-to-market* – intervalo de tempo entre o início do projeto e a disposição do produto no mercado.

produtivos (PAHL; BEITZ, 1995).

Parte desta experiência tem origem também na constatação de alguns erros de engenharia, sejam eles de projeto, fabricação, montagem ou utilização, cometidos em outras épocas ou até mesmo na atualidade (CHAPMAN; BAHILL; WYMORE, 1992, PETROSKI, 1994). A experiência do projetista sobre o domínio de aplicação é de extrema importância, como apresentado por Trinkel (HENEY, 1998) no Capítulo 1, onde nos setores de *hidráulica e pneumática* (H&P) a ausência de *especialistas humanos* (EH) dificulta a utilização de princípios de solução que utilizem estas fontes de energia.

Com as metodologias de projeto é possível ter instruções passo a passo para chegar ao melhor princípio de solução; são na realidade recomendações gerais de etapas a serem percorridas entre as necessidades dos *clientes* e o produto projetado (FRENCH, 1985).

A estruturação do problema e das tarefas no projeto facilita a identificação de soluções otimizadas, sejam elas fundamentadas em projetos anteriores ou com o uso de catálogos com dados de fabricantes, ainda nas fases iniciais e com menor esforço.

As metodologias de projeto propiciam flexibilidade, possibilitando a utilização dos computadores, através da aplicação de modelos de produtos previamente armazenados ou de *base conhecimento* (BC), que proporcionam a divisão das tarefas entre os computadores e os projetistas (PAHL; BEITZ, 1995). Além dos benefícios das metodologias de projeto já citados, French (1985) destaca:

- Aumento da qualidade e velocidade no trabalho do projetista;
- Cooperação entre equipes de trabalho, dentro e fora do escritório de trabalho;
- Documentação das fases de evolução do projeto, com as decisões tomadas em etapas anteriores;
- Comunicação mais rápida e clara entre os projetistas;
- Diversificação na condução do problema;
- Redução dos passos de evolução do projeto, exigindo menor esforço pelo projetista na transposição para etapas com maior detalhamento;
- Sugestão de princípios inventivos e reduzindo mudanças nas revisões do projeto.

Nas metodologias de projeto são utilizadas técnicas e ferramentas como QFD¹, TQM², ES, estratégias de desenvolvimento de produto, projeto para manutenibilidade, projeto para confiabilidade entre outros, que conciliam os esforços dos integrantes da equipe de projeto

¹ QFD (*Quality function deployment*) – Desdobramento da função qualidade (BACK; FORCELLINI, 199-?).

² TQM (*Total quality management*) – Controle da qualidade total (HARTLEY, 1998).

(HUBKA; EDER, 1996, BACK; FORCELLINI, 199-?).

O acompanhamento sistemático que as metodologias, de modo geral, podem oferecer aos projetos estende-se desde as tradicionais aplicações mecânicas, hidráulicas, pneumáticas, elétricas entre outras, como ao projeto de *software*. Metodologias específicas para *software* como nos SE seguem a mesma tendência, porém com algumas adequações ao campo de aplicação, como é apresentado no Capítulo 3. A aplicação das metodologias de projeto à condição do domínio pneumático é apresentado com especificidade no Capítulo 4.

2.2 – Etapas das metodologias de projetos

Para a estruturação das etapas do ciclo de vida dos produtos durante o projeto, diversas referências bibliográficas sugerem a utilização de metodologias que contemplem a passagem pelas seguintes etapas: projeto informacional, conceitual, preliminar e detalhado. (FRENCH, 1985, BACK; FORCELLINI, 199-?).

Na etapa informacional tem-se o início do projeto, a partir do levantamento das necessidades dos *clientes*, buscando transformá-las em requisitos que o projeto deve satisfazer, podendo assim quantificar e traduzir para o ambiente de projeto o que os *clientes* desejam, e com isso direcionar as decisões dos projetistas na próxima etapa, que é a conceitual.

A etapa conceitual caracteriza-se pela geração da função global do sistema, que é desdobrada em sub-funções mais simples que facilitam a pesquisa por princípios de soluções. São muitas as formas de alcançar os princípios de solução no projeto conceitual de um sistema, como leitura, pesquisa, analogias, brainstorming¹, engenharia reversa, entre outras, que podem ser utilizadas em conjunto, pois não são mutuamente exclusivas e devem ser aplicados conforme o problema, quantidade de informação, estágio do processo de projeto, habilidade e experiência dos projetistas. A utilização dos métodos de criação de soluções tem como potencial a geração de várias soluções, que são agrupadas e devem ser avaliadas de acordo com os requisitos dos *clientes* e meta da empresa para que se possa escolher as soluções mais apropriadas que têm prosseguimento no projeto (BACK; FORCELLINI, 199-?, PAHL; BEITZ, 1995).

Posteriormente ao projeto conceitual, tem-se a passagem pelas etapas preliminar e detalhada. Apesar de haver esta seqüência de prosseguimento dentro do projeto, é possível haver o retorno às etapas anteriores, caso sejam constatadas falhas, erros ou simplificações

¹ Brainstorming – técnica utilizada na busca de princípios de soluções inovadoras.

demasiadas. Para minimizar estes erros, evidencia-se que as fases iniciais possuem grande importância e impacto sobre o projeto, pois erros cometidos no início do projeto podem ter consequências que somente são detectadas em etapas muito avançadas, gerando atrasos e custos adicionais ao projeto.

French (1985) sintetiza na Figura 2.1 as etapas do desenvolvimento de projeto de produtos, seqüência esta que também é adotada por Pahl e Beitz (1995) e por Back e Forcellini (199-?).

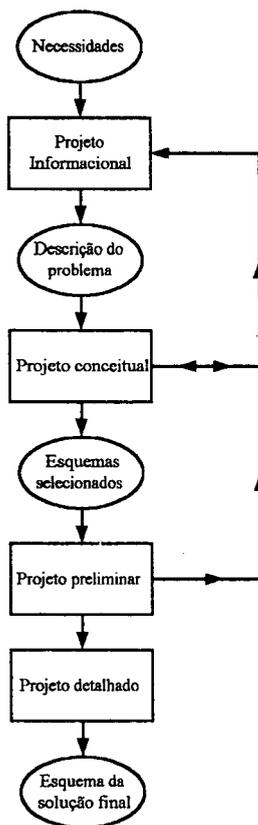


Figura 2.1 – Etapas dos projetos de produto (FRENCH, 1985).

Na Figura 2.1 constata-se a partida para o projeto a partir das necessidades, seguida das contínuas realimentações para obter os refinamentos desejados, até a conclusão do projeto com os detalhes dos esquemas do produto. As realimentações são esperadas apenas até a etapa preliminar, o que não impossibilita que estas ocorram na etapa detalhada; caso isto seja necessário, deve-se analisar o percurso do projeto até esta etapa e buscar constatar em qual ponto do projeto cometeu-se o erro e como ele perdurou sendo identificado apenas próximo da finalização do projeto, para que esta falha não se repita em outros projetos, pois isto certamente terá reflexo sobre os custos e prazos do projeto.

2.3 – Projeto de sistemas

A etapa inicial do ciclo de vida de um sistema é o projeto, no qual as diversas partes que formam o mesmo representam a função global inicialmente definida nas etapas do projeto, a qual pode ser desmembrada em outras sub-funções que possuem menor complexidade, e assim sucessivamente até atingir um nível baixo de complexidade que possa ser manipulado pelo projetista. Neste contexto, Nygaard (apud DE NEGRI, 1996, p. 25) define os sistemas como sendo:

[...] uma parte do mundo que uma pessoa (ou grupo de pessoas) escolhe para considerar como um todo constituído de componentes, cada componente caracterizado por propriedades que são escolhidas como sendo relevantes e por ações relacionadas a estas propriedades e àquelas de outros componentes.

Portanto, constata-se que as fronteiras de um sistema são traçadas conforme as necessidades, e as ações do projetista são delimitadas por estas fronteiras, possibilitando justamente o desmembramento do sistema em componentes que executam sub-funções dentro da função global do sistema. A análise do sistema como sendo formado por sub-funções possibilita a redução da complexidade durante a passagem de níveis de evolução do projeto, tornando gerenciável a atividade do projetista.

Em outra perspectiva, complementando a colocação anterior, Chapman et al (1992, p. 10) considera um sistema como sendo “[...] qualquer processo ou produto que recebe entradas e entrega saídas. [...] muitas vezes, partes dos sistemas são formadas pelo próprio sistema”. Portanto, um sistema projetado pode ser considerado como um subsistema de um outro sistema que o contenha. Esta divisão do sistema em subsistemas oferece maior modularidade ao projeto, pois pode-se utilizar subsistemas previamente existentes, através do desmembramento da função global a ser realizada.

Em construções mecânicas, a complexidade dos módulos pode-se dividir em:

- Grande, quando a montagem do módulo exige vários componentes;
- Pequena, quando o módulo é formado pelo próprio componente.

Em grandes projetos, a modularidade dos sistemas é de grande valia, pois pode-se ter diversas equipes trabalhando em paralelo para a obtenção dos subsistemas, que quando agrupados, formam o sistema final (CHAPMAN, 1992).

Porém, mesmo com o desmembramento os subsistemas em projeto podem exigir a atuação de mais de um profissional devido ao conhecimento multidisciplinar requerido. Os projetos realizados em paralelo requerem uma perfeita interação entre os colaboradores e um bom entendimento dos requisitos de projeto. Abordagens como a ES são de grande valia,

sendo o alvo das atenções da seção a seguir. As ferramentas computacionais também podem auxiliar durante as fases de projeto de sistemas, como o caso dos SE, que oferecem parte do conhecimento multidisciplinar que o projetista deve ter, complementando as deficiências que eventualmente a equipe de projeto tenha sobre um determinado domínio de problema, ou simplesmente sendo uma fonte auxiliar de consulta.

2.4 – Engenharia simultânea

Durante a etapa de projeto os conceitos de engenharia tomam forma. As decisões tomadas na etapa de projeto têm reflexo sobre as demais fases do *ciclo de vida* do produto. Percebe-se, assim, a importância existente na realização de um bom projeto, desde as etapas iniciais. Portanto, para realizar um bom projeto é primordial saber quais áreas têm interações com o produto ao longo do seu *ciclo de vida*, para por fim saber a que requisitos o projeto proposto deve satisfazer para ser considerado um bom projeto.

Pode-se considerar fazendo parte do *ciclo de vida* dos produtos os grupos responsáveis pela manutenção, operação, montagem, transporte entre outros, chegando até o projetista. Portanto, o ponto chave está em coletar os requisitos dos grupos que fazem parte do *ciclo de vida* do produto para que sirvam como metas norteadoras do projeto. Esta nova abordagem proposta, que a princípio é de consenso da grande maioria dos projetistas é, no entanto, poucas vezes colocada em prática devido à grande complexidade decorrente da sua execução, pois tem-se uma grande quantidade de informações sendo processadas. A sua implementação passa pela adoção de uma abordagem multicritério, que considere todas as necessidades dos grupos que fazem parte do *ciclo de vida* do produto, pois podem existir requisitos conflitantes entre si. Com isso, a muralha anteriormente existente entre o projetista e os demais grupos do *ciclo de vida* do produto, que isolava e colocava o projetista em uma situação de conforto, distante das reais necessidades dos *clientes* do projeto já não existe mais, sendo necessário uma maior interação para a obtenção do resultado meta esperado, pois o projetista deve realizar as suas atividades em um menor período de tempo possível, reduzindo assim o *time to market* e agregando qualidade ao projeto.

Para ampliar a abrangência do universo de princípios de soluções em estudo, combinações de diversos domínios como mecânica, elétrica, hidráulica, pneumática entre outros, podem revelar soluções mais eficientes. Porém, devido à amplitude de conhecimento exigido nestes domínios, chega-se a conclusão da inviabilidade de um projeto poder ser realizado por um único projetista, mas sim por um grupo de projetistas, haja vista a multidisciplinaridade requerida.

Em projetos que requeiram mais de um projetista para a sua criação, a sucessão com que ocorre a transposição das etapas do projeto, comumente é utilizado o modelo *seqüencial* ou então o modelo *paralelo convencional*, que não empregam a ES (HAMMER; CHAMPY, 1994).

No modelo *seqüencial*, cada sub-função do projeto é realizada por um projetista que possua o conhecimento técnico requerido. Deste modo, apenas alguns projetistas encontram-se efetivamente trabalhando sobre o projeto, o que demonstra a ineficiência deste modelo devido à não otimização do tempo dos projetistas que encontram-se ociosos durante certo período.

Em grandes projetos com domínios distintos, é preciso desenvolver em paralelo módulos individuais que formam o produto (PAHL; BEITZ, 1995). Sendo adotado o modelo *paralelo convencional*, cada sub-função que se deseja obter é encaminhada aos projetistas que possuam a capacidade técnica para realizá-la. Ao fim de suas tarefas, é feita a união de cada parte que foi projetada. Como os projetistas trabalham de forma independente entre si, e muitas vezes não existe uma perfeita interação entre as parte projetadas devido à deficiência na comunicação entre os grupos de projetistas, geralmente é necessário um reprojeto para adequá-las, demonstrando assim uma não otimização do tempo de projeto como um todo.

A forma com que o projeto de produto era desenvolvido até o início da década de 80 por empresas tradicionais seguia os modelos de projeto descritos no parágrafo anterior. Estes modelos demonstraram-se ineficientes devido ao surgimento de novos fabricantes que produziam com alta eficiência, lançando produtos que satisfaziam as reais necessidades dos *clientes* e que estavam disponíveis no mercado em menor tempo que as suas concorrentes, tendo portanto menores custos de produção, o que as tornavam mais competitivas (HARTLEY, 1998). A essência da alta eficiência destas empresas está na forma de condução das etapas do projeto de produto.

Comparações entre empresas européias, americanas e japonesas no desenvolvimento de produtos similares antes de serem lançados no mercado revela que o *time-to-market* de produtos no Japão é significativamente menor que nos Estados Unidos e na Europa, como apresentado na Figura 2.2, representando os meses antes do lançamento.

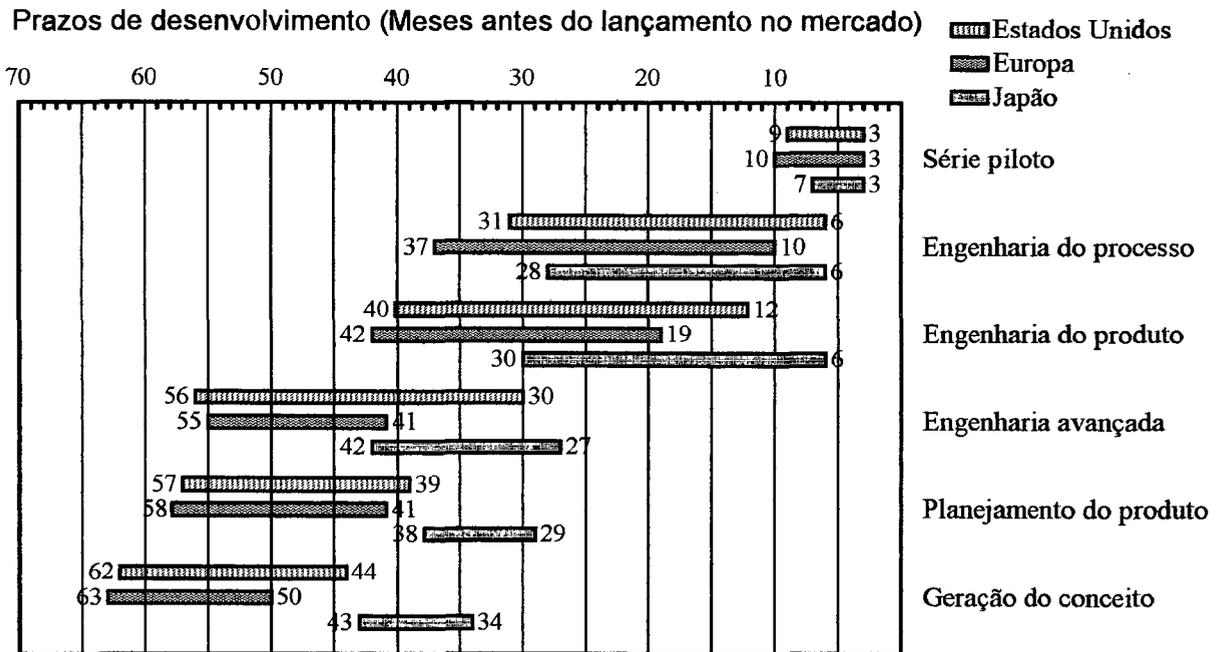


Figura 2.2 – Comparação do tempo de lançamento de produtos (HARTLEY, 1998).

Hartley (1998) aponta como indício da maior eficiência no desenvolvimento de produtos no Japão como sendo atribuído à importância e prioridade que as etapas mais avançadas do projeto têm sobre o produto, iniciando em conjunto com a etapa conceitual, o que facilita a inserção de modificações, como o caso da etapa de engenharia de processo, que possui interação com o projeto desde as etapas iniciais. Com isso, a construção de protótipos nas séries piloto ocorre somente quando tem-se total conhecimento das especificações do produto, reduzindo o número de modificações a serem feitas. Este exemplo demonstra o que vem a ser a ES e como ela pode auxiliar a etapa de projeto de produtos. Porém, a competitividade dos produtos não está associada apenas a etapa de projeto, mas a todo o ciclo de vida do produto, até o fim da sua vida útil (HARTLEY, 1998). A forma diferenciada com que a abordagem de ES trata o projeto em relação às abordagens tradicionais, manifesta a sua eficiência. Na Tabela 2.1 é feito o comparativo direto, ponto a ponto, entre as abordagens tradicionais e de ES.

Tabela 2.1 – Diferenças na condução do projeto (BULLINGER; WARSCHAT, 1996).

Abordagem tradicional	Abordagem de ES
Divisão das tarefas funcionais	Tarefas multifuncionais
Desenvolvimento seqüencial do produto	Desenvolvimento paralelo do produto
Fornecedores como opositores	Fornecedores como associação cooperativa
Modificação do produto	Desenvolvimento de produtos adaptados
Custo como principal instrumento de controle	Tempo como principal instrumento de controle
Diretrizes para a realização de objetivos	Lucro como objetivo principal
Indecisão às modificações de mercado	Tendência ao cenário de mercado
Ocultação das deficiências organizacionais	Revelação das deficiências organizacionais
Trabalho paralelo em outros projetos	Dedicação a um único projeto

A eficiência que a ES agrega ao projeto está, em grande parte, associada à existência de canais de comunicação eficientes entre os projetistas e os *clientes* do produto, que tem voz ativa no projeto. Isto propicia o conhecimento das necessidades dos *clientes* desde as fases iniciais, reduzindo portanto o número de correções e modificações que devem ser feitas no projeto. Chapman et al (1992) apresentam na Figura 2.3 o atraso que ocorre no lançamento de produtos em empresas que seguem a abordagem tradicional de projeto, quando comparadas às que utilizam a ES nas etapas de projeto, tornando possível assim reduzir o número de modificações no projeto, e quando estas são necessárias, elas ocorrem ainda nas fases iniciais, longe do início da produção.

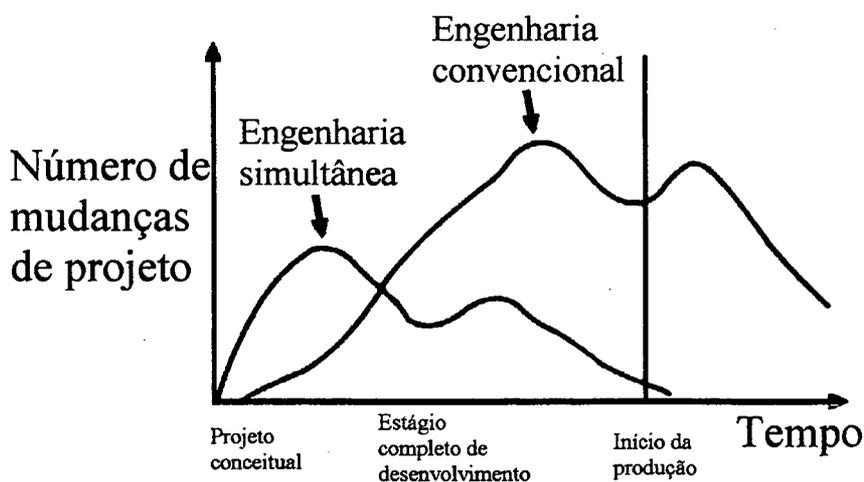


Figura 2.3 – Período de lançamento dos produtos (CHAPMAN et al, 1992).

Hartley (1998) demonstra na Figura 2.4 que as constantes alterações no produto não somente levam ao atraso no lançamento, como já comentado na Figura 2.2, mas também

geram queda de competitividade tanto devido à redução do período de rentabilidade do produto no mercado quanto dos gastos decorrentes das correções para que o mesmo possa ser lançado. Com a ES tem-se a redução no número de alterações no projeto, o que possibilita atingir o *melhor caso* de rentabilidade durante o percurso de evolução do projeto, apresentado na Figura 2.4.

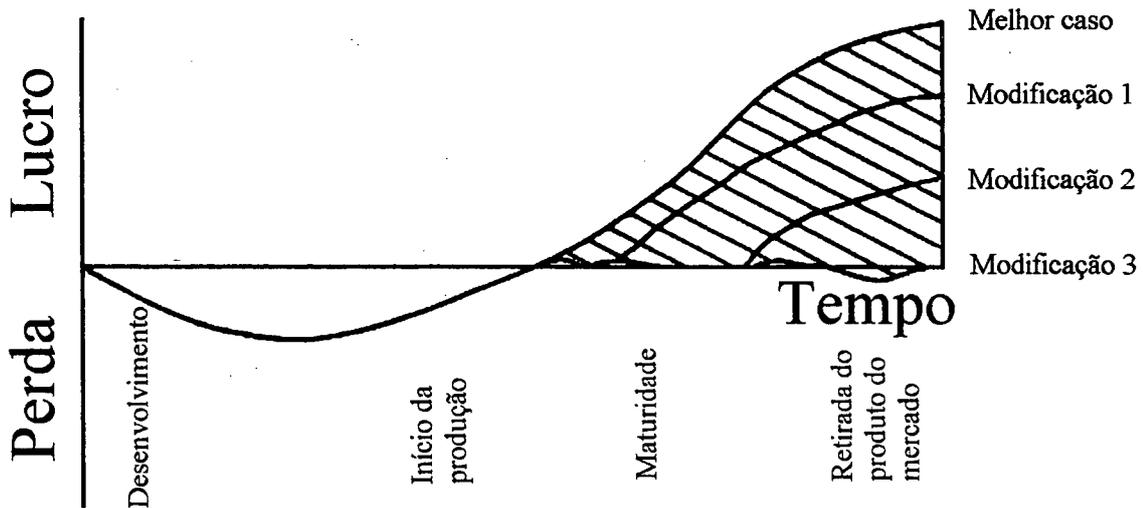


Figura 2.4 – Queda de competitividade devido às mudanças (HARTLEY, 1998).

O princípio de eficiência da ES pode ser justificado também pela etapa em que as modificações de projeto são realizadas. Segundo dados da *Computer Aided Manufacturing International* citados por Miller (1993) e apresentados na Tabela 2.2, constata-se o alto comprometimento dos custos sobre o produto nas fases iniciais do projeto, apesar do baixo investimento até então realizado. Portanto, as modificações nas etapas iniciais do projeto não indicam grandes perdas pelo baixo investimento despendido até então, o que não ocorre durante as etapas finais do projeto, em que todo o investido é afetado pelas modificações decorrente das adequações.

Tabela 2.2 – Custo das etapas de projeto (MILLER, 1993).

Etapa	Porcentagem do custo total – valores acumulativos	
	Gastos realizados	Comprometimento dos gastos
Concepção	3 % – 5 %	40 % – 60 %
Projeto de engenharia	5 % – 8 %	60 % – 80 %
Teste	8 % – 10 %	80 % – 90 %
Planejamento do processo	10 % – 15 %	90 % – 95 %
Produção	15 % – 100 %	95 % – 100 %

Analisando os dados da Tabela 2.2 em conjunto com dados coletados junto a empresas de produtos eletrônicos feitos pela *Dataquest Inc.* citados por Miller (1993), referentes a custos decorrentes de mudanças ao longo do projeto, evidencia-se a necessidade de constatar ainda nas fases iniciais do projeto possíveis enganos cometidos pelos projetistas, que direcionem a mudanças de paradigma no projeto, como apresentados na Tabela 2.3. Assim, soluções mais vantajosas podem ser propostas, e como o custo até então despendido com o projeto ainda é muito baixo, não existe o comprometimento do orçamento final do projeto e a perda de competitividade do produto. Chapman et al (1992) consideram que mudanças após o início da produção indicam o quanto pobre foi o tratamento de informações do projeto, que apresenta deficiências no planejamento e execução. A qualidade do produto deve estar presente portanto, desde as etapas iniciais do seu *ciclo de vida*. O sucesso de um produto não esta condicionado apenas a qualidade durante a fabricação, mas durante também o projeto.

Tabela 2.3 – Custo típico decorrente de mudanças (MILLER, 1993).

Etapa	Custo
Durante o projeto	\$ 1.000
Durante testes do projeto	\$ 10.000
Durante planejamento do processo	\$ 100.000
Durante testes de produção	\$ 1.000.000
Durante produção final	\$ 10.000.000

Apesar da grande diversidade de setores industriais pesquisados referentes aos benefícios da ES nos projetos, evidencia-se que a ES auxilia nas etapas de projeto, independente da área de aplicação.

Diversas aplicações podem beneficiar-se da ES. Em projetos pneumáticos, a evidência antecipada de possíveis simplificações ou considerações errôneas durante a etapa de projeto podem ser evidenciadas ainda durante o início do projeto, reduzindo assim o tempo com o dimensionamento, montagem das instalações ou até mesmo a substituição de componentes que já foram adquiridos, mas que não conseguem satisfazer as condições de operação previamente estabelecidas. A coleta de informações antes do início do projeto passa a ser determinante na eficácia do projeto às condições de operação.

Em projeto de *software* de modo geral, a coleta das informações referentes às necessidades dos *clientes* também é muito importante e a ES pode ser empregada desde as etapas iniciais, podendo oferecer melhorias na adequação do produto à situação de operação.

2.5 – Computadores no auxílio ao projeto

Desde o início das pesquisas com o desenvolvimento dos computadores, diversas áreas têm se beneficiado da sua utilização, em aplicações que simplesmente substituam ou então auxiliem parcialmente o homem. No setor de engenharia, o meio computacional ou o uso do computador auxilia no processo de produção, com a utilização de ferramentas computacionais do tipo CAD, CAE, CAM, ou na integração do computador a máquinas e processos como os CNC e os CIM, dentre tantas outras aplicações que são desenvolvidas para complementar as atividades de engenharia. Todas estas ferramentas são utilizadas no intuito de auxiliar e aprimorar a elaboração do produto com qualidade e rapidez. Porém, a velocidade de operação dos computadores pouco pode ser explorada se os *software* existentes oferecem poucos recursos que agreguem real valor ao projeto. O conhecimento especializado sobre os processos produtivos e de projeto são o ponto de partida para potencializar ferramentas computacionais que realmente auxiliem na condução do projeto.

Uma ferramenta que se destaca no auxílio ao projeto são os SE. Diversos setores têm se beneficiado da especialidade que estes *software* oferecem em atividades variadas de projeto. Informações vitais sobre os processos de fabricação como espessura de parede permissível, raio de aresta, ângulo de saída para peças conformadas ou moldadas, deformação dos materiais entre outros dados, que podem estar disponíveis aos projetistas, inserindo no ambiente de projeto a integração que a ES busca oferecer com os demais setores que fazem parte do *ciclo de vida* do produto.

Dym e Levitt (1991, p. 190) também destacam a aplicação dos SE como ferramenta de auxílio ao projeto, permitindo “[...] a integração do conhecimento em múltiplos níveis de abstração em domínios como a ES, a qual envolve múltiplas fontes de especialidades em um mesmo nível de abstração”. Como exemplos de aplicações dos SE no auxílio das etapas de projeto, cita-se:

- X-CON inicialmente chamado R1, que historicamente dá início à utilização dos SE ao projeto, em aplicações de auxílio à configuração de computadores da *Digital Equipment Corporation* (DEC), segundo personalização às necessidades dos consumidores (PARSAEI; SULLIVAN, 1993);
- *Moldflow*, *software* no auxílio à simulação da moldagem por injeção (HARTLEY, 1998)
- *Software* de auxílio ao projeto do *layout* de placas de circuito impresso, desenvolvido por algumas empresas (HARTLEY, 1998);

- PRIDE, *software* que segue a mesma linha de configuração do X-CON porém, atuando no projeto mecânico de fotocopiadoras (PARSAEI; SULLIVAN, 1993, DYM; LEVITT, 1991);
- *Schemebuilder – expert fluid power module*, atuando no projeto de sistemas hidráulicos (SILVA, 1998).

Como apresentado neste capítulo, a etapa conceitual dos projetos representa uma etapa muito importante nos mesmos, sendo a abordagem da ES uma opção conveniente. Portanto, dos objetivos inicialmente apresentado na seção 1.4, em que é destinada atenção prioritária as etapas conceituais dos projetos pneumáticos, evidencia-se o apropriado direcionamento dado ao trabalho, sendo a construção de um *software* de auxílio às tarefas dos projetistas um avanço no modo de construção de esquemas pneumáticos.

Paralelamente, as considerações apresentadas neste capítulo podem também ser estendidas ao auxílio à construção do protótipo computacional em si, esperando-se assim proporcionar melhores resultados durante a sua implementação.

No próximo capítulo são apresentados detalhes sobre a construção de SE, bem como a aplicação de metodologias de projeto no seu desenvolvimento, que em grande parte assemelha-se ao ciclo de desenvolvimento de produtos discutidos neste capítulo, com algumas adequações à condição específica de desenvolvimento de SE.

Capítulo 3 – Sistemas especialistas

O desenvolvimento de novas aplicações para a utilização dos computadores em conjunto com o grande desenvolvimento de *hardware* cada vez mais velozes e poderosos, têm requerido maior exploração das áreas de pesquisa em *software*, que supram e ofereçam a funcionalidade que é esperada de um computador.

Neste contexto, a *inteligência artificial* (IA) possui ainda um vasto campo de aplicações a serem exploradas, pois se tem um amplo horizonte de situações de utilização para os sistemas computacionais em conjunto com a grande quantidade de informações disponíveis, porém ainda não totalmente decifrada e repassada para o meio eletrônico, através de ferramentas computacionais que disponibilizem estes dados conforme a necessidade e aplicação do usuário. Dentre as diversas vertentes de pesquisa da IA, é explorada em mais detalhes neste capítulo a área dos *sistemas especialistas* (SE).

Para tornar as etapas de desenvolvimento de *software* devidamente fundamentadas, referente aos direcionamentos e decisões tomadas no seu processo de elaboração, são exploradas metodologias de projeto para *software*, para assim como as metodologias de projeto em aplicações tradicionais, como apresentado no Capítulo 2, obter os benefícios de sua utilização.

Por fim, são exploradas as técnicas de modelagem em objetos, fundamental em aplicações que envolvem extrema complexidade e que exigem a manipulação de muitos dados, facilitando a perfeita interação do meio computacional com o mundo real o qual é desejado modelar e interagir com os computadores.

3.1 – Abordagens de IA e SE

Para uma definição inicial de IA, pode-se dizer que “é a parte da ciência computacional que busca transformar e criar programas computacionais com capacidade de *pensar*, assim como o ser humano” (WATERMAN, 1986, p. 3, grifo nosso).

Dentre as diversas definições sobre IA, segue na Tabela 3.1 algumas referências complementares sobre o que é IA, suas aplicações e características.

Tabela 3.1 – Definições de IA.

Fonte	Definição de IA
Buchanan; Shortliffe (1984, p. 3)	“IA é a parte da ciência da computação que utiliza relações simbólicas e não algorítmicas na solução de problemas”
Dym; Levitt (1991, p. 10)	“[...] é a ciência que tenta reproduzir o comportamento inteligente humano em computadores”
Durkin (1994, p. 3)	“[...] um campo de estudo na ciência da computação que desempenha a finalidade de fazer um computador raciocinar de modo similar aos humanos”
Rich (1988, p. 1)	“[...] é o estudo de como fazer os computadores realizarem tarefas em que, no momento, as pessoas são melhores”

Dentre as diversas técnicas pesquisadas pela IA, sendo cada uma delas aplicada a uma determinada função, tem destaque os SE que assim como o objetivo principal da IA, tentam simular tarefas realizadas pelo homem, em atividades que necessitam de conhecimento especializado, como a que os *especialistas humanos* (EH) possuem, justificando a origem do nome *sistemas especialistas*. Na identificação da função do EH, Durkin (1994, p. 44, grifo nosso) define-o como sendo “uma pessoa que possui a *habilidade e conhecimento* para solucionar um *problema específico* de modo *superior* aos outros”, e a essência do seu conhecimento é a diferença existente entre eles e os não especialistas, através da relação:

$$\text{Conhecimento} = \text{Especialista} - \text{Não Especialista}$$

Equação 3.1 – Conhecimento dos especialistas (DURKIN, 1994).

Assim como os EH que possuem grande habilidade na solução de problemas que fazem parte da sua área de especialidade, os SE são elaborados para atuar em áreas específicas do conhecimento, intituladas de *domínio do conhecimento* (DC). O objetivo principal dos SE é inserir este conhecimento especializado, que é encontrado nos EH, em programas computacionais.

Na essência, os SE são um agrupamento de sistemas computacionais que solucionam problemas no domínio de interesse (WATERMAN, 1986). A tarefa de transferência do

conhecimento entre o EH e o computador é realizada pelo *engenheiro do conhecimento* (EC), que coleta o conhecimento, utilizando formas diversas, e codifica este no computador, formando a *base de conhecimento* (BC) para um determinado DC específico. A BC é o acumulador de *fatos* e associações do programa em áreas de aplicação diversas (BUCHANAN; SHORTLIFFE, 1984). Na Figura 3.1 é representada a estrutura elementar de transferência de conhecimento do EH para o SE, com a intervenção do EC, bem como a atuação de outros profissionais na realização do SE.

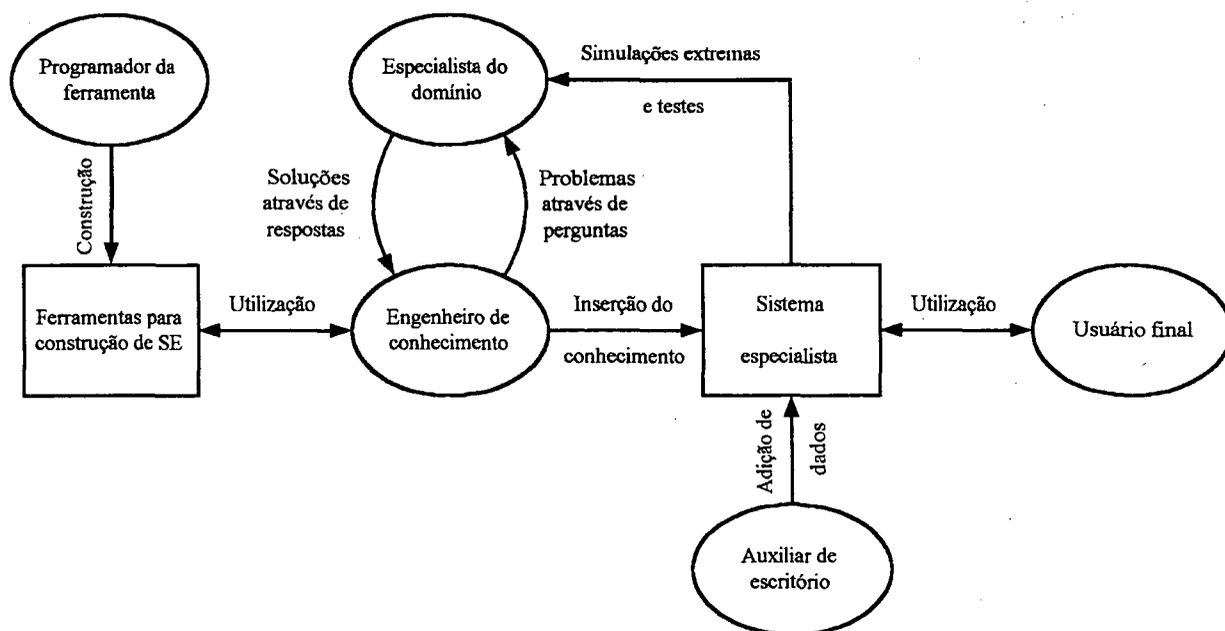


Figura 3.1 – Interações na construção dos SE (WATERMAN, 1986).

No ciclo de desenvolvimento do SE, evidencia-se outras interações como a do programador de desenvolvimento do ambiente de SE e de implementação da BC, que ainda pode ser complementado por um auxiliar de escritório para atividades rotineiras. Em projetos de menor escala, estas funções podem ser conduzidas pelo próprio EC. Por fim, existe a função do usuário final, que é o beneficiado pelo SE.

3.1.1 – Origem da abordagem de SE

Historicamente, no início da utilização dos computadores, os mesmos eram escalados para aplicações numéricas ou de processamento de dados (BUCHANAN; SHORTLIFFE, 1984).

Dentre estas aplicações, os modelos utilizados nos computadores podem ser divididos conforme o tipo de informação manipulada, na seguinte classificação (DYM; LEVITT, 1991):

- Modelos de princípios básicos, utilizando as leis fundamentais da física clássica, como as leis de Newton, conservação de massa, momento, energia entre outras;
- Modelos de fenômenos, expressos através de razões, equações diferenciais ou outras formas matemáticas não discretas, obtidas a partir de resultados experimentais ou de extrapolações das leis fundamentais;
- Modelos analíticos, que expressam de forma exata ou aproximada casos específicos ou partes dos modelos de fenômenos ou de princípios básicos, empregando equações matemáticas em conjuntos contínuos;
- Modelos numéricos, que possuem aplicação similar aos modelos analíticos, porém em condições discretas, como no caso de estudo de elementos finitos.

A aplicação destes modelos em programas computacionais implica na utilização de abordagens algorítmicas que conforme Buchanan e Shortliffe (1984, p. 3) é um “procedimento que garante achar a resposta correta em um tempo finito ou indicar que não existe solução”.

Mas no caso de problemas de origens diversas, que necessitam de processamento de dados em que os modelos anteriormente citados não demonstram eficiência, utiliza-se os modelos heurísticos, fundamentados em conhecimentos empíricos, obtendo soluções em menor tempo, ao contrário das abordagens algorítmicas de *tentativa e erro* (BUCHANAN; SHORTLIFFE, 1984). A heurística é fundamentada em instruções ou opiniões que não necessariamente estão fundamentadas na ciência (HUBKA; EDER, 1996). A formulação intuitiva na solução de problemas é algo inerente para o domínio de projeto, em que a heurística é aplicada em condições que sejam exigidas decisões com a análise de muitas informações (ROOZENBURG; EEKELS, 1995). O modelo heurístico é o conceito fundamental da técnica de SE, que tem como característica a simulação das habilidades humanas pelos computadores.

Durante o ciclo evolutivo dos SE, a partir da origem entre a década de 60 até a primeira metade da década de 70, as pesquisas eram direcionadas para a obtenção de métodos para a solução de problemas de forma generalizada, que devido à grande amplitude do campo de atuação, não ofereciam grandes recursos na solução de problemas complexos. Porém, na segunda metade da década de 70, as pesquisas direcionaram-se para aplicações em DC específicos, que apesar de limitarem o campo de atuação dos SE, ofereciam grande

performance em operação (WATERMAN, 1986).

3.1.2 – Diferenças entre as abordagens algorítmicas e heurísticas

Ao contrário das abordagens computacionais algorítmicas, que Buchanan e Shortliffe (1984) definem como sendo ineficientes em situações em que as informações a serem processadas são predominantemente qualitativas ao invés de quantitativas, as abordagens heurísticas buscam suprir esta deficiência de representação.

A essência na diferença entre as abordagens algorítmicas e as heurísticas pode ser ilustrada por um exemplo geral apresentado por Waterman (1986) em um sistema para detecção de seqüestradores de aviões, apresentado na Figura 3.2.

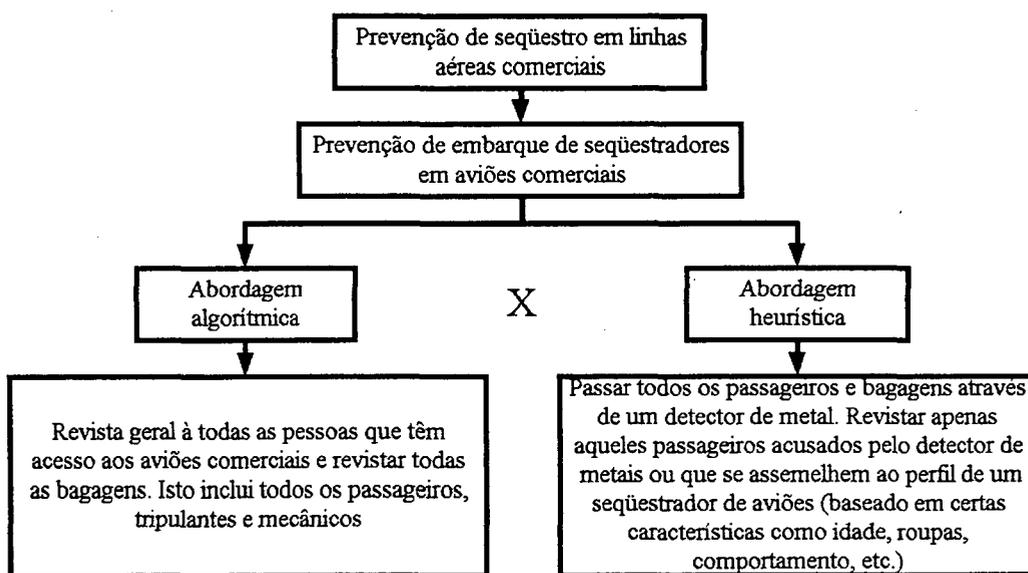


Figura 3.2 – Abordagens algorítmicas versus heurísticas (WATERMAN, 1986).

Na situação do exemplo da Figura 3.2, a abordagem algorítmica certamente necessita de muito tempo com um alto custo. Em contrapartida, a abordagem heurística oferece uma solução mais simples e prática, porém não possui total garantia de eficiência. No contexto deste exemplo, conclui-se que os SE buscam estabelecer atalhos e simplificações, considerando a essência na solução dos problemas.

A preferência pela abordagem heurística é devido às características do próprio domínio do problema, como em aplicações complexas em que o número de possibilidades a serem examinadas é muito grande. Outra situação é quando o algoritmo aplicável que oferece a resposta correta ao problema é muito complexo, ou é uma aproximação, ou na pior das

situações, simplesmente não existe (GONZALEZ; DANKEL, 1993):

A diferença entre as abordagens algorítmicas e as heurísticas é refletida nas características e na forma de construção do *software*, como demonstrado na Tabela 3.2.

Tabela 3.2 – Diferenças entre programas computacionais convencionais e SE (DYM; LEVITT, 1991, DURKIN, 1994, WATERMAN, 1986).

	Programas convencionais	SE
Representação	De dados	De conhecimento
Conhecimento e controle	São integrados	São separados
Processamento	Algorítmico	Inferencial
Manipulação	Grandes bancos de dados	Grandes BC
Desenvolvimento	Mais rígido	Mais flexível
Capacidade de explanação	Impossível	É uma característica
Entidades manipuladas	Numéricas	Simbólicas
Modificações	Difícil	Fácil
Informações processadas	Precisas	Com incerteza
Centrado	Solução	Problema
Desenvolvimento	Programador trabalha sozinho	Esforço de uma equipe
Etapas de desenvolvimento	Seqüencial	Interativo

Apesar de em ambos os casos o resultado final ser o mesmo, isto é um *software*, que para o usuário pode ser imperceptível algumas especificidades, pois a diferenciação ocorre apenas na forma de abordar o problema, o resultado dos SE, em alguns casos, pode ser superior aos dos programas computacionais convencionais.

Outra característica dos SE relatado por Gonzalez e Dankel (1993), refere-se à facilidade de modificação e atualização dos SE, devido à separação entre a BC em relação as demais partes do programa. Outra vantagem é a possibilidade de solução de problemas com dados incompletos ou que possuam incerteza.

3.1.3 – Aplicação da técnica de SE

A aplicação da técnica de SE deve ser explorada em situações específicas, que justifiquem a sua aplicação ao invés da utilização do EH. Na Tabela 3.3, Durkin (1994) e Waterman (1986) confrontam algumas características relevantes na comparação da utilização dos SE em relação aos EH.

Tabela 3.3 – Comparação entre EH e SE (DURKIN, 1994, WATERMAN, 1986).

	EH	SE
Tempo disponível	Horário de trabalho	Todo o tempo
Disponibilidade geográfica	No local	Em qualquer lugar
Segurança	Insubstituível	Substituível
Percibilidade	Sim	Não
Performance	Variável	Constante
Velocidade	Variável	Constante, usualmente rápida
Custo	Alto e permanente	Apenas no desenvolvimento
Criatividade	Sim	Não
Atualizações	Adaptativo	Necessita de atualizações
Entrada de dados	Experiência sensorial	Entrada simbólica

Dentre os itens listados na Tabela 3.3, os fatores *percibilidade* e *segurança* merecem destaque, que em certas circunstâncias podem ser antagônicos. O que leva a utilização da técnica de SE, seja em contextos competitivos (ambientes empresariais) ou em contextos essenciais (setores médicos), é a necessidade de reter o conhecimento e disponibilizá-lo de forma perene.

Se por um lado a opacidade na obtenção de informações essenciais identificadas somente com EH é considerada como uma *segurança*, pois pessoas estranhas não possuem contato com dados sigilosos, isto demonstra também a vulnerabilidade e a alta dependência de uma única pessoa ou poucas pessoas, o que pode ser prejudicial.

Razões diversas como aposentadoria, relocação de pessoal ou transferência para empresas concorrentes, ou até mesmo razões inesperadas como morte, indicam a necessidade de representação do conhecimento que os EH possuem, que além de tornar o conhecimento permanentemente disponível, pode ser usado como treinamento de novos EH para exercer a mesma função, por estar disponível em *tempo integral* em *qualquer lugar* (DURKIN, 1994).

Outro conjunto de itens de relevância da Tabela 3.3 são *criatividade*, *atualização* e modo de *entrada de dados*. Apesar dos benefícios dos SE, estes itens refletem algumas das deficiências dos mesmos. Referente ao item *atualização*, pelo fato dos SE serem desenvolvidos para exercerem funções de modo similar ao que os EH realizam, o campo de atuação é limitado ao conhecimento inserido e a performance depende da *atualização* de novos conhecimentos, que no caso dos EH é feita através de novas experiências profissionais, cursos de atualização entre outros. No caso dos SE as *atualizações* são mais elaboradas e feitas em etapas mais complexas de *engenharia do conhecimento*. Por fim, as limitações dos SE devido à inexistência de *criatividade* e a limitação na *entrada de dados*, impossibilitam a

coleta de informações utilizando a intuição como os EH o fazem, o que não permite a construção de *software* com desempenho similar ao que o EH possui.

Além dos itens listados na Tabela 3.3, deve-se considerar outros tópicos adicionais que descrevem os SE, que dependendo da aplicação a qual o SE será empregado, algumas desvantagens devem ser previamente conhecidas para evitar a sua utilização em situações incorretas. Dentre as limitações dos SE é citado (GONZALEZ; DANKEL, 1993):

- As respostas de saída dos SE podem conter erros, pois as fontes de conhecimento, que são os EH, também cometem erros, sendo estes possíveis de serem transferidos para a BC do SE;
- Não é inerente aos SE a capacidade de distinguir informações de *sensu comum* como o homem possui. Para isto, é preciso implementar estas funções na BC que, dependendo da situação, pode ser muito complexo;
- Limitação de atuação do SE apenas no seu DC.

Devido ao elevado custo envolvido na construção dos SE, Waterman (1986) reestrutura e complementa alguns dados da Tabela 3.3, conforme o tipo de questões que se deseja responder, sugerindo um guia para identificar a viabilidade da elaboração da técnica de SE, segundo critérios referentes as suas limitações e justificativas para a sua implementação sobre a perspectiva de limitações do emprego de EH e características do domínio do problema, como apresentado na Tabela 3.4.

Tabela 3.4 – Critérios para a seleção dos SE (WATERMAN, 1986).

Possibilidade de desenvolvimento do SE		Justificativa do desenvolvimento do SE		Desenvolvimento do SE é apropriado		
E	Tarefa não requer sensu comum	On	Custo com EH grande	Natureza	Tarefa requer manipulação simbólica	
	EH podem articular seus métodos		Perda do EH		Tarefa requer solução heurística	
	Existe EH		EH raro	Complexidade	Tarefa não é tão fácil	
	Há consenso entre EH		EH requerido em vários locais	E	Escopo	Tarefa tem valor prático
	Tarefa requer também habilidades heurísticas		Atuação do EH em ambiente hostil			Tarefa é de proporção gerenciável
	A tarefa é plenamente explorada e conhecida					

Antecedendo qualquer proposta de desenvolvimento de um SE, é conveniente seguir as constatações de adequação desta técnica conforme a Tabela 3.4.

3.1.4 – Campos de atuação dos SE

A expansão da aplicação dos SE na década de 90 demonstra a constatação dos benefícios desta área da IA. Os dados da pesquisa feita por Durkin (1994) resultam na constatação de aproximadamente 2.500 SE desenvolvidos em 1992. Estes dados, porém, estima-se representarem algo em torno de 20% do universo pesquisado, sendo portanto o número real em torno de 12.500 SE. Esta quantidade indica um grande crescimento dos SE, quando comparado com a quantidade de SE criados em outros anos, indicando um retrospecto de crescimento na utilização, como ilustrado na Figura 3.3.

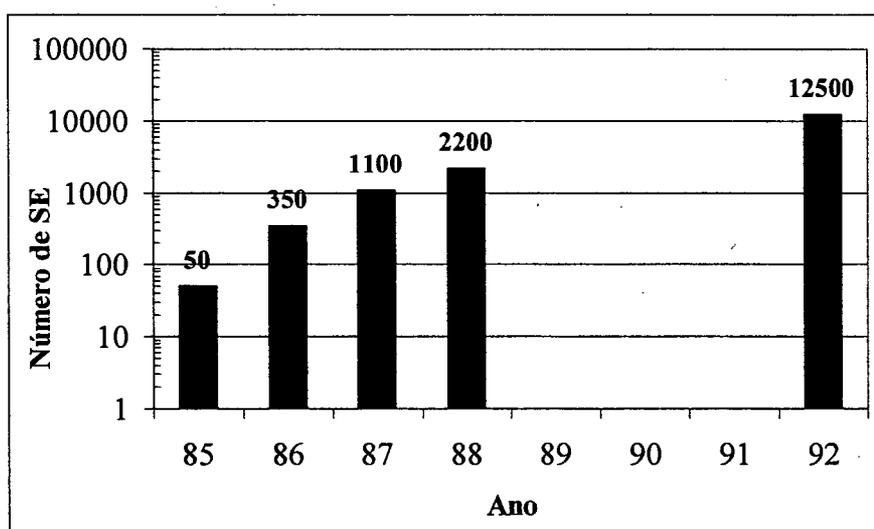


Figura 3.3 – Número de SE desenvolvidos por ano (DURKIN, 1994).

Da totalidade de 2.500 SE relacionados por Durkin (1994) no ano de 1992, na Figura 3.4 tem-se os diversos setores beneficiados pelos SE. Destaque para as aplicações dos SE em engenharia, ocupando a 5ª colocação em número de SE construídos.

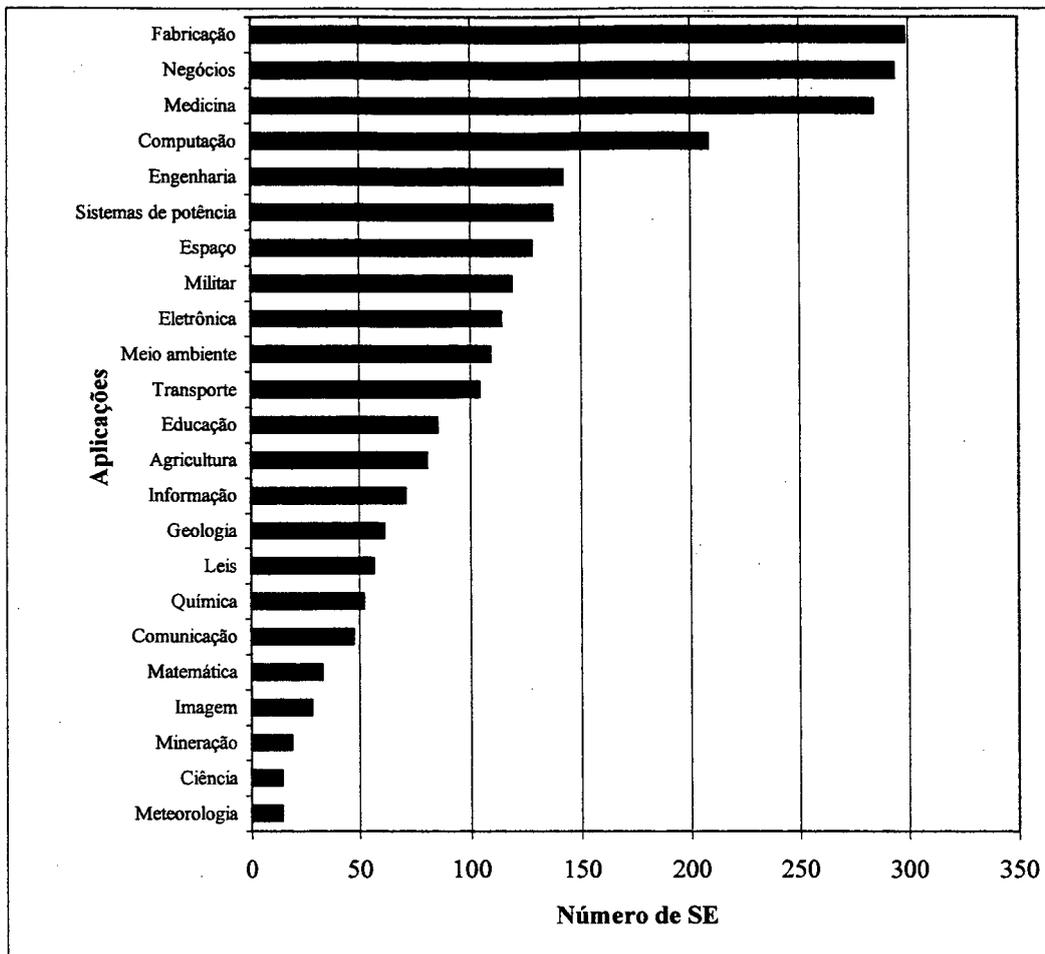


Figura 3.4 – Número de SE desenvolvidos por tipo de aplicação (DURKIN, 1994).

Das diversas funções possíveis de serem realizadas pelos SE, Durkin (1994) obteve em sua pesquisa a distribuição ilustrada na Figura 3.5.

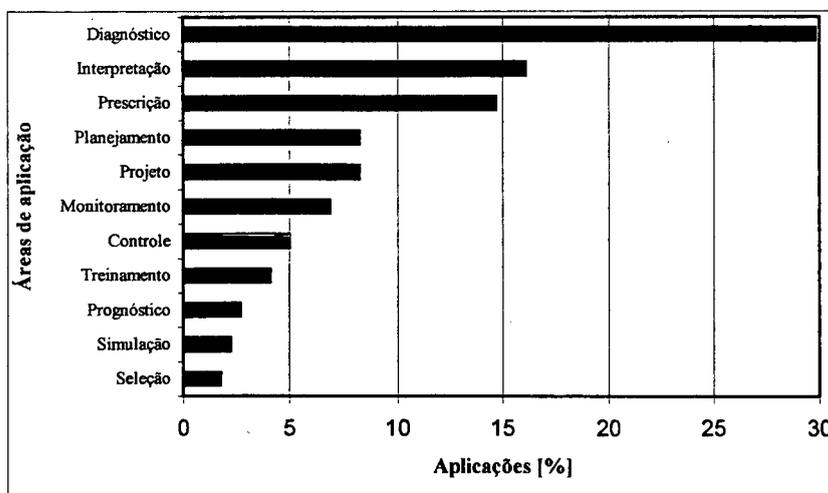


Figura 3.5 – Funções exercidas pelos SE (DURKIN, 1994).

Destaque para as aplicações de diagnóstico, devido à forma heurística com que os EH utilizam na identificação de falhas e os procedimentos para solução, como no caso do SE desenvolvido por Alves (2001) para a identificação de falhas de operação em sistemas hidráulicos navais. As aplicações de projeto e planejamento também têm destaque dentro das possibilidades de funções realizadas.

3.1.5 – Estrutura computacional dos SE

No desenvolvimento de SE são utilizados programas computacionais específicos, que possuem fundamentação nesta técnica, pois tratam o DC de forma diferenciada em relação aos demais programas computacionais convencionais que seguem abordagens algorítmicas. Dentre os elementos existentes nos SE, conforme Figura 3.6, cita-se:

- BC, que é formada pelo conhecimento específico em um determinado DC, construído pelo EC com auxílio do EH ou de outro tipo de fonte de conhecimento. Durante a utilização do SE, a BC permanece sempre inalterada;
- Memória de trabalho, que armazena os dados fornecidos pelo usuário durante a utilização do SE, os quais representam a descrição do problema a ser solucionado. Para cada seção de operação, a memória de trabalho é limpa, para que dados de uma determinada seção não sejam analisados em conjunto com dados de seções anteriores;
- Seletor de *regras* ou *agenda*, que realiza a correlação entre os dados do problema fornecido pelo usuário, com os dados existentes na BC, que dependendo da situação, pode gerar mais de um encadeamento ou caminho a ser percorrido, conhecido como *conflito*. Nesta etapa, os SE diferenciam-se dos programas computacionais que adotam os procedimentos algorítmicos, devido à prioridade que pode ser dada a um determinado encadeamento, conforme especificação prévia do EC para a solução dos *conflitos* gerados;
- Interpretador de *regras* ou *máquina de inferência*, que executa os procedimentos existentes quando da seleção de um determinado encadeamento.

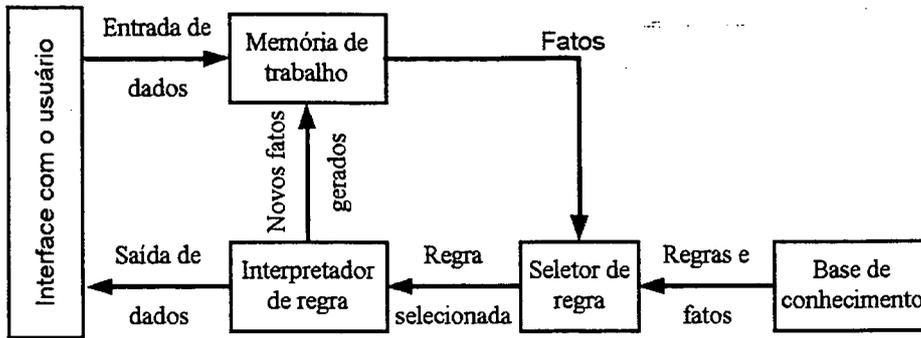


Figura 3.6 – Elementos computacionais dos SE (DYM;LEVITT, 1991).

3.1.6 – Sistemas de produção ou regras

A unidade elementar utilizada na formação da BC dos SE são as *regras*. A estrutura fundamental das *regras* pode ser representada pela Figura 3.7 em uma relação *se/então*, onde as condicionais da *regra* localizadas entre os termos *se* e *então*, também conhecidas como lado *esquerdo* da *regra*, são satisfeitas por *fatos* ou por qualquer outra forma de manipulação do conhecimento.

Havendo a satisfação das condicionais da *regra*, pode-se dar prosseguimento as ações existentes após o item *então*, conhecido como lado *direito* da *regra*. Os *fatos* manipulados pelas *regras* podem ter origem tanto na entrada de dados do usuário, como através do próprio SE, em um encadeamento para a busca de soluções.

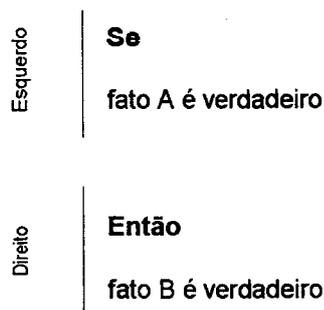


Figura 3.7 – Estrutura padrão das regras (BUCHANAN; SHORTLIFFE, 1984).

A BC de um SE, dependendo da abrangência e complexidade, é formada por diversas *regras*, as quais têm interações umas com as outras. Porém, estas *regras* devem ser inseridas na BC buscando evitar informações redundantes, que levem a *looping* ou que possuam conclusões contraditórias.

Gonzalez e Dankel (1993, p. 88) definem o processo de solução de problemas em SE como sendo “a criação de uma série de inferências que criam um caminho entre a definição do problema e a sua solução”.

Dentro do domínio de aplicação dos SE, as *regras* representam a menor parte do conhecimento transcrito para a BC. Ao contrário dos programas computacionais convencionais, as *regras* que formam os SE conferem alta modularidade¹.

Esta modularidade deve-se à inclusão, exclusão ou alteração de qualquer uma das *regras*, que não necessariamente implica na alteração de outras *regras*. Além disso, as *regras* oferecem maior uniformidade na formação da BC, e a relação *se/então* existente nas *regras* é a forma natural com que o conhecimento em um domínio é manipulado pelos EH, facilitando o trabalho do EC (GONZALEZ; DANKEL, 1993).

3.1.7 – Tipos de encadeamentos entre *regras*

Dependendo da natureza da aplicação, os problemas podem ser classificados como sendo de *derivação* ou de *formação*, conforme apresentado na Figura 3.8. Cada um destes tipos de problemas exigem abordagens distintas a serem exploradas pelo EC, na formação do caminho entre os dados do problema e a sua respectiva solução, conforme o conhecimento representado e organizado no SE (WATERMAN, 1986).

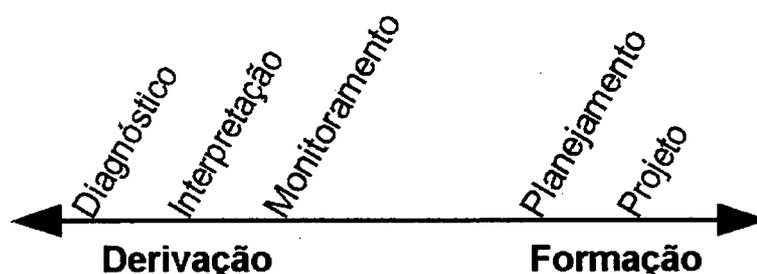


Figura 3.8 – Classificação dos tipos de DC (DYM; LEVITT, 1991).

A característica da abordagem de *formação* é de obter a solução do problema agregando e gerando elementos que, em conjunto, dão origem a um *novo objeto*; ao contrário, a abordagem de *derivação* manipula informações originadas de *fatos* e dados de *objetos previamente existentes* (DYM; LEVITT, 1991).

¹ Buchanan e Shortliffe (1985, p. 39) definem modularidade em programas computacionais como “o grau de separação de suas unidades funcionais em partes isoladas”.

Dentro das abordagens de *formação* ou *síntese*, são utilizados os encadeamentos entre *regras* do tipo *direto* nos quais o processo de busca de solução tem início com os dados do problema que conduzem a respectiva solução. Ao contrário, nas abordagens de *derivação* ou *classificação* é utilizado o encadeamento *reverso*, pois é percorrido o caminho inverso da abordagem anterior, em que a partida para o processo de busca de solução tem início com a geração de uma determinada hipótese, a qual deverá ser provada a sua veracidade a partir da entrada de novos dados do problema pelo usuário (GONZALEZ; DANKEL, 1993).

Para tornar mais simples o entendimento destes tipos de encadeamentos em relação ao fluxo de informação ao longo do encadeamento de *regras*, Chorafas (1990) na Figura 3.9 compara simultaneamente o procedimento de encadeamento de *regras* e *fatos*, desde o início do problema até a chegada na sua solução.

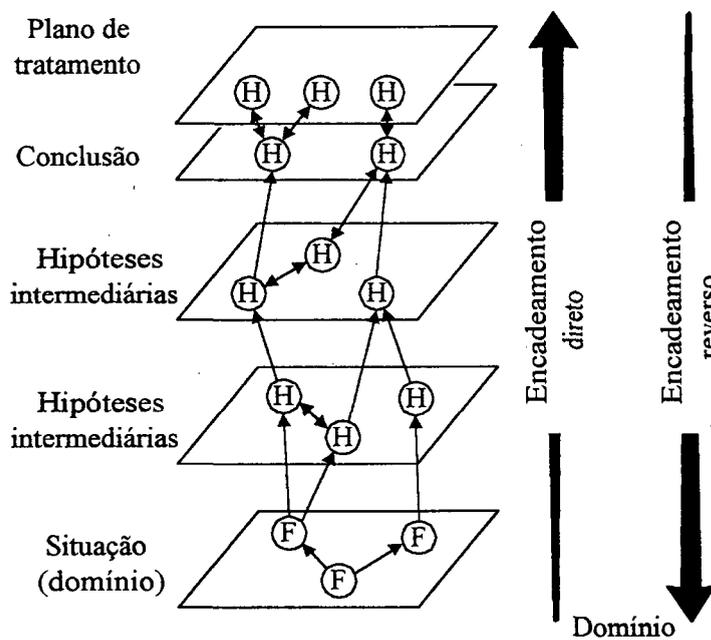


Figura 3.9 – Percursos dos encadeamentos *direto* e *reverso* (CHORAFAS, 1990).

O que diferencia os tipos de encadeamentos entre *regras* é a direção do fluxo de informações entre o início da busca até a chegada à solução, que em cada caso inicia-se em extremidades opostas.

Uma das características dos SE é a possibilidade de oferecer explicações sobre o motivo de chegar a uma determinada conclusão, que para os SE que utilizam o encadeamento *reverso* é mais comum do que os SE que utilizam encadeamento *direto* (GONZALEZ; DANKEL, 1993).

Dentro da hierarquia das *regras*, existe a possibilidade de distinção entre as *regras*, que podem ser compartimentadas conforme áreas do DC que deseja-se pesquisar na BC. A formação de *meta-regras* auxilia o desenvolvimento de SE, pois pode-se priorizar o processo de busca por solução em áreas específicas da BC (GONZALEZ; DANKEL, 1993).

Porém, na utilização de SE as *regras* possuem desvantagens na sua utilização, referente a fatores como (GONZALEZ; DANKEL, 1993):

- Ineficiência, pois quando um novo *fato* é gerado ou excluído, todas as *regras* devem ser analisadas novamente pela *máquina de inferência*, para verificar quais delas podem ser acionadas, o que em BC grandes pode consumir muito tempo de processamento. Porém, a tecnologia dos algoritmos *Rete*¹ tem simplificado e otimizado o mecanismo de busca sobre o encadeamento entre *regras*;
- Opacidade, pois é muito complexo examinar o encadeamento e a seqüência com que as ações são tomadas;
- Cobertura do DC, que dependendo da complexidade do domínio, pode requerer até milhares de *regras* o que torna muito difícil o seu desenvolvimento e manutenção.

3.1.8 – Busca por solução

A diferença fundamental entre os programas computacionais convencionais e os SE é a seqüência com que as *regras* são executadas, que depende da prioridade de execução existente na *agenda*. Esta classificação de prioridades, que soluciona o *conflito* entre *regras*, pode ser baseada segundo diversas técnicas, conforme a exigência e a forma com que o EC define o encadeamento das *regras*, gerando a *rede de inferência* que conduz o SE, em conjunto com os dados de entrada do usuário, à solução do problema.

Para melhor visão da abrangência da BC, a *rede de inferência* pode ser ilustrada utilizando o formato gráfico como apresentado na Figura 3.10, que possui os *nós* representando as *regras*, e as *conexões* entre estes *nós* indicando a relação entre os mesmos.

¹ O algoritmo *Rete* realiza a otimização do mecanismo de comparação entre as *regras* e os *fatos*, considerando no processo de busca por *regras* satisfeitas, apenas os *fatos* que foram modificados através de inclusão ou exclusão, ao invés de analisar todos os *fatos*, inclusive os inalterados, em relação às *regras* (GONZALEZ; DANKEL, 1993).

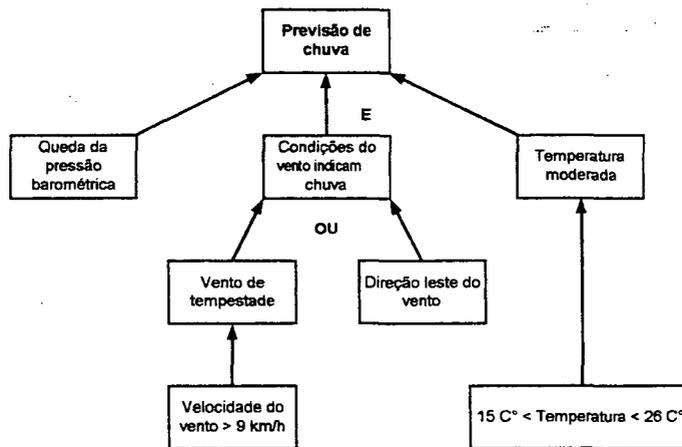


Figura 3.10 – Exemplo de *rede de inferência de regras* (DURKIN, 1994).

A técnica de resolução dos *conflitos* define o caminho a ser percorrido na *rede de inferência*, que pode seguir o caminho de busca utilizando (GONZALEZ; DANKEL, 1993, DURKIN, 1994):

- Busca em profundidade, que consiste em seguir um caminho na *rede de inferência* até atingir o nível de profundidade que propicie a chegada à solução do problema, em um caminho exaustivo de busca, conforme Figura 3.11;
- Busca em amplitude, que verifica simultaneamente todas as possibilidades de encadeamentos possíveis em um mesmo nível da *rede de inferência*, que após buscas exaustivas chega-se à solução, conforme Figura 3.12;
- Busca da melhor opção, que combina características da busca em profundidade e em amplitude que durante o procedimento de busca por solução, com a chegada a cada nó da *rede de inferência* existe a possibilidade de optar pelo rastreamento de outro nó que estejam no mesmo nível de especialização, ou prosseguir com a busca por outros nós que tenham maior profundidade. O direcionamento do mecanismo de busca por solução é feito conforme o julgamento do EC, fundamentado no conhecimento do EH.

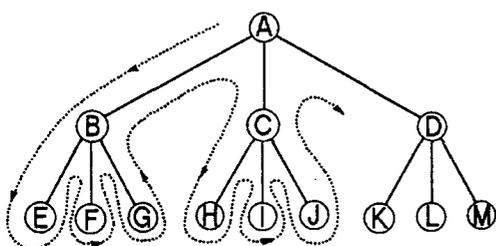


Figura 3.11 – Busca em profundidade.

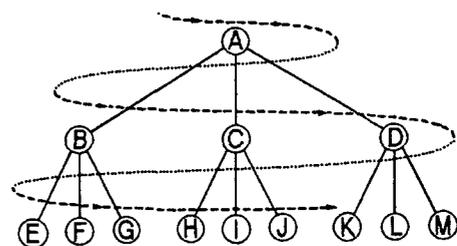


Figura 3.12 – Busca em amplitude.

Dependendo dos dados do problema e da BC do SE, a solução para os problemas pode estar localizada em qualquer um dos nós existentes na *rede de inferência* da BC. Caso a solução esteja nos nós iniciais da *rede de inferência*, é considerado que a solução é superficial. Sendo a solução localizada nos nós mais distantes a partir do início da *rede de inferência*, é considerado que a solução é profunda. A utilização de cada uma destas técnicas de busca por solução oferece vantagens e desvantagens, como apresentado na Tabela 3.5.

Tabela 3.5 – Características das técnicas de busca (DURKIN, 1994).

Busca em profundidade	
Vantagens	Desvantagens
Certeza na obtenção da solução, caso ela exista	Uniformidade, pois a escolha inicial por um caminho incorreto é constatado apenas no fim da busca
Velocidade na obtenção de uma das possíveis soluções	Inapropriado para domínios com muitas ramificações na <i>rede de inferência</i>
Concentração de esforços em apenas uma direção	Ineficiente na busca por soluções que são superficiais na <i>rede de inferência</i>
Busca em amplitude	
Vantagens	Desvantagens
Certeza na obtenção da solução, caso ela exista	Uniformidade, pois são percorridos todos os caminhos até chegar no fim da busca
Eficiência na busca por soluções que estejam localizadas nos níveis mais altos da <i>rede de inferência</i>	Inapropriado para domínios com muitas ramificações na <i>rede de inferência</i>
Concentração de esforços em apenas uma direção	Ineficiente na busca por soluções distantes do início da <i>rede de inferência</i> , sendo elas consideradas profundas
	Interação com o usuário deficiente, pois questões inapropriadas podem ser feitas ao usuário
Busca pela melhor opção	
Vantagens	Desvantagens
Utiliza o conhecimento para direcionar a busca pela solução	Não garante a chegada na solução do problema
Representa com melhor eficiência o modelo humano de inferência	

A aplicação da técnica de busca baseada na melhor opção deve ser optada apenas em circunstâncias em que o EH garante a chegada à solução. Com isso, constata-se a importância dos EH, que com intensa interação com o EC, traçam o caminho para chegar à solução pela melhor opção. Em circunstâncias em que o problema possui mais de uma solução, e sendo desejado listá-los na integralidade, todos os nós da *rede de inferência* devem ser percorridos.

Deste modo, o mecanismo de busca é indiferente na performance do *software* em SE, pois todos os nós são percorridos, sendo a diferença na adoção de uma determinada solução refletida apenas na seqüência com que estas são constatadas e propostas.

3.1.9 – Sistemas *shell*

Historicamente, o início das pesquisas em SE ocorre com o “[...] programa Dendral, que foi o primeiro programa de IA que priorizava a importância do conhecimento especializado através de métodos generalizados de solução de problemas” (BUCHANAN; SHORTLIFFE, 1984, p. 8). Estes métodos generalizados para solução de problemas formam a *máquina de inferência* (WATERMAN, 1986).

A finalidade do SE Dendral é representar o conhecimento especializado em química orgânica, a partir dos conhecimentos empíricos empregados na análise de dados analíticos provenientes de espectrômetros de massa (BUCHANAN; SHORTLIFFE, 1984).

O sucesso obtido no projeto Dendral impulsionou a exploração de outros campos de aplicação dos SE em DC específicos. Na seqüência, a técnica de SE foi empregada no setor médico, originando o *Mycin*, programa que destina-se ao diagnóstico e tratamento de doenças infecciosas. Neste projeto a programação concentrou-se tanto na formação da BC como da *máquina de inferência*.

A separação entre o mecanismo de solução de problemas em relação à BC é uma das características que tornam algumas aplicações de IA como não classificadas na categoria de SE, pelo fato do conhecimento estar distribuído ao longo de todo o código do programa (WATERMAN, 1986, GONZALEZ; DANKEL, 1993). Sendo o *Mycin* um *software* que se enquadra na categoria de SE por possuir a distinção entre a BC e a *máquina de inferência*, tornou possível a exploração da possibilidade de retirada do conhecimento médico especializado do *Mycin*, restando apenas o código responsável pela inferência, originando assim o *Empty-Mycin* ou simplesmente *EMycin*.

O *EMycin* é um programa computacional que possui apenas o mecanismo para a solução de problemas. O conhecimento necessário para a solução de problemas específicos pode ser inserido posteriormente, conforme o DC a ser explorado. Surge assim o primeiro programa computacional para desenvolver SE, criando a categoria de *software shell*, que na sua concepção disponibilizam os mecanismos para a solução de problemas segundo a técnica de SE, sendo o conhecimento específico em um determinado DC inserido posteriormente.

Porém, mesmo com a existência do *shell*, nada impede a programação por parte da equipe de projeto do SE em desenvolver todas as partes do programa que não fazem parte da

BC, como *máquina de inferência*, *agenda*, interfaces com usuário, entre outros. Durkin (1994) na Figura 3.13 indica os caminhos possíveis a serem percorridos na construção dos SE, referente as ferramentas computacionais a serem utilizadas.

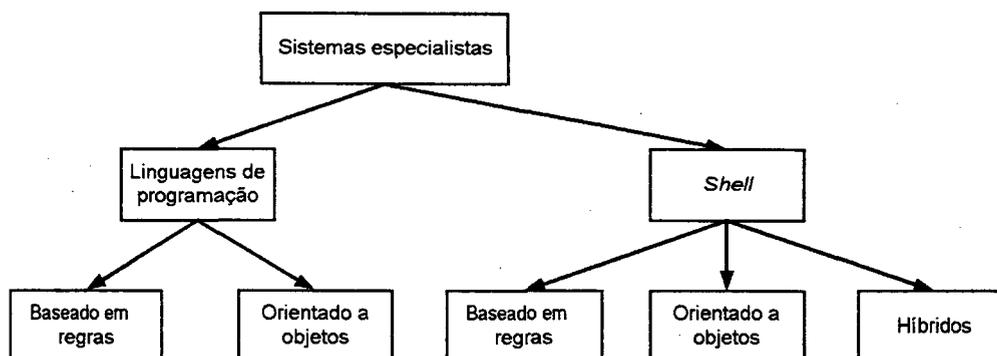


Figura 3.13 – Categorias de *software* para SE (DURKIN, 1994).

Tanto Waterman (1986) como Gonzalez e Dankel (1993) abordam algumas características referentes ao desenvolvimento de um sistema *shell* ou simplesmente utilizar um *shell* comercialmente disponível:

- Utilização de sistemas *shell*;
 - Requer menor esforço e tempo na construção do SE durante as etapas meramente computacionais;
 - Grande diversidade de *shell* já criados, que podem adequar-se a uma ampla variedade de aplicações;
- Programação do *shell*;
 - Maior flexibilidade técnica com relação à representação, organização e acesso do conhecimento;
 - Economia na compra do *shell*;
 - Possibilidade de solução de possíveis *bugs*;
 - Personalização do programa às necessidades do projeto.

A opção entre utilizar ou criar um *shell* depende dos requisitos iniciais estabelecidos no projeto do SE. Sendo feita a opção pela utilização de um sistema *shell*, resta identificar dentre a grande variedade de *shell* comercialmente disponíveis qual deve ser utilizado. Como esta é uma das tomadas de decisão iniciais no projeto e que tem efeito nas demais fases de desenvolvimento do SE, segundo uma abordagem de *engenharia simultânea* (ES) os requisitos de projeto devem ser conhecidos desde as etapas iniciais do projeto. Gonzalez e

Dankel (1993) abordam vários critérios relevantes a serem considerados durante a escolha de um sistema *shell*, destacando-se:

- Representação do conhecimento: além de *regras* e *fatos*, os sistemas *shell* podem manipular outras formas de representação do conhecimento, que dependem dos requisitos do projeto inicialmente estabelecidos;
- Desempenho: referente aos diversos tamanhos de BC processadas;
- Custo: que muitas vezes não está diretamente correlacionado com o desempenho do *shell*;
- Flexibilidade do *shell*, relacionado aos seguintes critérios:
 - Existência prévia de funções ou rotinas que facilitem a formação da BC;
 - Possibilidade de criar funções ou rotinas definidas pelo programador da BC no *shell*;
 - Interação do *shell* com outros programas computacionais;
- Suporte do fornecedor, relacionado aos seguintes critérios:
 - Documentação, contendo material completo de auxílio, como exemplos e tutoriais;
 - Ajuda *on-line*, fornecida na própria tela do computador;
 - Canais de comunicação, como telefone, *fax*, correios e correio eletrônico;
 - Treinamento e consultoria;
- Funções auxiliares como:
 - Editor da BC;
 - Telas de listagem dos itens em processamento, como variáveis, *fatos*, *regras na agenda* e as suas prioridades de execução, entre outros;
 - Auxílio no teste e verificação da BC enquanto está em desenvolvimento;
 - Ambiente gráfico para construção e disponibilização da BC, que é frequentemente exigido na entrada dos dados do usuário ou na explanação dos resultados de saída.

3.2 – Metodologia de projeto de SE

Assim como todo tipo de projeto de origens diversas que se beneficiam das abordagens metodológicas, os SE também possuem pesquisas referentes a sua metodologia de desenvolvimento.

As etapas do ciclo de vida no desenvolvimento de programas computacionais, segundo

Gonzalez e Dankel (1993) podem ser seguidas conforme a Figura 3.14, utilizando o modelo *waterfall*¹ ou então o modelo incremental.

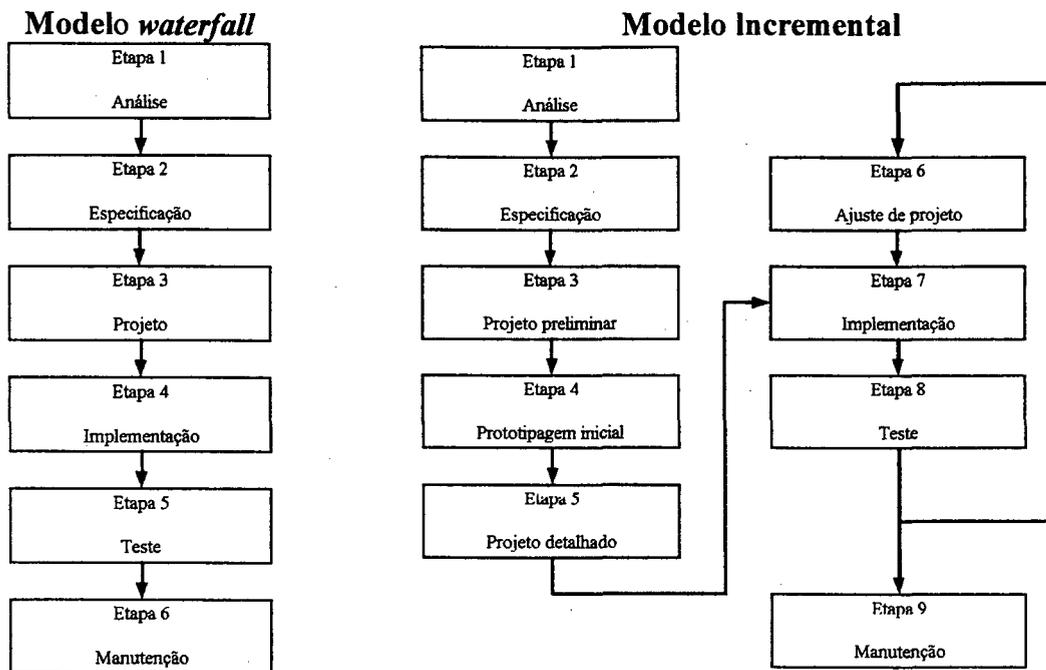


Figura 3.14 – Etapas de elaboração de *software* (GONZALEZ; DANKEL, 1993).

Uma das diferenças entre o desenvolvimento de um programa computacional convencional e de um programa de SE está na origem e quantidade de conhecimento a ser pesquisado, que para os SE dificilmente é totalmente conhecido mesmo para os EH, o que dificulta a determinação do esforço total a ser despendido (GONZALEZ; DANKEL, 1993).

Esta particularidade dos SE deve ser considerada na escolha do modelo de desenvolvimento deste, que conforme apresentado na Figura 3.14, tem-se as seguintes características em relação ao conhecimento inserido na BC:

- Modelo *waterfall*: é muito utilizado em programas computacionais convencionais, porém é inadequado para SE, pois muitas vezes a abrangência sobre o DC é muito grande, o que torna muito complexo e rígido este modelo, não permitindo realimentações e mudanças de paradigma ao longo do ciclo de desenvolvimento do SE;

¹ Modelo de desenvolvimento de programas computacionais, também conhecido como cascata, onde todas as etapas são percorridas de forma que, para ocorrer o prosseguimento para uma etapa posterior, a etapa atual deve estar completamente concluída.

- **Incremental:** que é um refinamento do modelo *waterfall*, que utiliza o modelo de inserção do conhecimento na BC em partes, o que torna o desenvolvimento do SE mais interativo, com possíveis ciclos de realimentação com informações advindas tanto por parte do EH como dos usuários, sendo portanto flexível permitindo mudanças de paradigma nas etapas do ciclo de desenvolvimento, conforme seja exigido.

Com a preferência pela adoção do modelo incremental é possível que as etapas do ciclo de desenvolvimento dos SE sejam seguidas utilizando apenas pequenas partes de conhecimento em relação à totalidade do DC, permitindo retornos às etapas anteriores, caso seja constatado algum tipo de erro ou inadequação em alguma tomada de decisão sobre o projeto do SE, seguindo assim os conceitos de ES na sua construção.

O conhecimento é, portanto, dividido em pequenas partes, que em conjunto formam a BC, seguindo as etapas descritas no modelo incremental da Figura 3.14. Durante a formação da BC, apesar desta não estar ainda concluída, pode-se obter uma funcionalidade parcial com algumas limitações, mas que demonstra a possibilidade de crescimento do SE; ao contrário, os programas computacionais convencionais precisam estar totalmente concluídos para poderem ser utilizados e testados (GONZALEZ; DANKEL, 1993). Em aplicações de SE, Chorafas (1990) afirma que com o SE 30% concluído já é possível testá-lo.

Dentro da seqüência metodológica de desenvolvimento dos SE, pelo modelo incremental tem-se as seguintes características das etapas de evolução:

- **Análise:** nesta etapa inicialmente tem-se a definição do domínio do problema, em conjunto com o estudo de adequação da técnica de SE para este domínio de problema, que pode ser auxiliado pelas questões da Tabela 3.4
- **Especificação:** que define as fronteiras do campo de aplicação e identificação de funcionalidades desejadas no SE;
- **Projeto preliminar:** que define as formas com que o conhecimento é inserido na BC, escolhas das ferramentas computacionais e do EH ou fontes de conhecimento alternativas a serem consultada;
- **Prototipagem rápida:** que é a construção de um SE, com limitações de robustez e abrangência de atuação, porém com possibilidade de obter conclusões sobre a adequação das decisões tomadas na etapa de projeto preliminar;
- **Projeto detalhado:** que faz a readequação das decisões tomadas na etapa de projeto preliminar fundamentado nos resultados da prototipagem rápida;

- **Implementação:** inicia-se nesta etapa o ciclo de desenvolvimento incremental. Com as novas definições e paradigmas estabelecidos na representação do conhecimento, prossegue-se com a programação definitiva de parte do conhecimento adquirido;
- **Teste:** que objetiva ter o retorno do desempenho do SE, que pode ser subdividida nas etapas de *verificação* que é realizada pelo EC, e na etapa de *validação* que é conduzida pelo EH e usuários do sistema;
- **Ajustes de projeto:** que visa realizar pequenos ajustes a partir do retorno das conclusões da etapa de teste. Após esta etapa, inicia-se um novo ciclo de implementação e teste, que é conduzido para cada parte de conhecimento a ser inserido no SE;
- **Manutenção:** após ser atingida as metas de abrangência da BC, chegando a finalização do SE, existe a etapa de manutenção que é realizada para correções de falhas não identificadas durante a construção do SE, atualizações da BC para as novas condições do DC ou expansão da atuação da BC em relação ao DC;

Conforme as etapas de desenvolvimento incremental são percorridas, a BC atinge a maturidade e a capacidade de resolução de problemas cada vez mais complexos e com grande amplitude de atuação. Dentro das etapas de construção dos SE, pode-se classificá-los conforme a sua atuação em (RILEY; GIARRATANO, 1994, WATERMAN, 1986):

- **Protótipo rápido:** que serve para demonstrar idéias, despertar interesse e identificar funcionalidades. Conforme a seqüência incremental da Figura 3.14, a prototipagem rápida é a primeira fase de implementação de qualquer SE;
- **Sistema refinado (teste alfa):** no qual o SE é submetido à teste pelo EC e o EH;
- **Teste de campo (teste beta):** sendo o SE submetido à teste com usuários selecionados;
- **Sistema comercial:** que exige treinamento, suporte e documentação do SE;
- **Manutenção e avaliação:** destinada à correção de *bug*¹, ampliação da capacidade ou atualizações da BC. Chorafas (1990) relata que a probabilidade de serem necessárias mudanças na BC para o primeiro ano é de 50%, para o segundo ano mais de 80% e com três anos estas mudanças na BC são inevitáveis.

A passagem dos estágios iniciais para os estágios finais no ciclo de desenvolvimento

¹ *Bug*: erros de sintaxe ou semântica durante a implementação do programa computacional (GONZALEZ; DANKEL, 1993)

dos SE caracteriza uma maior abrangência sobre o domínio do problema em conjunto com o aumento da robustez, que possibilita a utilização do SE em condições reais adversas. O desenvolvimento dos SE inicia-se realmente após o término das etapas iniciais compreendidas entre a *análise* e o *projeto detalhado* que definem e justificam a sua aplicação, especificando os requisitos do programa computacional e projetando o SE (GONZALEZ; DANKEL, 1993). Dentro da seqüência de etapas percorridas durante a evolução e desenvolvimento dos SE, independente do modelo adotado, tem-se obrigatoriamente a passagem pelas fases de:

- Aquisição do conhecimento;
- Representação do conhecimento;
- Implementação do conhecimento;
- Teste do SE;

, que encontram-se distribuídas ao longo de todo o ciclo de desenvolvimento dos SE.

A opção por um determinado modelo de desenvolvimento tem influência apenas na seqüência com que estas etapas são percorridas. Técnicas e ferramentas de auxílio nestas etapas são discutidas com maior profundidade nas próximas seções.

3.2.1– Aquisição do conhecimento

A etapa de aquisição do conhecimento é responsável pela coleta do conhecimento especializado para a formação da BC. Nesta etapa tem-se a ação do EC na coleta do conhecimento em fontes diversas, sendo a mais comum através de EH, que é aplicado principalmente em circunstâncias em que o conhecimento encontra-se disponível apenas com esta pessoa.

A técnica mais utilizada para a aquisição do conhecimento entre o EC e o EH é através de entrevistas, que pode ser um processo que necessite de semanas ou até mesmo alguns meses. A sistemática das entrevistas segue a relação entre o EC e o EH estabelecida na Figura 3.1. Como é necessário dedicar muito tempo para realizar esta atividade, este processo torna-se muito cansativo e tedioso para o EH, que na grande maioria das situações é uma pessoa extremamente ocupada e com tempo limitado.

Mesmo para um EH e um EC ambos experimentados, a passagem do conhecimento para a formação da BC pode ser dificultada pelas limitações pessoais de ambos. As maiores dificuldades na aquisição do conhecimento são (DURKIN, 1994):

- O EH pode fornecer conhecimento irrelevante, incompleto, incorreto ou inconsistente;

- O EH pode não ser capaz de verbalizar o conhecimento ou não ter noção do conhecimento que ele utiliza;

Na primeira situação, o problema está com o não comprometimento do EH com o projeto, sendo portanto preciso despertar o interesse do mesmo, demonstrando a importância do EH para o sucesso do projeto, apresentação dos objetivos do trabalho, os benefícios que o EH pode ter ao auxiliar o projeto ou, em último caso, havendo uma estrutura hierárquica, buscar o comprometimento com o projeto a partir dos superiores hierárquicos do EH. Na segunda situação, o EH tem o domínio sobre uma determinada especialidade, porém não consegue repassá-lo ao EC. Isto deve-se ao fato do conhecimento estar compilado, o que ocorre com maior frequência em situações em que o conhecimento possui origem em experiências profissionais, sendo preciso a descompilação para a sua aquisição (DURKIN, 1994).

Uma das formas de solucionar esta barreira de comunicação é através da adoção de técnicas de *observação*, em que o EC constata os procedimentos que o EH segue para buscar a solução de um problema e tenta entendê-lo e reproduzi-lo. A técnica de *observação* pode ser conduzida das seguintes formas (GONZALEZ; DANKEL, 1993):

- Observação em silêncio: EC verifica as etapas e procedimentos que o EH utiliza no seu cotidiano de trabalho. Esta técnica tem como desvantagem a pouca interação entre o EH e o EC, sendo recomendada apenas para as etapas iniciais de aquisição do conhecimento, para que o EC tenha o primeiro contato com o DC, conhecendo as nomenclaturas adotadas e observe a magnitude do problema;
- Observação com discussão: é um refinamento do método anterior, com maior interação entre o EC e o EH, através de indagações ao longo do processo de busca de solução pelo EH;
- Solução de problemas já resolvidos: em que o EC submete o EH à problemas anteriormente constatados e solucionados. Com isso, busca-se coletar a técnica para solução do problema.

Dentre estas formas de aquisição do conhecimento, Dym e Levitt (1991) dão preferência em permitir que o EH demonstre a sua especialidade em problemas exemplos, obtidos de experiências anteriores, ao invés do EH simplesmente descrever o que sabe.

Contudo, em circunstâncias em que o EC possui um significativo nível de entendimento sobre o DC, pode ser adotada a técnica *intuitiva*. Nesta condição o EC atua como um pseudo EH, utilizando técnicas intuitivas para coletar o conhecimento, através de estudos e

abordagens seguidas pelos EH (GONZALEZ; DANKEL, 1993):

Nesta circunstância também pode ser utilizado como fonte de conhecimento materiais documentados em livros e manuais (GONZALEZ; DANKEL, 1993). Podem também ser consultadas normas, artigos, guias, esquemas, inspeções visuais ou até mesmo o usuário final como fonte de conhecimento, pois a princípio deve-se possuir um certo nível de conhecimento mínimo que possibilite responder as perguntas de entrada e interpretar as respostas das saídas de um SE, sendo neste caso consultado as funcionalidades que o SE deve atender (DURKIN, 1994, GONZALEZ; DANKEL, 1993).

3.2.2 – Representação do conhecimento

Após a obtenção do conhecimento, prossegue-se a etapa de organização do conhecimento que relaciona as *entradas* e as *saídas*, identificando a porção *intermediária* do conhecimento que conecta estas duas partes. Esta etapa deve ser conduzida pelo EC, que deve projetar a forma mais adequada de transcrever o conhecimento dos EH para uma linguagem que possa ser repassada para o computador, e que tenha a máxima eficiência. Esta etapa é conhecida com representação do conhecimento.

A codificação do conhecimento é preciso, pois o EH não processa e manipula o seu conhecimento utilizando *regras*, *fatos* ou outro formato qualquer, sendo preciso a transcrição para a sintaxe que um programa computacional utiliza (DYM; LEVITT, 1991). Por isso, as técnicas de representação do conhecimento buscam utilizar formatos que se assemelhem à forma de pensamento dos EH, como o caso das *regras* e *fatos*, que simplificam a tarefa de codificação do EC.

A escolha da forma com que este conhecimento é manipulado pela *máquina de inferência* depende e está limitado às possibilidades de representação do conhecimento permissíveis pela ferramenta computacional adotada. Portanto, tanto a seleção da técnica de representação do conhecimento quanto à escolha da ferramenta computacional a ser utilizada devem ser feitas de modo integrado.

Dentre as formas de representação do conhecimento, desde os mais simples aos mais complexos SE utilizam-se das *regras* e *fatos*, que é o formato da essência dos SE. Cada *regra* pode ser considerada como sendo parte do conhecimento obtido do EH e inserido no SE, que em conjunto formam a BC.

Complementando as *regras*, são aplicadas técnicas híbridas na formação da BC. Além dos tradicionais itens processados pelas *regras* como variáveis e *fatos*, os SE híbridos permitem a manipulação do conhecimento estruturado de formas diversas. Uma das

aplicações dos SE híbridos é na representação *orientada a objeto* (OO), tanto na formação da BC quanto na inserção dos dados do problema do usuário a ser solucionado pelo SE.

A OO é aplicada em circunstâncias em que deseja-se descrever e modelar o mundo real para ser inserido em um programa computacional, considerando critérios que são relevantes em uma determinada perspectiva de análise.

Dentro da OO são empregados os termos *classe* e *objeto*. Na definição de Coad e Yourdon (1997, p. 50) as *classes* são “uma descrição de um ou mais Objetos com um conjunto uniforme de Atributos e Serviços, incluindo uma descrição de como criar novos Objetos na Classe”. Portanto, as *classes* podem ser consideradas como *matrizes* na modelagem do mundo real.

Coad e Yourdon (1997, p. 50, grifo do autor) também definem os *objetos* como “uma *abstração* de alguma coisa em um domínio de problema, exprimindo as capacidades de um sistema de manter informações sobre ela, interagir com ela, ou ambos, em um *encapsulamento* de valores de Atributos e seus Serviços exclusivos”. Os *objetos* são portanto, os elementos do mundo real que são formados segundo a *matriz* de uma determinada *classe*.

Surgem nestas definições novos termos que são:

- *Encapsulamento*: “princípio, usado no desenvolvimento de uma estrutura global de programa, de que cada componente do programa deve conter uma única decisão de projeto [...] a interface para cada módulo é definida de forma a revelar o mínimo possível sobre seu funcionamento interno” (OXFORD, 1986 apud COAD; YOURDON, 1997, p. 13). O *encapsulamento* busca identificar e agrupar partes do programa que possuem alta e baixa volatilidade podendo ser submetido a muitas ou poucas mudanças no decorrer do seu ciclo de vida, tornando a manutenção e a reutilização de código mais simples. Na OO a unidade básica do encapsulamento é a *classe* (MADEIRA, 1995);
- *Abstração*: “ignorar aspectos de algumas entidades que não são relevantes para o problema atual que pode-se concentrar plenamente nestes aspectos” (GONZALEZ; DANKEL, 1993, p. 198). A *abstração* pode ser considerada a modelagem simplificada de um determinado item do mundo real.
- *Atributos*: “[...] é um dado (informação de estado) para o qual cada Objeto em uma Classe tem seu próprio valor” (COAD; YOURDON, 1997, p. 117). Os atributos são justamente os itens importantes considerados na *abstração* da modelagem e que possuem as informações que caracterizam o *objeto*;

- *Serviços*: “[...] é um comportamento específico que um Objeto deve exibir [...] sendo a comunicação necessária entre Objetos” (COAD; YOURDON, 1997, p. 141-142). Os *serviços* são os procedimentos aos quais os *objetos* são submetidos. Os *serviços* são a única forma de os elementos modelados, que neste caso são os *objetos*, poderem comunicar-se entre si (WATERMAN, 1986).

Na modelagem OO existe uma inter-relação entre as *classes* que pode ser tanto do tipo *todo-parte*¹ como do tipo *generalização-especialização*². Na notação de Coad e Yourdon (1997) as *classes* e a inter-relação entre elas pode ser representada pela Figura 3.15. No caso da dependência *todo-parte*, as *classes parte 1* e *parte 2* em conjunto formam a *classe* de origem *todo*. Para a dependência *generalização-especialização*, as *especializações 1* e *2* são casos particulares da *classe* *generalização*.

Outra funcionalidade que a OO propicia é a *herança*, que possibilita às *classes* herdarem *atributos* e *serviço* de outras *classes* situadas hierarquicamente acima. Na estrutura de classes da Figura 3.15, as *classes parte 1* e *2* bem como *especialização 1* e *2* herdam os *atributos* e *serviços* das *classes* *todo* e *generalização*, respectivamente.

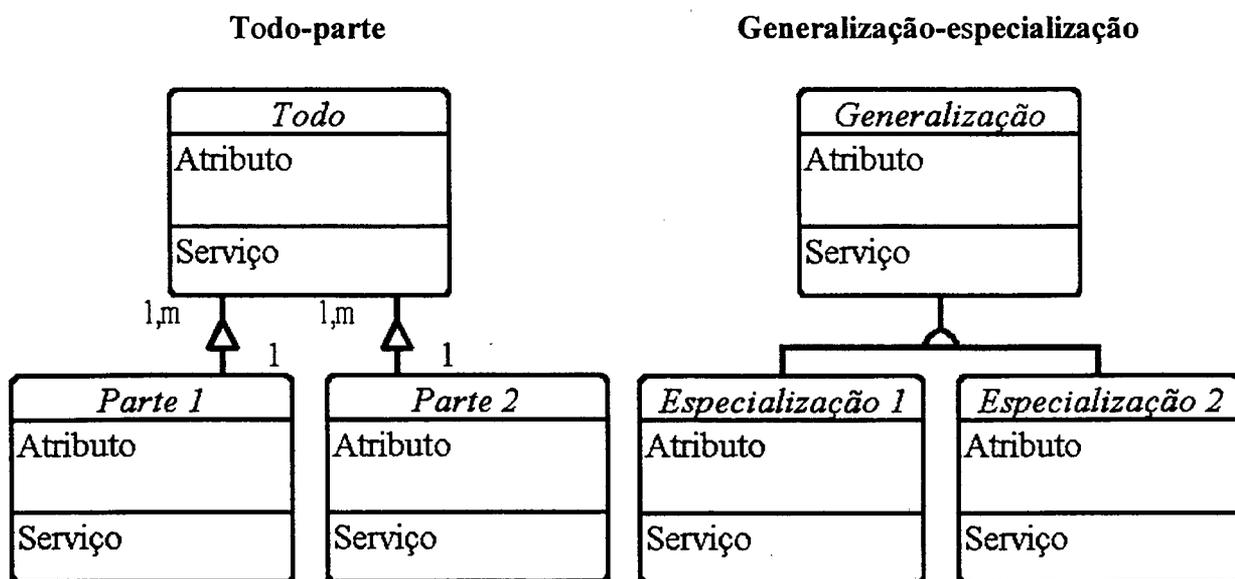


Figura 3.15 – *Todo-parte* e *generalização-especialização* (COAD; YOURDON, 1997).

¹ *Todo-parte*: estabelece uma relação do tipo *tem um* entre as *classes* (COAD, YOURDON, 1997).
² *Generalização-especialização*: pode ser interpretada como uma relação *é um* ou então *é um tipo de* entre as *classes* (COAD, YOURDON, 1997).

A flexibilidade da OO propicia também a reutilização de código, reduzindo o tempo e esforço na representação do conhecimento, aumentando a modularidade, possibilitando que mesmo partes projetadas separadamente possam posteriormente atuarem em conjunto (GONZALEZ; DANKEL, 1993).

3.2.3 – Implementação do conhecimento

Prosseguindo o processo de desenvolvimento dos SE, segue-se a etapa de implementação. Nela a representação do conhecimento é transferida para a linguagem de programação do SE. Apesar de ser a etapa que mais consome tempo dentre todas as demais, pouco pode ser acrescido sobre ela, sendo essencialmente uma tarefa de programação.

3.2.4 – Teste do SE

Na continuidade, tem-se as fases de *verificação e validação* (VV), que em conjunto formam a etapa de teste, fechando assim o ciclo incremental de desenvolvimento dos SE, podendo ser reiniciado um novo ciclo para aprimorar e potencializar a BC. Detalhes sobre a VV são descritos com maior profundidade no Capítulo 6, com a aplicação práticas das técnicas de VV.

Dentro das exposições apresentadas neste capítulo, previamente pode-se definir que o trabalho de desenvolvimento do SE sobre o DC da pneumática deve estender-se sobre o conhecimento heurístico utilizado nestes tipos de projetos. Com antecipação pode-se também definir que será utilizado o modelo incremental de desenvolvimento do protótipo, pela flexibilidade e a grande interação possível de obter-se no desenvolvimento de SE.

Tópicos especificamente relacionados à adequação da técnica de SE ao DC dos projetos pneumáticos, definição da equipe de desenvolvimento do SE, ferramentas computacionais a serem utilizadas, seleção da forma e estrutura da representação do conhecimento, encadeamento entre regras e forma de busca por soluções, são apresentados com detalhes no Capítulo 5.

No próximo capítulo são apresentados conceitos e informações a respeito do DC de projetos de sistemas pneumáticos, que fundamentam a possibilidade de desenvolver um SE neste domínio, que em conjunto com o conteúdo apresentado no capítulo atual, fornece subsídios para detalhar informações sobre o SE a ser apresentado no Capítulo 5.

Capítulo 4 – Projeto de sistemas pneumáticos

Neste capítulo é detalhada a importância das metodologias de projeto na elaboração de sistemas pneumáticos, bem como a listagem de tipos de aplicações para a pneumática, e a coleta de requisitos de projetos para modelagem de cargas mecânicas, além de técnicas usuais e normatizadas para demonstrá-las. Também são abordadas técnicas de projetos, segundo metodologias para o domínio pneumático e eletropneumático, bem como as atuais aplicações. Por fim, a esquematização de sistemas pneumáticos, segundo normas para pneumática.

4.1 – Aplicações da pneumática

Conforme destacado no Capítulo 1, a pneumática possui amplo campo de atuação, diversidade de opções e modularidade nas suas construções. Em aplicações de automação industrial, são diversas as possibilidades de utilização da pneumática.

Na grande maioria das aplicações da pneumática, o ar comprimido é utilizado como meio de transferência de energia, assim como nas aplicações hidráulicas. Para melhor classificar cada uma das aplicações da pneumática que exercem funções especializadas, que em conjunto propiciam a funcionalidade final desejada, pode-se propor a seguinte estrutura de divisão para os diferentes tipos de circuitos pneumáticos:

- Circuitos de produção de ar comprimido, que são constituídos de componentes responsáveis pelas etapas de compressão, resfriamento, filtragem, secagem e armazenagem do ar comprimido. Conforme detalhes como consumo de ar comprimido, quantidade e distância entre estes pontos de consumo, flutuações no consumo entre outros parâmetros, pode-se ter a instalação de unidades de compressão centralizadas ou descentralizadas (PRINCÍPIOS, 19--);
- Circuitos da rede de distribuição de ar comprimido, responsáveis pela condução do ar comprimido até os pontos de consumo. São utilizados quando se opta por circuitos de produção de ar comprimido centralizados (PRINCÍPIOS, 19--);
- Circuitos operados por ar comprimido, que são os locais onde o ar comprimido é empregado como fonte de energia mecânica.

A grande maioria das aplicações que utilizam a pneumática, empregam unidades de produção de ar comprimido centralizadas, pois se pode assim obter (PRINCÍPIOS, 19--):

- Menor custo inicial na construção da instalação;
- Maior eficiência no ciclo de compressão;
- Menor custo de supervisão e manutenção;
- Instalações otimizadas, pois muitas vezes os picos de consumo de ar comprimido dos diversos pontos de utilização não ocorrem simultaneamente.

Com isso, os circuitos de produção de ar comprimido podem ser considerados como unidades comuns aos pontos de consumo de ar comprimido, e o seu projeto deve satisfazer aos requisitos globais de qualidade exigido nas suas aplicações como: teor de umidade, impurezas, vazão, pressão e as oscilações máximas aceitáveis destes parâmetros.

Portanto, por simplicidade pode-se realizar o projeto dos sistemas que utilizam o ar comprimido como fonte de energia, sem que para isso seja preciso a representação da unidade produtora de ar comprimido, pois é considerado que o ar comprimido é oferecido nas condições ideais, de acordo com as necessidades da respectiva aplicação pela rede de distribuição.

O direcionamento deste trabalho é a apresentação dos diferentes tipos de circuitos que utilizam a energia existente no ar comprimido, iniciando com as definições sobre os *comandos*.

4.2 – Classificação dos circuitos de *comando* pneumático

A palavra *comando* é utilizada no cotidiano em contextos diversos. Porém, por definição segundo a norma DIN 19226, *comando* é “[...] um processo num *sistema*, mediante o qual uma ou mais grandezas de *entrada* influenciam uma ou mais grandezas de *saída*, de acordo com as *características próprias deste sistema*” (apud BOLLMANN, 1997, p. 21, grifo nosso). A Figura 4.1 ilustra na forma gráfica a representação genérica dos esquemas de *comando*.

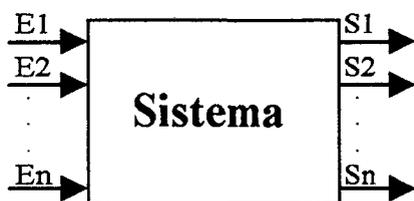


Figura 4.1 – Esquema de aplicações de comando (BOLLMANN, 1997).

Para as etapas iniciais de projetos, a utilização do modelo da Figura 4.1 auxilia no entendimento das necessidades a serem satisfeitas, pois é adotado a formatação de *caixa preta*, em que são considerados os pontos primordiais desejados na saída, que são influenciados pelas entradas. Em etapas iniciais do projeto este modelo é vantajoso, pois são considerados apenas dados referentes à funcionalidade a ser atendida, sem que para isto seja preciso identificar qual princípio de solução deve ser adotado.

Dentre as possibilidades de aplicações de *comando* na pneumática, os circuitos do tipo binário ou também conhecido como *on/off* possuem ampla aplicação, que vão desde os circuitos de *simples comando*¹ até os mais especializados circuitos de automação que metodologicamente podem ser classificados, de acordo com o tipo de operação que realizam, em *comando seqüencial* e *comando combinatório*, conforme ilustrado na Figura 4.2.

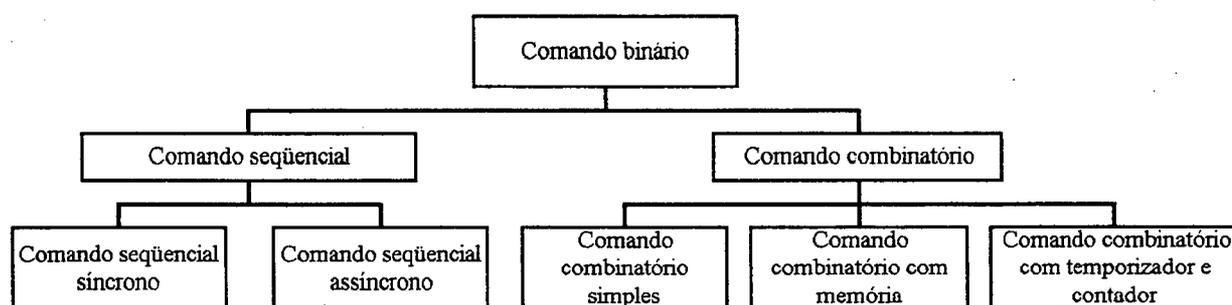


Figura 4.2 – Principais tipos de comandos binários (BOLLMANN, 1997).

Em processos de automação industrial, os *comandos seqüenciais* são de ampla aplicação, possibilitando a execução de tarefas de forma sucessiva, conforme procedimentos são realizados e concluídos.

Dentre eles, os *comandos seqüenciais assíncronos*, também conhecidos como comandos de *trajetória programada*, tem especial destaque devido à maior freqüência com que são aplicados (HASEBRINK; KLOBER, 1975). A sua utilização é destaque em situações que, por motivos de segurança ou inerentes ao processo automatizado, seja preciso a conclusão de uma etapa para dar prosseguimentos nas demais subseqüentes.

¹ Como circuitos de *simples comando* interpreta-se como sendo as aplicações em pneumática que utilizam apenas elementos de atuação e comando, os quais não se enquadram nos comandos mais elaborados e que já possuem nomenclatura como *seqüencial* e *combinatório*.

Com os *comandos seqüenciais*, têm-se a redução ou a eliminação do elemento humano no processo automatizado, que muitas vezes é requerido devido a exigências de qualidade no processo, referentes à higiene, repetibilidade e velocidade que afetam a produtividade, ou por operações em ambientes insalubres. Porém, atenção especial deve ser dedicada ao impacto social gerado na implantação de sistemas automatizados, que na grande maioria das vezes resulta na redução do número de postos de trabalho, sem a relocação desta mão de obra. Portanto, fatores sociais devem também ser alvo de estudo durante implantação de processos automatizados.

Com a interação da pneumática com outros domínios de energia, as aplicações da pneumática em projetos seqüenciais têm um novo destaque nas automações industriais, como em setores de empacotamento ou de transporte de material, através da programação da seqüência de movimentos a serem feitos (SCHNEIDER; HITCHCOX, 1998).

4.3 – Estrutura de sistemas pneumáticos

Geralmente, em projetos pneumáticos as unidades elementares da construção são os componentes. Com os componentes, pode-se obter uma grande modularidade no projeto, característica esta fundamental na reutilização de partes isoladas para a obtenção da funcionalidade final esperada.

Apesar dos componentes serem formados por várias partes, o projetista pode manipulá-los com um grau de abstração que, ao ser feita a opção por um certo componente, é subentendido a seleção por todas as peças com que este é formado, simplificando o trabalho do projetista, restando apenas informar como estes são interconectados.

Quando são requeridas soluções personalizadas, que devem satisfazer a necessidades específicas, o projeto de componentes é aplicável, desde que fatores como tempo e custo adicionalmente despendidos sejam requisitos de menor importância e recompensem o resultado final.

Para auxiliar o projetista em seu trabalho, devido à grande diversidade de componentes existentes, são utilizadas padronizações referentes à representação destes, através de simbologia gráfica normatizada simplificada. Etapas de representação de esquemas beneficiam-se destas simbologias, pois propiciam maior agilidade na descrição dos componentes, principalmente nas etapas conceituais do projeto, pois não é preciso um detalhamento minucioso das partes que formam os componentes, sendo estes tratados como módulos. A norma ISO 1219 parte 1 (ISO 1219-1, 1991) é dedicada justamente a simbologia dos componentes tanto pneumáticos como hidráulicos.

Como muitas vezes os componentes não podem satisfazer isoladamente uma determinada funcionalidade, estes são interconectados originando os circuitos, que são agrupamentos de componentes destinados a obter uma funcionalidade em comum, a qual pode ser um desmembramento da funcionalidade geral que se deseja obter com o sistema.

Conforme discutido no Capítulo 2, um sistema pode ser formado por apenas uma unidade elementar, até um agrupamento de elementos. O que define a abrangência de um sistema é o referencial ou *ponto de vista* que um observador tenha sobre o todo.

Devido à flexibilidade na determinação da abrangência das fronteiras de um sistema, pode-se considerar a hierarquia dos mesmos como sendo formado por todos os circuitos que devem ser gerados para satisfazer o conjunto de requisitos das cargas mecânicas informadas pelo usuário. Na modelagem de sistemas hidráulicos utilizada por Silva (1998), é seguida a estrutura da Figura 4.3.

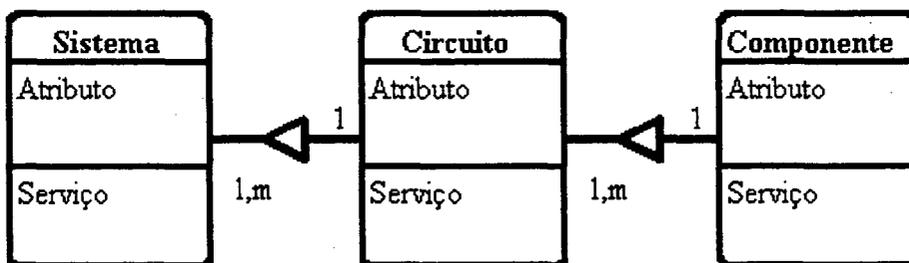


Figura 4.3 – Estrutura orientada a objeto para sistemas hidráulicos (SILVA, 1998).

Neste contexto, pode-se também modelar os sistemas pneumáticos conforme uma estrutura *todo-parte orientada a objeto* (OO). Portanto, um sistema pneumático pode ser formado por um ou mais circuitos. O mesmo ocorre com os circuitos pneumáticos que são formados por vários componentes, e em especial pode ser formado por apenas um componente, caso comum em aplicações de componentes personalizados.

4.4 – Elementos de sistemas pneumáticos

Conforme a hierarquia estabelecida na seção 4.3, as unidades elementares para os sistemas pneumáticos são os componentes, os quais podem ser funcionalmente agrupados de acordo com a função que realizam dentro do sistema.

Para o caso de sistemas que utilizam somente a pneumática como fonte de energia, estes agrupamentos podem ser caracterizados conforme esquema apresentado na Figura 4.4. De acordo com a complexidade do sistema, estes grupos podem ser formados por um ou mais

componentes; no caso de sistemas simples, pode-se ter um componente realizando diversas tarefas, pertencendo portanto a mais de um grupo de elemento.

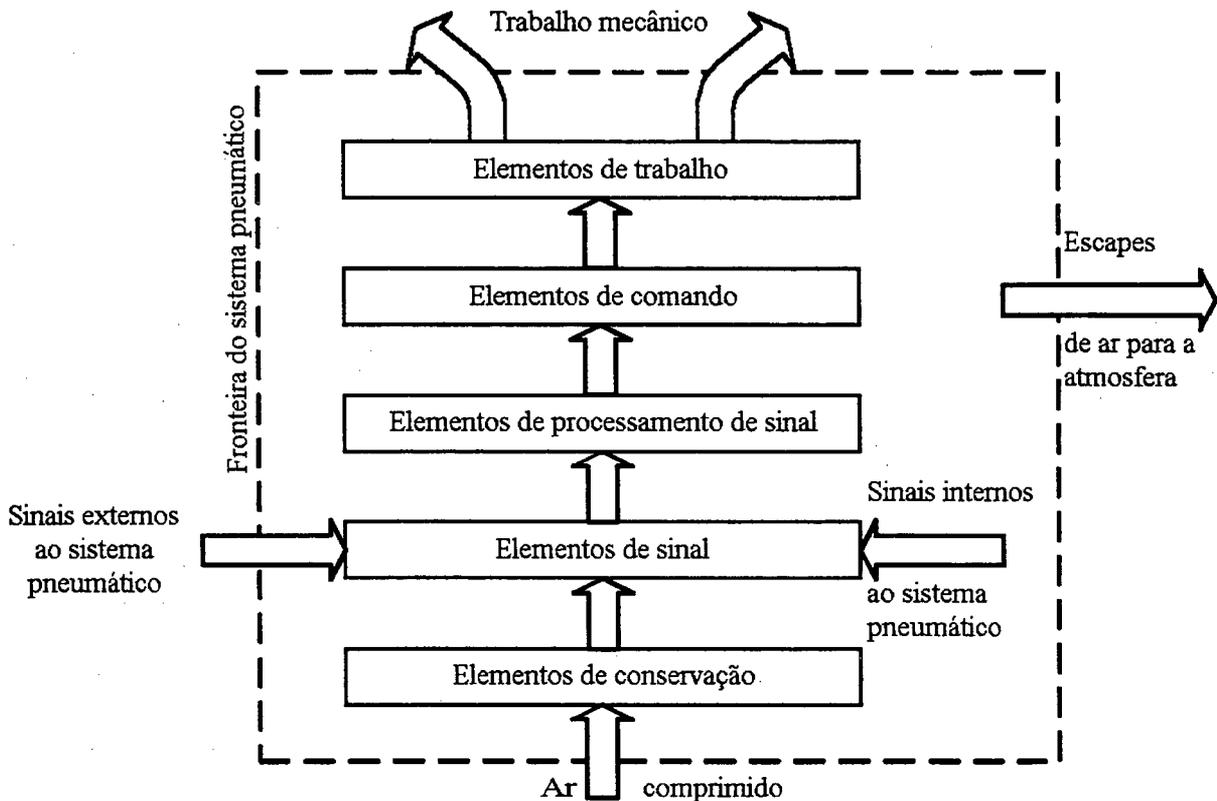


Figura 4.4 – Elementos de sistemas pneumáticos (HASEBRINK; KLOBER, 1988).

Portanto, esta classificação não considera a funcionalidade que deve-se satisfazer na construção dos circuitos que formam o sistema. Circuitos distintos entre si podem possuir componentes que pertençam a uma mesma classificação, bem como um único circuito pode possuir componentes que pertençam a diferentes classificações.

Domínios diversos de energia podem também ser inclusos em um sistema pneumático, para suprir a operação de um determinado princípio de solução, conforme os requisitos que devem ser satisfeitos pelo projeto. Dentre as combinações mais comuns da pneumática, tem-se a sua utilização em conjunto com a elétrica e a eletrônica que formam, respectivamente, a eletropneumática e a pneutrônica, e que possuem componentes preparados para operarem especificamente em cada um dos tipos de elementos de sistema apresentados na Figura 4.4. Na Tabela 4.1 são listados os componentes usualmente empregados tanto na pneumática, como na eletropneumática e na pneutrônica.

Tabela 4.1 – Componentes de sistemas pneumáticos (BOLLMANN, 1997).

	Comandos pneumáticos	Comandos eletropneumáticos	Comandos pneumáticos eletrônicos
Elementos de trabalho	Atuadores pneumáticos lineares e rotativos;		
Elementos de comando	Válvulas direcionais		
	Acionadas por pressão;	Acionadas por solenóide;	
Elementos de processamento de sinal	Válvulas pneumáticas 3/2 NA, 3/2 NF, lógica <i>Ou</i> , lógica <i>E</i> ;	Relés ¹ , chaves e interruptores NA e NF, ligados em série e em paralelo;	CLP
Elementos de sinal	Válvulas 3/2 NA, 3/2 NF e sensores pneumáticos (barreira de ar);	Botões, interruptores elétricos e sensores eletrônicos (óticos, magnéticos, indutivos, capacitivos, etc.);	

4.5– Requisitos de projeto em pneumática

Como apresentado no Capítulo 2, a etapa inicial de um projeto passa pela coleta das necessidades dos *clientes*, para posterior definição dos requisitos de projeto. Estes requisitos devem ser conhecidos por todos envolvidos com o projeto, para a otimização do produto ou serviço às necessidades dos clientes (CHAPMAN; BAHILL; WYMORE, 1992).

Devido à abrangência de atuação da pneumática e a sua interação com outros princípios de solução, conclui-se que o projeto de sistemas pneumáticos requer um conhecimento multidisciplinar entre domínios distintos, conforme discutido no Capítulo 2. Portanto, é preciso o esforço de diversos profissionais que possuam especializações em áreas distintas.

Como ocorre na grande maioria das aplicações de automação industrial, a pneumática destina-se a realizar trabalho mecânico, através da movimentação de cargas mecânicas, sendo de vital importância a perfeita caracterização das mesmas, pois a forma com que uma carga deve ser movimentada tem influência nas tomadas de decisão e na escolha dos princípios de solução adotados na forma de circuitos.

¹ Neste texto é adotada a ortografia relé, apesar de relê também ser utilizada, sendo ambas corretas.

Para tornar a coleta de requisitos metódica, sem falhas nesta etapa primordial, a modelagem de cargas mecânicas é de grande valia, através da simplificação das condições reais que devem ser consideradas como requisitos do projeto, eliminando informações que ou são irrelevantes para uma determinada etapa de projeto, ou até mesmo não tem influência nas tomadas de decisão, em detrimento das informações primordiais ao projeto. A *engenharia simultânea* (ES) pode auxiliar durante esta etapa de coleta dos requisitos, pois insere no meio do ambiente de projeto todos os potenciais *clientes* e as suas necessidades. Tópicos sobre modelagem de cargas mecânicas são discutidos na seção 4.5.1.

Sendo requerido um maior refinamento na coleta dos requisitos de projeto, através da consideração de parâmetros que não podem ser representados somente através da modelagem das cargas mecânicas, pode-se realizar a inclusão de novas restrições ao projeto, de acordo com as necessidades do mesmo. Descrições adicionais são tratadas na seção 4.5.2.

Conforme o tipo de projeto, requisitos de operação especiais podem ser indicados durante o interfaceamento do sistema pneumático proposto pelo projeto e o usuário. Na seção 4.5.3 são discutidos tópicos referentes a comandos especiais em projetos seqüenciais e requisitos para a sua elaboração.

4.5.1 – Modelagem de cargas mecânicas

Na etapa de determinação dos requisitos do projeto pneumático, a descrição das funções a serem realizadas é de extrema importância. A etapa de modelagem visa descrever, segundo requisitos coletados, a representação simplificada das funções que devem ser operacionalizadas.

De Negri (1996) define os modelos, resultados de uma modelagem, como sendo a explicitação de uma abstração. Para os modelos funcionais é descrita a capacidade de desempenhar uma ação ou um conjunto de ações.

Em um projeto de sistema pneumático, esta abstração da funcionalidade desejada possui a profundidade conforme o grau de detalhamento exigido na representação e construção do modelo. Os modelos podem sofrer simplificações, ou quando necessário, novas inclusões de detalhes podem ser feitas, conforme a exigência por maiores detalhamentos em relação aos critérios que devem ser explorados, tendo como fator limitante o número de dados que devem ser coletados em um projeto para que seus requisitos possam ser completamente descritos, buscando evitar considerações ou interpretações incorretas dos requisitos de projeto, que podem induzir a erros de concepção nos princípios de solução propostos.

Em circuitos de *simples comando*, Silva (1998) explora a modelagem funcional das cargas mecânicas com a aplicação dos seguintes parâmetros:

- Domínio: que indica se a carga pertence ao domínio linear ou rotacional;
- Posição: referindo à influência da gravidade em relação à carga movimentada;
- Reação: que indica a reação que o elemento de atuação recebe ao movimentar a carga mecânica;
- Controle: que descreve como a carga deve ser movimentada, referente a critérios de controle de força/torque, velocidade linear/angular ou posição.

Silva (1998) define estes parâmetros como sendo gerais e aplicáveis a todo tipo de carga mecânica, independente do princípio de solução optado na sua movimentação. Portanto, apesar desta modelagem de cargas ter sido utilizada como descrição de parâmetros para elaboração de projetos hidráulicos, esta pode ser estendida também para projetos pneumáticos.

Em projetos pneumáticos seqüenciais, antes da definição de detalhes específicos dos circuitos que formam o sistema pneumático a ser implementado, deve-se realizar a identificação e coleta inicial das necessidades dos *clientes*, tornando assim possível definir os seguintes parâmetros e detalhes do projeto:

- *Esquema da instalação*, cuja finalidade é oferecer ao projetista maior familiarização com o processo a ser comandado e a disposição física espacial das partes que o compõe, através de um croqui simplificado;
- *Formulação verbal* do problema, que em conjunto com o *esquema da instalação*, descreve-se a forma com que os elementos do esquema são acionados e interagem com os demais.

Portanto, os projetos pneumáticos seqüenciais iniciam-se com a pré-definição do princípio de solução a ser adotado, representado através do esquema da instalação e da formulação verbal, acrescido da definição de que o mesmo deve ser solucionado através da pneumática. Porém, é conveniente que as metodologias de projeto sejam também utilizadas na concepção física dos princípios de solução destas etapas iniciais do projeto, segundo as necessidades dos *clientes* do projeto, que pode ser auxiliada pela ES. Portanto, deve-se previamente constatar a adequação das decisões tomadas no projeto em fases anteriores.

A escolha por um *esquema de instalação* que ofereça um princípio de solução pobre sobre a perspectiva das necessidades dos *clientes* da automação, pode exigir muitas seqüências de movimentação, muitos elementos de trabalho, que podem ser simplificados

com a adoção de princípios de solução que simplifiquem estes parâmetros. A adoção de um princípio de solução pobre tem influência direta sobre a performance negativa do resultado final do projeto.

É de extrema importância portanto a exploração desta etapa inicial, que descreve os requisitos do sistema de automação, de forma mais direta aos objetivos finais do processo, sem detalhar informações de *como* o sistema deve realizá-los, pois isto é obtido a partir do *esquema da instalação* e da *formulação verbal*, mas sim *o que* deve ser realizado, pois isto indica as necessidades dos *clientes*. Thibaut (1979), por exemplo, sugere algumas montagens para a transmissão de movimentos mecânicos, a partir apenas de atuadores lineares, demonstrando a ampla possibilidade de movimentos que são possíveis de serem obtidos como na transformação de movimentos lineares para rotativos, sistemas multiplicadores de força através de alavancas entre outras possibilidades.

Tendo a definição conclusiva da solução a ser adotada, pode-se prosseguir com as demais etapas do projeto. Para tornar a representação das ações e a seqüência de movimentações das cargas mecânicas em um formato explícito, de fácil leitura e rápido entendimento, evitando interpretações dúbias ou incompletas na leitura da *formulação verbal*, são empregadas diversas formas de representação alternativas, que oferecem a visão global da *formulação verbal*. Dentre as formas usuais cita-se o formato de tabela, seqüência cronológica e abreviada (FESTO DIDATIC-BRASIL, 1995).

Bollmann (1997), sugere a aplicação dos seguintes formatos:

- *Diagrama trajeto-passo* sendo uma forma não normatizada, porém de grande aplicação e utilização por profissionais que atuam na área de automação seqüencial, que pode ainda ser modificado, representando a escala de tempo em conjunto, recebendo o nome de *diagrama trajeto-tempo*;
- *Diagrama funcional*, sendo este normatizado pela IEC 848¹.

Em projetos pneumáticos seqüenciais que exijam seqüências de movimentações simples, a estrutura do *diagrama trajeto-passo* supre as necessidades de representação gráfica das seqüências de movimento. Porém, em projetos seqüenciais elaborados, que requeiram funções como: seqüências simultâneas, alternativas, saltos e repetições, bem como memórias e atrasos no comando, a representação por *diagramas funcionais* torna-se superior às demais formas de representação não normatizadas anteriormente citadas.

¹ “O Grafcet é um diagrama derivado da Rede de Petri e tornou-se base para norma IEC-848, sob a denominação de diagrama funcional. A norma alemã DIN 40719-6, na edição de 1992, agregou integralmente (em seus capítulos 1 a 8) o texto da IEC-848, uniformizando estas normas” (ATTIÉ, 1998).

O que torna a aplicação dos *diagramas funcionais* ainda não corriqueira e usual é a difícil aceitação por parte dos atuais projetistas, que preferem a utilização de ferramentas antigas, porém de seu conhecimento, em detrimento de ferramentas mais eficientes, indo ao encontro com as colocações de Trinkel (HENEY, 1998) referente as deficiências dos projetistas em hidráulica e pneumática. A mudança deste cenário passa pela formação dos novos profissionais e a atualização dos que já estão em atuação que, porém, não deve ser realizada bruscamente.

A vantagem adicional da aplicação destas formas, normatizadas ou não, de representação de seqüências de automação é a perfeita documentação do projeto. Porém, deve-se dar preferência pelas formas que sejam normatizadas por, a princípio, seguirem uma estrutura padronizada, bastando conhecer a norma empregada na documentação para uma leitura e coleta de informações sobre o projeto. A transição das formas usuais para os formatos normatizados deve ser conduzido visando reduzir ao máximo qualquer impacto significativo sobre o trabalho dos projetistas.

4.5.2 – Requisitos qualitativos complementares

Para tornar o projeto pneumático o mais adequado às finalidades dos seus consumidores, é preciso a interpretação das informações existentes na *formulação verbal* para a sua posterior inserção na elaboração dos esquemas pneumáticos.

Contudo, algumas destas informações muitas vezes não são possíveis de serem representadas através dos *diagramas trajeto-passo e funcional*. São requisitos não diretamente relacionados com a seqüência de automação de um processo, mas sim com o impacto direto ou indireto gerado pelo projeto, em setores que possuem algum tipo de relação com o equipamento a ser idealizado. Segundo uma abordagem de ES, considerações complementares sobre o projeto que venham a aproximá-lo dos objetivos dos integrantes do ciclo de vida deste produto em projeto devem ser almejados.

Bollmann (1997) considera como fatores complementares os relacionados às condições ambientais, como temperatura, umidade, impureza entre outros em conjunto com a flexibilidade na mudança da seqüência de operação e a disponibilidade da fonte de energia empregada. Vinadé, Pereira e De Negri (1999) também listam alguns requisitos importantes em um projeto pneumático, além dos fatores ambientais, que incluem considerações sobre custo, manutenção, rapidez e desempenho.

Dentre a vasta possibilidade de critérios a serem considerados como requisitos de projeto, Silva (1998) considera parâmetros como: eficiência de potência, custo, precisão,

facilidade de manutenção e facilidade de operação para o campo da hidráulica.

Como os componentes pneumáticos permitem a sua interação com outros domínios de energia, as possibilidades de solução são diversas, podendo ser personalizadas para aplicações específicas. Portanto, deve-se fazer a coleta destes requisitos para a sua consideração no projeto.

4.5.3 – Requisitos para a interface de projetos pneumáticos seqüenciais

Na interface entre o sistema de automação pneumático, os elementos de sinais realizam a entrada de informações. No caso dos projetos seqüenciais do tipo assíncronos, o prosseguimento das etapas do projeto seqüencial tem como fonte a conclusão de cada etapa de movimentação dos atuadores. Porém, em projetos mais elaborados, funções complementares podem ser requeridas, adequando acionamentos específicos necessários à automação. Portanto, a entrada destes dados é realizada com o auxílio dos elementos de sinais.

Seguindo o esquema da Figura 4.4, modificações podem ser inclusas nos blocos de elementos de sistemas, que podem ser ampliados e adaptados para tipos específicos de circuitos, conforme apresentado na Figura 4.5, que é um aprimoramento para os comandos seqüenciais assíncronos.

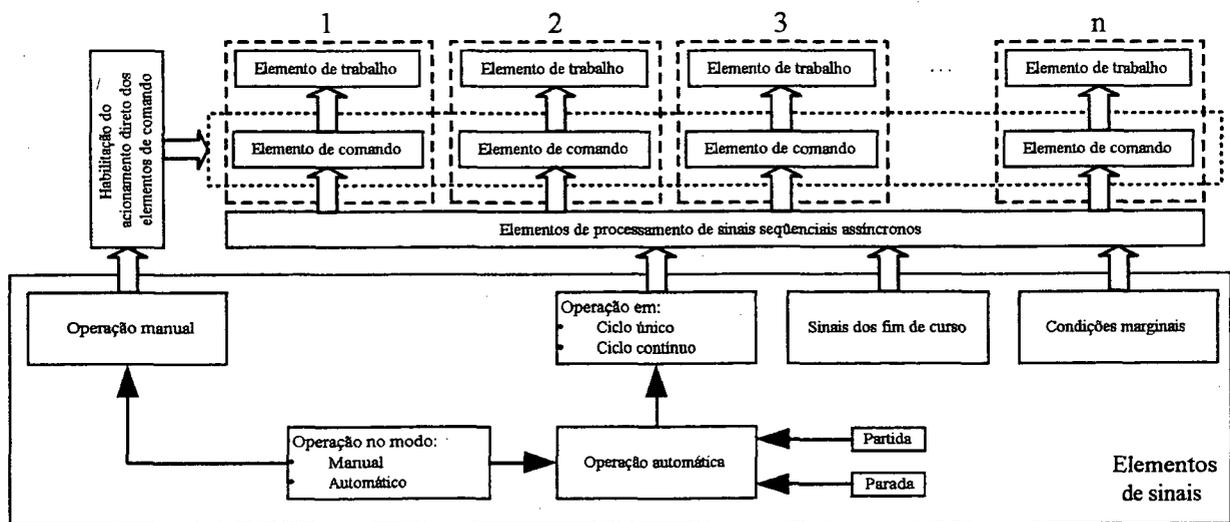


Figura 4.5 – Elementos de sistema de projeto seqüenciais (FESTO DIDATIC, 1992).

Dentre as funções tradicionais aplicáveis na interface entre os projetos de sistemas pneumáticos seqüenciais e o usuário, tem-se os acionamentos de *partida* e *parada* do ciclo seqüencial e sinais de *fim de curso*, utilizados em projetos seqüenciais assíncronos para gerar

a cadeia de acionamentos seqüenciais. Funções complementares podem ser inclusas, conforme requisitos adicionais de interface a serem satisfeitos pelo projeto de automação com os usuários, como (MEIXNER; KLOBER, 1986):

- *Reset* das cargas mecânicas, que reposicionam os atuadores para a posição inicial que os mesmos devem ocupar antes de iniciar a seqüência de movimentação das cargas mecânicas;
- *Condições marginais* de operação, nas quais enquadram-se os elementos que fornecem as condições mínimas de garantia de operação do sistema, garantindo segurança e requisitos mínimos para a realização das operações conforme o resultado final esperado como, por exemplo, gabinete do equipamento fechado, peça devidamente posicionada, material no depósito entre outras opções;
- *Operação manual* ou *automática*, onde neste são habilitados os comandos para a movimentação das cargas mecânicas conforme a seqüência previamente estabelecida; naquele são habilitados os comandos de acionamento direto dos elementos de comando, para a movimentação das cargas mecânicas, conforme requerido pelo operador e independente de uma prévia seqüência de movimentação estabelecida;
- Operação em *ciclo contínuo* e *único*, onde neste a seqüência de movimentação das cargas mecânicas dá-se início com o acionamento *partida*, sendo finalizado após a conclusão do último movimento; naquele tem-se o diferencial de que a seqüência de movimentação é reiniciada sem a necessidade de novos acionamentos de *partida*, operando de modo contínuo.

A interdependência dos acionamentos deve também ser considerada como, por exemplo, na habilitação do comando *partida* apenas quando anteriormente tenha sido acionado o comando *reset* das cargas mecânicas, entre outras possibilidades. Funções complementares diversas podem ainda ser inclusas, conforme as exigências do projeto como: contagem de peças, regulagens de níveis de força/torque ou velocidade de operação dos atuadores.

A inserção destes acionamentos deve ser feita com cautela e as propostas de projeto devem ser profundamente analisadas para a garantia de que o projeto seja suficientemente robusto e que ofereça segurança durante a operação. Como a grande maioria destes sinais de comando coletados são do tipo binários, o número de combinações possíveis entre estes comandos é expresso segundo uma potência de 2^n , em que n indica o número de comandos

possíveis de serem acionados. Num projeto em que se tem, por exemplo, acionamentos simples de *partida*, *parada*, *ciclo único*, *ciclo contínuo*, *operação manual* e *operação automática* o que totaliza 6 acionamentos, gerando 2^6 combinações, ou seja, 64 possibilidades de acionamentos. Apesar de alguns destes acionamentos não serem consistentes, por informarem ordens contraditórias ou divergentes, constata-se mesmo assim a complexidade do problema, pois o projetista deve garantir a perfeita operação do equipamento para todas estas possibilidades de combinações de acionamentos.

4.6 – Abordagens aplicadas aos projetos de sistemas pneumáticos

Segundo a perspectiva de Pahl e Beitz (1995), a funcionalidade global esperada por um sistema em projeto pode ser decomposta em sub-funções quando tem-se o projeto de um sistema complexo, buscando facilitar a busca por soluções e a posterior combinação destes princípios de solução na proposta do sistema final.

Dentre as abordagens para a elaboração de princípios de solução para as sub-funções de um sistema, Pahl e Beitz (1995) classificam em:

- *Originais*, com a geração de princípios de solução personalizados para cada aplicação;
- *Adaptativos*, nos quais o princípio de solução é adaptado de outras aplicações;
- *Variantes*, que com a inclusão, exclusão ou adaptação do princípio de solução pode-se chegar a funcionalidade desejada no sistema.

No contexto de projeto de sistemas pneumáticos, a abordagem *original* é aplicável a projetos que devem satisfazer a uma aplicação específica e que dificilmente o mesmo esquema proposto pode ser estendido para outra aplicação por possuírem detalhes específicos, como é o caso dos projetos seqüenciais, em que para cada seqüência de movimentação das cargas mecânicas requer um projeto personalizado.

Em contrapartida, os projetos *adaptativos* são utilizados quando, através de comparação, verifica-se a possibilidade de utilização de circuitos já conhecidos em uma determinada aplicação, que apesar de serem utilizados em situações diversas, devido à semelhança entre os requisitos de projeto, possuem a mesma funcionalidade final desejada.

No caso das soluções baseadas em *variantes*, são utilizadas as abordagens metódicas para a pneumática, que são aplicadas a situações que requeiram o projeto de sistemas personalizados, porém os princípios de solução são obtidos através de adaptações do projeto em relação às sub-funções exigidas no sistema. Nas próximas seções são discutidos com maior profundidade as aplicações e estruturas destes métodos.

4.6.1 – Abordagens *adaptativas* na construção de esquemas pneumáticos

As abordagens *adaptativas* são aplicadas quando se tem uma grande semelhança entre projetos que possuem requisitos semelhantes e que, portanto, podem ser resolvidos pelo mesmo princípio de solução, com base na experiência de *sucesso* de projetos anteriores, sem a necessidade de passagem por etapas que sirvam para a adequação do princípio de solução a nova situação proposta.

A fonte para obter estes princípios de solução bem sucedidos pode ser desde a experiência de especialistas humanos (EH), bibliografia técnica ou esquemas de outros projetos, com a reutilização de módulos que desempenham sub-funções que sejam convenientes para uma determinada utilização. Este é o caso das aplicações de projeto de circuitos de *simples comando*, comumente formados somente por elementos de comando e atuação.

4.6.2 – Abordagens *originais* na construção de esquemas pneumáticos

Nesta abordagem, a elaboração do projeto é baseada na experiência do projetista, em conjunto com testes de *tentativa e erro*, buscando criar uma combinação de componentes que seja capaz de satisfazer os requisitos do projeto, sendo a influência do projetista marcante, porém não tem-se a certeza de ser possível obter realmente uma solução.

Algumas vezes, as propostas de projeto obtidas desta forma demonstram ser mais compactas, pois é feito um projeto especificamente para uma determinada aplicação, de forma personalizada. Em contrapartida, a interpretação do projeto final por projetistas que não elaboraram o projeto muitas vezes requer um esforço adicional para a leitura de todo o esquema, devido às experiências profissionais distintas que cada profissional tem ao longo de sua vida profissional.

Nota-se que nesta abordagem a participação do projetista é marcante. Infelizmente a dependência extrema por fatores humanos no projeto pode ser prejudicial, devido a motivos pessoais que o projetista possa ter, que interfiram no seu desempenho.

Portanto, dentre o universo de possibilidades de esquemas que sejam adequados à aplicação em projeto, existe o risco do projetista selecionar a opção menos otimizada, segundo as necessidades dos *clientes* do projeto. Assim, as vantagens na aplicação de uma abordagem personalizada, que ofereça soluções específicas para uma determinada situação pode não existir.

Nas abordagens *originais* dos projetos pneumáticos, como o caso dos esquemas construídos intuitivamente, Bollmann (1997) lista como desvantagens desta abordagem a discordância com requisitos de qualidade no projeto como: rápida elaboração, facilidade para manutenção, supervisão e modificação. Com isso neste tipo de abordagem não devidamente estruturada, a fase de documentação deve receber destaque e atenção especial, devido às deficiências destes princípios de solução.

4.6.3 – Abordagem de *variante* na construção de esquemas pneumáticos

Nas abordagens de *variantes*, os esquemas pneumáticos são idealizados segundo prescrições e diretrizes precisamente estabelecidas, sendo a ação do projetista limitada à aplicação da abordagem selecionada. Fatores pessoais do projetista como capacidade, disposição, humor e outros tornam-se irrelevantes, e o resultado final pode ser facilmente interpretado por pessoas que não participaram do projeto (HASEBRINK; KLOBER, 1975).

Geralmente as abordagens *variantes* são aplicáveis a uma ampla variedade de projetos. Portanto, o esquema final, na grande maioria das vezes, não é otimizado. Isto ocorre pois, para propor uma metodologia generalista de solução, é de se esperar a não personalização do projeto em relação à aplicação do mesmo.

Mas em muitos casos, esta falta de otimização é compensada com a economia de tempo no projeto em outras etapas do seu ciclo de vida, como na manutenção a qual dependerá do volume global do mesmo e a facilidade de leitura dos esquemas (HASEBRINK; KLOBER, 1975). Em projetos que utilizam ferramentas de *hardware* poderosas como *controladores lógicos programáveis* (CLP), é preciso uma estrutura bem definida e metódica na elaboração do seu *software*, para usufruir de forma mais eficiente dos benefícios da sua aplicação.

4.6.4 – Princípios de solução para projetos de sistemas pneumáticos

Buscando identificar as aplicações, métodos e esquemas mais usuais nos projetos pneumáticos, segundo as abordagens apresentadas nas seções 4.6.1, 4.6.2 e 4.6.3, nesta seção são apresentados as montagens mais usuais de utilização destas abordagens nos projetos pneumáticos, conforme as partes que formam usualmente um esquema pneumático, como os elementos de comando/atuação, utilizados nos sistemas de *simples comando*, ou os elementos de processamento de sinais, como os utilizados nos projetos seqüenciais.

Em projetos de sistemas pneumáticos de *simples comando*, a perspectiva da abordagem *adaptativa* possui grande aplicação devido à simplicidade de correlação entre o modelo da carga mecânica a ser operada e o princípio de solução a ser proposto. Dentre as aplicações

tradicionais, tem-se os circuitos com controle de força e velocidade, que podem ser obtidos através dos princípios de solução listados na Tabela 4.2, que representam algumas das possibilidades de circuitos que oferecem estes tipos de controle.

Tabela 4.2 – Circuitos de *simples comando* utilizados segundo critérios de *adaptação*.

Tipo de controle	Dispositivo de comando		Nome do circuito
	Válvula direcional	Válvula de controle	
Velocidade	4 ou 5 vias 2 ou 3 posições	Reguladora de vazão unidirecional entre atuador e a VD, com regulagem na entrada	<i>Meter in</i>
	4 ou 5 vias 2 ou 3 posições	Reguladora de vazão unidirecional entre atuador e a VD, com regulagem na saída	<i>Meter out</i>
	5 vias 2 ou 3 posições	Reguladora de vazão de qualquer tipo no escape da VD	<i>Meter out no escape</i>
Força	4 ou 5 vias 2 ou 3 posições	Reguladora de pressão unidirecional entre atuador e a VD, com regulagem da pressão sobre o atuador	Controle de força

Porém, requisitos adicionais como descritos na seção 4.5.1 referentes à modelagem de cargas mecânicas podem ser considerados, onde indicações de preferência por um determinado tipo de solução ou até a não aplicabilidade técnica de outras são fatores a serem considerados.

No caso das abordagens *originais* e de *variação*, para os projetos pneumáticos sequenciais usualmente podem ser utilizadas as abordagens descritas na Tabela 4.3.

Tabela 4.3 – Principais métodos de projeto pneumáticos (BOLLMANN, 1997).

	Comandos pneumáticos	Comandos eletropneumáticos	Comandos pneutrônicos
Métodos tradicionais de projeto sequencial assíncrono	Intuitivo Passo-a-passo Cascata	Intuitivo Seqüência máxima Seqüência mínima	Não existem metodologias formalizadas
Métodos apoiados pela lógica binária	Método passo a passo generalizado		

Dentre estas abordagens, as intuitivas podem ser classificadas como seguidoras da linha de projetos *originais*, pois como descrito na seção 4.6.2, oferecem soluções personalizadas situação que busca-se atender, sendo assim *original*. As demais abordagens enquadram-se

como aplicações *variantes* do método que as deram origem, pois como descrito na seção 4.6.3, seguem precisamente as diretrizes dos métodos utilizados.

Conforme o tipo de comando binário a ser aplicado, em conjunto com o domínio do comando pneumático selecionado (pneumático puro ou eletropneumático), dentre as abordagens metódicas e intuitivas existentes, é a função do projetista selecionar qual princípio de solução a ser adotado, decisão esta que deve ser baseada e fundamentada nos requisitos de projeto, que são descrições complementares sobre a automação em projeto como apresentado na seção 4.5.2.

Nos métodos tradicionais de projeto pneumático seqüencial, o objetivo é realizar o projeto dos circuitos que fazem parte dos elementos de processamento de sinal. O restante dos componentes que classificam-se em elementos de sinal, comando e trabalho, tem pouca influência sobre o tipo de método a ser utilizado, sendo o domínio de energia que circula pelos circuitos a principal influência, podendo ser elétrico ou pneumático. Este parâmetro determina o tipo de sinal informado pelos elementos de sinal e o tipo de acionamento das válvulas direcionais de comando. Os elementos de trabalho sofrem pouca influência, com exceção do sensoriamento exigido na sua movimentação, que pode processar tanto sinais pneumáticos quanto elétricos.

Independentemente do domínio de energia empregado, de acordo com a semelhança entre os métodos estruturados, estes podem ser divididos em dois grupos:

- Passo a passo e seqüência máxima, que são métodos que utilizam circuitos de memória, com intertravamentos no seu acionamento, aplicáveis a qualquer projeto seqüencial assíncrono, sendo porém requeridos muitos componentes;
- Cascata e seqüência mínima, que utiliza uma estrutura similar aos métodos anteriores, com a diferença de que cada circuito de memória construído é destinado ao acionamento de uma parte da seqüência de movimentação das cargas mecânicas, de tal forma que não ocorra a movimentação de mais de uma vez da mesma carga mecânica por uma mesma memória, tanto no avanço como no recuo. Caso isto ocorra, deve-se construir uma nova memória para acionar o restante das movimentações das cargas. Esta divisão é feita para que seja evitado um duplo acionamento sobre as válvulas de comando dos atuadores. Estes métodos têm como vantagem a economia de componentes na grande maioria dos casos, porém em alguns projetos a solução oferecida pelo método deve ser posteriormente analisada para uma adequação e correção de possíveis inconsistências, o que torna a sua aplicação não totalmente direta.

As formações básicas destes quatro métodos citados são as estruturas de memórias, as quais, conforme descrito pelo nome, tem a função de armazenar algum tipo de informação, que neste caso são informações referentes em que etapa da seqüência de movimentação das cargas mecânicas o equipamento encontra-se realizando. Na representação de diagramas de blocos lógicos, a simbologia de uma memória pode ser representada pela Figura 4.6, na forma simplificada.

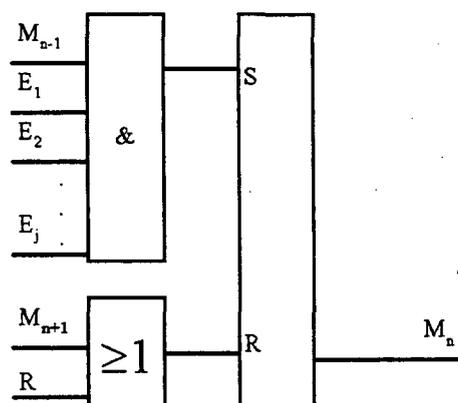


Figura 4.6 – Diagrama das memórias com intertravamento.

As memórias possuem como característica de operação uma entrada de sinal *S* ou *Set*, que gerará uma saída de sinal contínua, mesmo quando da retirada do sinal *S*. Para encerrar a saída de sinal da memória, a entrada *R* ou *Reset* deve ser acionada. Dentre os tipos de memórias *Set/Reset* possíveis de se obter, tem-se três tipos:

- *Set* dominante, em que ao ser fornecido simultaneamente tanto os sinais de *Set* e *Reset*, a memória possui preferência por oferecer o sinal de saída;
- *Reset* dominante, em que ao ser fornecido simultaneamente tanto os sinais de *Set* e *Reset*, a memória possui preferência por *não* oferecer o sinal de saída;
- Biestável, em que tem-se duas possibilidades de saída, estando uma delas sempre acionada, sendo a habilitação de cada uma delas obtida pelos comandos de *Set* ou *Reset*.

A memória apresenta na Figura 4.6 tem como característica especial o intertravamento, obtido em uma cadeia de memórias, que ocorre quando um dos sinais necessários para o *Set* da *enésima memória* (M_n) é proveniente da memória acionada anteriormente (M_{n-1}), em conjunto com outros sinais, que para o caso dos projetos pneumáticos seqüenciais assíncronos

tem como origem os elementos fim de curso que indicam a conclusão das movimentações das cargas mecânicas do passo anterior. O sinal da memória posterior (M_{n+1}) garante o desacionamento da memória anterior, gerando assim um ciclo de acionamentos e desacionamentos, ou uma cadeia de intertravamentos, mantendo sempre apenas uma memória acionada por vez.

O desacionamento da memória anterior deve ser realizado, pois somente uma memória pode estar acionada por vez, para que não somente seja possível identificar qual etapa da seqüência de automação está sendo realizada, como já descrito anteriormente, mas também para evitar que exista um duplo acionamento sobre as memórias, com ordens contraditórias como *Set* e *Reset*, que justamente pode ser evitado com o desacionamento da memória anterior, pois um dos sinais necessários para o acionamento de uma memória é o sinal da memória anterior. Em conjunto tem-se a possibilidade de desacionar todas as memórias através do sinal de *Reset* “R”.

Para a obtenção das memórias são utilizando as funções lógicas elementares “E”, “OU” e “NÃO”, as quais dão origem a todas as demais funções lógicas. Na Figura 4.7 é representada a forma expandida do diagrama de funções lógicas de uma memória com intertravamento, em que o sinal de saída da memória é utilizado como realimentação no diagrama lógico, para os casos *Set* dominante e *Reset* dominante.

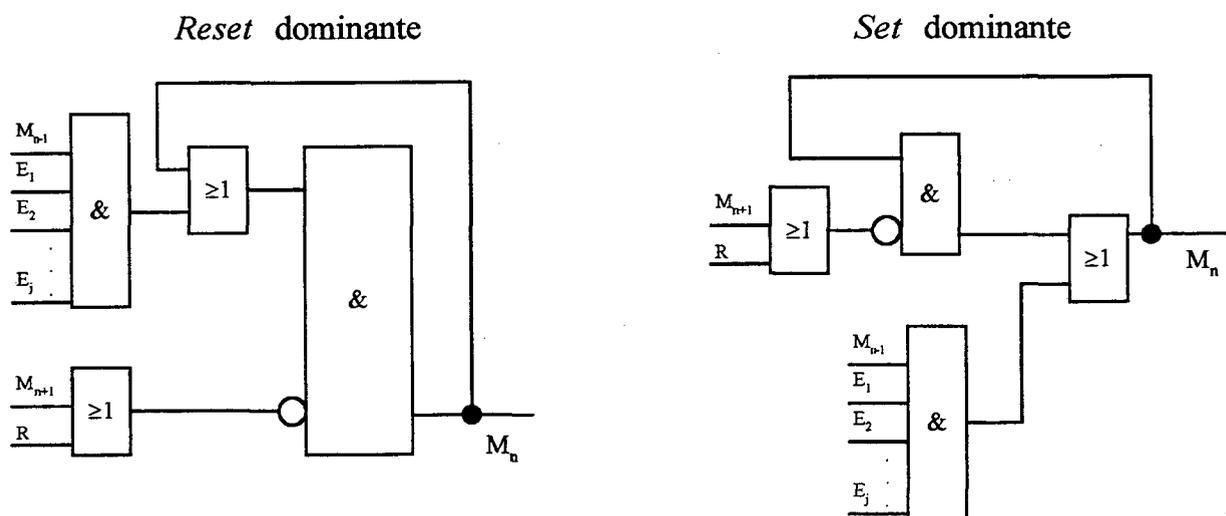


Figura 4.7 – Diagrama expandido das memórias com intertravamento.

Os projetos que se beneficiam do método passo-a-passo como princípio de solução, utilizam componentes puramente pneumáticos em seus esquemas. Uma das formas de

construção das memórias nestes métodos é utilizando válvulas direcionais sem retorno por mola. Neste tipo de componente, ao ser fornecido o sinal para sua comutação, ocorrerá o redirecionamento do ar, sendo mantida a posição da válvula mesmo quando da retirada do sinal responsável pela comutação da mesma. A alteração do seu comportamento ocorrerá somente quando for fornecido sinal que solicite ação contrária à anteriormente realizada e armazenada. Na Figura 4.8 tem-se o esquema do circuito que realiza a memorização de sinais, segundo uma seqüência de intertravamentos, cujos índices adotados representam a mesma funcionalidade que os diagramas da Figura 4.6 e Figura 4.7. Este tipo de memória pode apresentar tanto a configuração *Set* dominante como *Reset* dominante, através da utilização de válvulas direcionais com áreas de pilotagem diferenciadas.

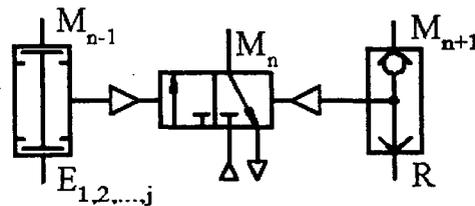


Figura 4.8 – Memória pneumática pura utilizando componentes com memorização.

Nestes circuitos, como é desejada uma função de intertravamento, para o acionamento de uma memória através do comando *Set*, são requeridos sinais que indicam a conclusão dos passos anteriores, e o sinal da memória anterior. O seu desacionamento ocorre quando a memória posterior é acionada, o que gera uma seqüência de encadeamentos na seqüência com que as cargas são acionadas. A característica deste circuito é que a memorização é realizada utilizando componentes que possuem memória, pois a válvula direcional permanece na posição do último sinal de acionamento recebido, devido a uma característica inerente ao componente.

Construções alternativas podem ser obtidas para componentes que, apesar de não possuírem a capacidade de memorizar, podem, em conjunto, possibilitar a memorização de um sinal, tornando-o permanente conforme apresentado na Figura 4.9, o qual segue a mesma lógica do diagrama da Figura 4.7.

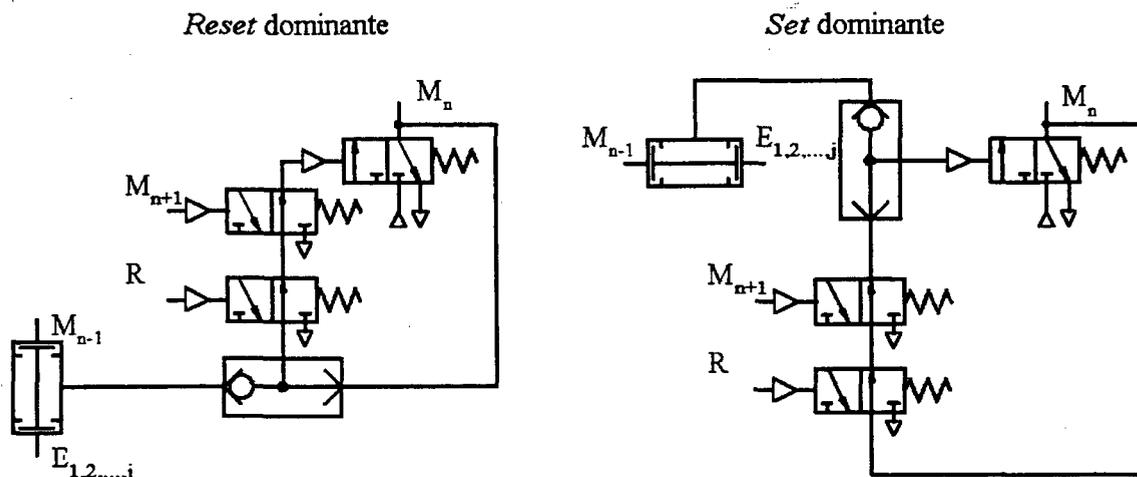


Figura 4.9 – Memória puramente pneumática utilizando componentes sem memorização.

Estes tipos de construção de memória em pneumática não são tão usuais devido à utilização de componentes adicionais em relação a memória da Figura 4.8. Neste tipo de construção, existe a possibilidade de obter tanto a formação com *Set* dominante como com *Reset* dominante, através da simples alteração na disposição dos componentes.

Para os métodos seqüência máxima e seqüência mínima, que utilizam componentes elétricos na construção das memórias, existe a preferência pela adoção por componentes que não possuem memorização, como o caso dos relés simples que possuem opção preferencial quando comparado aos componentes que possuem a capacidade de memorização como os relés de impulso de corrente e relés de remanência, devido à simplicidade e a flexibilidade que os relés simples oferecem aos circuitos elétricos. Portanto, na construção de memórias eletropneumáticas segue-se a lógica da Figura 4.7, conforme representado na Figura 4.10, onde o contato M_n^1 realiza a realimentação dos terminais da bobina do próprio componente M_n , tornando este permanentemente acionado.

¹ Apesar da simbologia de componentes elétricos adotar a letra “k” para designar relés, optou-se por adotar a simbologia “M”, para que a mesma funcionalidade possa facilmente ser identificada e comparada com os diagramas lógicos e as memórias pneumáticas demonstradas anteriormente.

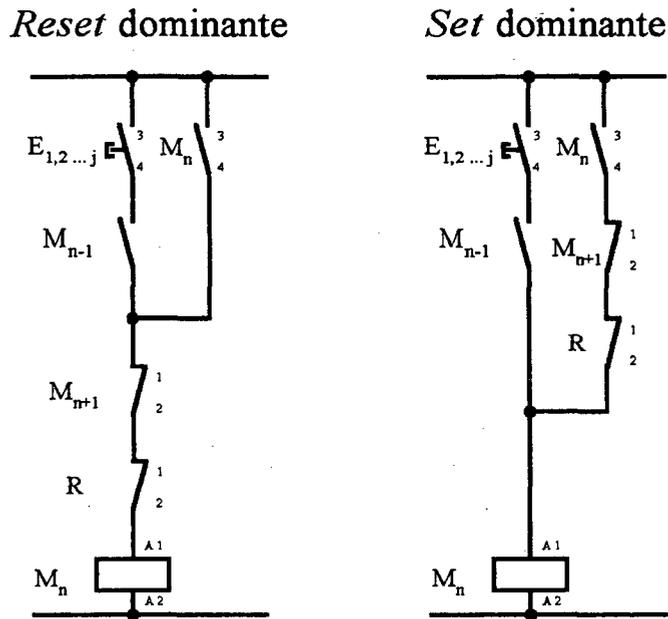


Figura 4.10 – Memória elétrica utilizando relé simples.

A forma de disposição dos componentes elétricos, assim como na Figura 4.9 para o esquema pneumático puro, na solução eletropneumática da Figura 4.10 proporciona obter memórias tanto *Set* dominante como *Reset* dominante.

Tanto nas memórias puramente pneumáticas como nas elétricas, ambas destinadas a realizarem intertravamentos, constata-se que são necessários na sua construção três sinais de memória, um proveniente da memória anterior, outro proveniente da memória posterior e por fim da própria memória que é acionada. Portanto, em um projeto seqüencial a ser resolvido pelos métodos passo-a-passo ou seqüência máxima, são necessárias no mínimo três memórias para obter-se a cadeia de intertravamento, ou seja, estes métodos são aplicados para projetos que tenham no mínimo três passos, pois para cada passo é destinada uma memória. Este limite mínimo de passos satisfaz plenamente os projetos seqüenciais, haja vista que movimentações seqüenciais de cargas mecânicas com um ou dois passos não podem ser classificados como projetos seqüenciais.

Em destaque, com o método passo-a-passo generalizado é possível realizar uma coleta de requisitos de projeto de forma metódica na fase inicial e previamente idealizar a funcionalidade dos circuitos, sem que para isso seja preciso transformar diretamente requisitos de projeto em um esquema. Portanto, é possível selecionar previamente a estrutura de intertravamento, para posteriormente optar-se por um método em específico (BOLLMANN, 1997).

4.6.5 – Combinação de princípios de solução

Através da modularidade que se obtém com os diversos circuitos apresentados nas subseções anteriores, pode-se com o agrupamento dos mesmos obter a funcionalidade global desejada, conforme requisitos do projeto.

Mesmo com o desmembramento das unidades funcionais em circuitos, nem sempre pode-se realizar todas as possíveis combinações entre os mesmos. Dentre os fatores que impedem a exploração de todas as combinações entre os circuitos, segundo detalhes dos projetos pneumáticos, tem-se destaque aqueles relacionados ao perfeito fluxo de energia e sinal por entre os circuitos, segundo condições tecnicamente e economicamente viáveis (PAHL; BEITZ, 1995).

Por este motivo deve-se previamente optar por um determinado domínio de energia no projeto, como pneumático puro, eletropneumático ou pneutrônico, para que se tenha a perfeita interação entre os circuitos construídos, e que o resultado final atenda as necessidades dos *clientes* do projeto, pois alguns circuitos são destinados a operarem prioritariamente em um determinado domínio de energia.

Dentre a gama de combinações propostas para o projeto, apenas a opção que apresenta melhor desempenho deve ter prosseguimento nas demais fases do projeto. Portanto, o conjunto formado pelos princípios de soluções que possuem melhor desempenho deve ser priorizado. Em um ambiente de ES, diversas ponderações podem ser exploradas, conforme alguns requisitos de projeto apresentados na seção 4.5.2. Mas muitas vezes os requisitos impostos sobre o projeto podem ser contraditórios entre si, sendo a função do projetista avaliar a seleção da solução prioritária.

As informações que determinam a classificação de um determinado agrupamento de princípios de solução podem ter base segundo origens diversas, como fontes bibliográficas e a experiência profissional do projetista. Bollmann (1997) lista na Tabela 4.4 algumas limitações de alguns princípios de solução, segundo parâmetros para projetos nos domínios pneumático puro, eletropneumático e pneutrônico, que de acordo com o grau de profundidade e detalhamento exigido na fase em que se encontra o projeto, podem ser decisivos nos direcionamentos a serem adotados, podendo ser explorados outros fatores relevantes.

Tabela 4.4 – Limitações dos princípios de solução (BOLLMANN, 1997).

	Elementos pneumáticos	Relés	Elementos eletrônicos	CLP
Tempo de comutação	15 ms	15 ms	30 ns	2 ms
Número de comutações	10 ⁹	10 ⁷	Ilimitado	Ilimitado
Temperatura de operação	-20°C a 60°C	-20°C a 60°C	0°C a 70°C	0°C a 55°C
Sensibilidade				
Campos elétricos	Não	Não	Uso limitado	Uso limitado
Impurezas no ambiente	Um pouco	Sim	Não	Não
Uso em ambiente explosivo	Sim	Não	Limitado	Limitado
Ocupação de espaço	Média	Grande	Média	Pequena
Possibilidade de reutilização	Pequena	Pequena	Média	Muito boa
Fabricação do componente	Difícil	Muito difícil	Difícil	Muito simples
Possibilidade de mudanças	Difícil	Difícil	Difícil	Muito fácil

Estes dados podem auxiliar na seleção do domínio de energia do princípio de solução, segundo parâmetros referentes às condições de operação do projeto. Porém, com a contínua evolução das tecnologias em diversas áreas, alguns destes parâmetros podem sofrer alterações, sendo portanto necessário uma atualização destes dados antes de qualquer direcionamento no projeto.

Fontes bibliográficas de origens diversas listam algumas informações complementares que podem ser obtidas mais especificamente referentes à operação de diversos tipos de componentes como as válvulas direcionais e os elementos de sinais sensores. Detalhes mais específicos sobre limitações de determinados componentes podem também ser consultados com os respectivos fabricantes.

4.7 – Ferramentas computacionais nos projetos pneumáticos

Continuamente os meios utilizados na realização dos projetos tem evoluído, não somente com relação ao modo de abordagem do problema, mas também nas ferramentas de auxílio ao projetista.

O desenvolvimento de programas computacional é uma das formas de auxiliar o trabalho dos projetistas, que tem aprimorado a qualidade do projeto num todo, desde a:

- Estruturação de etapas com planejamento, coleta de necessidades dos *clientes*, documentação;
- Passando pela busca de princípios de soluções, combinação de soluções, escolha da solução mais adequada, fases de detalhamento do projeto;

- E finalizando com o projeto satisfazendo as necessidades dos *clientes* que fazem parte do restante do ciclo de vida do mesmo.

A ação destes programas computacionais, vão desde simples agendamentos de tarefas, passando por *software* de CAD, e chegando aos *software* especificamente de engenharia como CAE e CAM. A forma com que estes programas computacionais abordam o problema de projeto pode diferenciar o modo com que este é desenvolvido e as ferramentas computacionais utilizadas no seu desenvolvimento. Estes *software* são destinados a aplicações diversas como a que é exigido nos cálculos, processamento e manipulação de equações complexas, como nas análises de simulação em elementos finitos.

Para as áreas de hidráulica e pneumática, bem como para aplicações elétricas e eletrônicas, além da combinação destes domínios, *software* que auxiliam na montagem de esquemas através da constatação da funcionalidade do conjunto, como o *Automation Studio* (AS), são utilizados para a constatação de incoerências e posteriores correções, além de simplificar as funções básicas de CAD, com biblioteca de componentes. Diversas outras aplicações também destinadas à simulação dinâmica matemática para a área de engenharia estão disponíveis.

Porém, em todas estas aplicações a função do projetista permanece sendo de extrema importância, pois é a partir deste profissional que as decisões finais sobre o projeto são tomadas, sendo os *software* destinados apenas a submeter o projeto em execução a condições similares de operação, sendo a função do projetista tomar as decisões para o seu refinamento, fundamentado no seu conhecimento prévio.

Nesta circunstância o projetista deve previamente possuir conhecimento suficiente para poder fazer as opções corretas. Com isso, os *sistemas especialistas* (SE) são uma aplicação viável na oferta de conhecimento especializado ao projeto, suprimindo uma possível deficiência do projetista em uma determinada área de conhecimento ou simplesmente exercendo uma função complementar junto à equipe de projeto. Neste sentido, *software* diversos têm aplicação no auxílio ao projeto, conforme descrito no fim do Capítulo 2.

No próximo capítulo são apresentadas as etapas de construção de um protótipo computacional para a realização de projetos no domínio pneumático, descrevendo as evoluções segundo a técnica de SE discutidas no Capítulo 3, em conjunto com a contextualização das funcionalidades disponíveis e os princípios de solução utilizados na sua construção.

Capítulo 5 – Etapas de desenvolvimento do protótipo

Com o embasamento teórico apresentado até então neste trabalho, neste capítulo são descritas as etapas de desenvolvimento do protótipo computacional destinado à realização de projetos pneumáticos, conforme proposta do Capítulo 1.

As etapas seguidas no desenvolvimento do protótipo, bem como as descrições e justificativas pela adoção de determinados princípios de solução na elaboração do protótipo são fundamentadas com base nos conceitos teóricos apresentados nos capítulos anteriores.

Neste capítulo são abordados tópicos referentes às fases seguidas na elaboração do protótipo, desde a seleção de princípios de solução até a metodologia de desenvolvimento, etapas percorridas na construção do protótipo, técnicas computacionais utilizadas, ferramentas de apoio, paradigmas de implementação e finalizando com a descrição do protótipo.

5.1 – Metodologia de projeto de sistema especialista adotada no protótipo

Inicialmente, o objetivo deste trabalho é construir um sistema protótipo, com algumas limitações funcionais, mas com robustez suficiente para demonstrar a possibilidade de exploração da técnica de *sistemas especialistas* (SE) em projetos de sistemas pneumáticos, sendo modular e que possibilite futuras ampliações da *base de conhecimento* (BC), sem grande impacto sobre o trabalho já realizado. A prototipação tem como benefícios (COAD; YOURDON, 1993):

- Melhoria na dinâmica do desenvolvimento do *software*;
- Possibilidade de experiência na interação humana sobre o projeto;
- Possibilita descobrir requisitos desconhecidos;
- Teste preliminar do projeto;
- Liberação das partes funcionais tão cedo quanto possível.

As etapas de evolução deste protótipo seguem a abordagem metodológica de desenvolvimento de SE proposta por Gonzalez e Dankel (1993) na Figura 3.14. A passagem pelas etapas de evolução seguem o modelo *incremental* devido à semelhança existente entre esta abordagem e a forma com que as etapas de aperfeiçoamento dos programas em SE são realizadas, além de modularizar a sua construção.

A fase inicial de desenvolvimento do protótipo é formada pela *prototipação rápida*, que visa construir um protótipo com grandes limitações, mas cujos benefícios são a demonstração da potencialidade dos SE no *domínio de conhecimento* (DC) dos projetos de sistemas pneumáticos e direcionar o restante das etapas do projeto para posteriores refinamentos, até chegar ao protótipo com todas as funcionalidades desejadas. A *prototipação rápida* também oferece conclusões referentes a desempenho, operacionalidade e viabilidade (CHORAFAS, 1990).

5.1.1 – Análise da viabilidade dos SE no projeto de sistemas pneumáticos

Como apresentado no Capítulo 1, a ampla aplicação da pneumática em soluções industriais, aliada à escassez de projetistas que dominem o campo da pneumática, evidencia a possibilidade de exploração de ferramentas que complementem e auxiliem os projetistas em suas atividades.

Conforme apresentado na Tabela 3.4, aplicando os critérios considerados por Waterman (1986) na indicação da relevância da técnica de SE em projetos de sistemas pneumáticos, tem-se as constatações apresentadas na Tabela 5.1, relacionadas ao DC, ao *especialista humano* (EH), campo de atuação entre outros parâmetros.

Tabela 5.1 – Viabilidade da técnica de SE em projetos pneumáticos.

Possibilidade de desenvolvimento do SE		
E	Tarefa não requer senso comum	Os dados manipulados em projetos de sistemas pneumáticos são todos técnicos e não são requeridos conhecimentos de senso comum
	EH podem articular seus métodos	Parte do conhecimento em projetos pneumáticos está disponível em artigos técnicos ou bibliografias correlatas. Maiores detalhes podem ser obtidos com EH ou através da análise da documentação de projetos já realizados
	Existe EH	Sim
	Há consenso entre EH	Pode haver algumas divergências devido às experiências profissionais distintas que cada EH teve ao longo de sua vida profissional, mas que pode ser benéfico em projetos que utilizem mais de um EH, oferecendo maior refinamento
	Tarefa requer também habilidades heurísticas	Apesar de algumas fases do projeto de sistemas pneumáticos possuir atividades rotineiras como no desenho de esquemas, existem muitas etapas que exigem conhecimento especializado
	A tarefa é plenamente explorada e conhecida	Muitas áreas da pneumática já são conhecidas e aplicadas a algumas décadas, sendo reservada a técnica de SE as áreas da pneumática que não possuam alta volatilidade ou que estão em constante mudança
Conclusão: Como foram satisfeitas todas as condicionais, existe a possibilidade de construir um SE		

Tabela 5.1 – Viabilidade da técnica de SE em projetos pneumáticos (continuação).

Justificativa do desenvolvimento do SE		
Ou	Custo com EH grande	Como na grande maioria dos domínios de aplicação, o conhecimento especializado é valorizado, sendo portanto oneroso
	Perda do EH	Existe a possibilidade de perda
	EH raro	Conforme apresentado na seção 1.2, Bud Trinkel (HENEY, 1998) considera que existem poucas pessoas que tenham algum tipo de especialidade atuando nos setores de hidráulica e pneumática na indústria
	EH requerido em vários locais	Em algumas circunstâncias isto pode ser exigido, porém esta deficiência pode ser minimizada com os meios de transporte mais velozes e os canais de comunicação mais eficientes, como telefone, internet, videoconferência, ambientes virtuais de projeto, entre outros
	Atuação do EH em ambiente hostil	Este item não pode ser considerado como justificativa para aplicação da técnica de SE, pois a maior parte do trabalho do EH em projetos de sistemas pneumáticos é em ambiente de escritório
Conclusão: são satisfeitas a maioria das justificativas de aplicação da técnica de SE em projetos pneumáticos		
Desenvolvimento do SE é apropriado		
E	Tarefa requer manipulação simbólica	Em algumas das etapas do projeto de sistemas pneumáticos é utilizado conhecimento simbólico, que na grande maioria das circunstâncias ocorre quando é exigido o conhecimento especializado, como no levantamento de princípios de solução, priorização das soluções entre outros
	Tarefa requer solução heurística	Nas circunstâncias em que é exigida a manipulação de dados simbólicos, a atuação do EH prevalece, pois para realizar o seu trabalho é utilizado o conhecimento heurístico adquirido ao longo de sua vida profissional
	Tarefa não é tão fácil	Em algumas etapas dos projetos pneumáticos é preciso que o projetista tenha conhecimento especializado para selecionar o melhor princípio de solução
	Tarefa tem valor prático	A pneumática, como apresentado no Capítulo 1, tem amplo campo de atuação e um SE no auxílio ao projeto pode ser conveniente
	Tarefa é de proporção gerenciável	Apesar do campo de atuação da pneumática ser muito amplo, é possível construir um SE protótipo que, apesar de algumas limitações, pode demonstrar as potencialidades e os benefícios a serem obtidos. Trabalhos futuros podem ser conduzidos ampliando a atuação da BC do protótipo. Pesquisas no domínio hidráulico realizadas por Silva (1998) demonstram a viabilidade dos SE
Conclusão: a técnica de SE é apropriada para ser utilizada nos projetos de sistemas pneumáticos		

Constata-se após o estudo de aplicação da técnica de SE em projetos de sistemas pneumáticos, a viabilidade de prosseguir com o projeto de um protótipo que utilize esta técnica da *inteligência artificial* (IA).

5.1.2 – Especificação do protótipo

Após a constatação da viabilidade de aplicar a técnica de SE no domínio de projetos de sistemas pneumáticos, é percorrida a próxima etapa de *especificação* de requisitos que o protótipo deve satisfazer.

A etapa de *especificação* visa limitar o campo de atuação do protótipo inicialmente proposto, estabelecendo as metas a serem atingidas e a abrangência do DC a ser pesquisado. Esta recomendação é de extrema importância em projetos, que devido à grande abrangência do DC, como o caso dos projetos de sistemas pneumáticos, para que os esforços sejam direcionados (GONZALEZ; DANKEL, 1993).

No contexto da *engenharia simultânea* (ES), o protótipo deve ser desenvolvido abordando questões relevantes, referentes aos *clientes* do mesmo e as suas necessidades devem nortear o desenvolvimento do protótipo, além de que o mesmo pode possuir em sua estrutura conceitos da ES que auxiliem também no desenvolvimento dos sistemas pneumáticos.

Os SE, como apresentado na Figura 3.1, possuem muitos agrupamentos de *clientes*, tanto internos quanto externos ao projeto do SE e que possuem necessidades particulares, que em determinadas situações podem ser consideradas como irrelevantes; em outras podem ser contraditórias entre si. Para conciliar e adequar o tempo disponível aos esforços despendidos, e trazendo ao centro o objetivo principal deste trabalho, que é demonstrar a viabilidade de utilizar a técnica de SE nos projetos de sistemas pneumáticos, são considerados como prioritárias as necessidades dos *clientes* externos, personificados pelos usuários finais. Esta decisão é fundamentada nas considerações de Gonzalez e Dankel (1993) que priorizam as necessidades e a aplicabilidade do SE pelos usuários finais.

A *prototipagem rápida* tem a finalidade de também fornecer o retorno dos usuários, referente às reais especificações que devem ser priorizadas. Como meta inicial são consideradas as seguintes necessidades presumidas como sendo prioritárias pelos usuários:

- Interface de entrada e saída amigável e com alta interatividade com o usuário;
- Robustez, tanto durante a operação do SE pelo usuário como nos projetos de sistemas pneumáticos propostos;
- Velocidade de processamento alta;
- Versatilidade e adequação aos computadores e sistemas operacionais usuais;
- As saídas devem conter informações possíveis de serem interpretadas pelos usuários que formam o público alvo do protótipo ou que tenham algum tipo de conhecimento em pneumática, pois esta é uma característica inerente aos SE;
- Saída dos esquemas do projeto utilizando simbologias normatizadas;
- Facilidade de instalação e desinstalação, sendo todo o processo realizado pelo próprio protótipo;

- Ambiente computacional integrado, que não necessite de outros *software* para operar;
- *Help* das funções básicas do protótipo;
- Robustez na entrada dos dados sobre o projeto;
- Ambiente de avaliação do desempenho do protótipo simplificado e integrado ao mesmo.

Estas necessidades que acredita-se satisfazer as características esperadas pelos usuários, servem como meta inicial a ser alcançada pelo protótipo, e que podem ser confirmadas com a transição do trabalho para etapas mais avançadas. Conforme a construção do protótipo, é possível demonstrar a funcionalidade do mesmo aos usuários finais e, a partir deste ponto, confirmar a exigência ou não destas necessidades ou constatar outras que não foram detectadas.

Complementando o grupo de *clientes* do projeto do protótipo, em segundo plano nas prioridades está o *engenheiro do conhecimento* (EC), que por ser o responsável pela maioria das etapas deste trabalho e estar intimamente ligado ao trabalho, inclusive no levantamento das especificações do protótipo, este não requer suposições de potenciais necessidades, por ter grande interação com o mesmo.

Para simplificar o percurso entre o início do projeto do protótipo até chegar ao protótipo concluído satisfazendo as necessidades dos usuários, são citadas como necessidades do EC os seguintes itens:

- Modularidade, que permite e facilita futuras expansões;
- Minimização de trabalhos meramente computacionais de programação, priorizando as atividades de *engenharia do conhecimento*, apesar da grande parcela de dedicação sobre o protótipo ser na etapa de implementação;
- Satisfazer as necessidades dos demais *clientes* do protótipo, porém adequando o tempo e recursos disponíveis no trabalho.

Como o projeto deste protótipo é fundamentado predominantemente por conhecimento obtido de referências bibliográficas em pneumática, não foram consideradas as possíveis necessidades dos EH, devido à pequena interação com estes nas fases iniciais. Como os EH são priorizados apenas na etapa de *validação*, não existem necessidades específicas que devam ser satisfeitas.

Dentre os tipos de sistemas pneumáticos que possuem funções especializadas, conforme apresentado na Figura 4.2, para iniciar a cobertura do DC em projetos de sistemas

pneumáticos pelo protótipo é considerado o contexto dos projetos seqüenciais, que possuem grande aplicação na automação de processos industriais. Portanto, nesta fase da *prototipagem rápida*, o protótipo deve suprir as funcionalidades básicas de entrada de dados para projetos seqüenciais, e ser capaz de interpretar estes dados para realizar o projeto pneumático, e oferecer algumas saídas que descrevam como este deve ser.

Como descrito no Capítulo 4, dentre as formas tradicionais, porém não normatizadas de representar a seqüência de acionamentos de cargas mecânicas, o *diagrama trajeto-passo* é o formato mais difundido. Mas como este formato requer representações gráficas, que exige maiores esforços e tempo na implementação, não satisfazendo o objetivo principal nesta fase inicial que é construir um protótipo simples, é adotado o formato *abreviado*¹ que torna-se mais conveniente pois o usuário pode informar a seqüência de movimentação desejada, utilizando apenas caracteres. Dentre as diversas soluções possíveis para a pneumática, nesta fase inicial são exploradas as metodologias de projeto que utilizam a *pneumática pura* (PP), devido à proximidade entre este domínio e a formação básica do EC, sendo inicialmente direcionado o protótipo para a metodologia passo-a-passo.

5.1.3 – Projeto preliminar

Com a definição de algumas fronteiras da *prototipagem rápida*, pode-se prosseguir para as demais etapas do projeto do protótipo, tendo a certeza de que, no decorrer destas etapas, os esforços sobre o protótipo possuem direcionamento e fronteira. Na etapa de *projeto preliminar* da *prototipagem rápida* em SE, Gonzalez e Dankel (1993) sugerem a tomada de decisão referente à técnica de representação do conhecimento, método de inferência, definição de ferramentas e time de desenvolvimento. Nas demais subsecções são descritas as opções a serem utilizadas por cada tópico que compõe o *projeto preliminar*.

5.1.3.1 – Técnica de representação do conhecimento

Conforme descrito no Capítulo 3, a representação do conhecimento correlaciona o conhecimento coletado, com o formato que o conhecimento possui ao ser inserido em um programa computacional.

¹ Forma *abreviada* – é o modo de representar a seqüência com que as cargas mecânicas com deslocamento linear devem movimentar-se, utilizando letras para representá-las (exemplo: A, B) acrescido dos símbolos + para indicar o avanço da carga mecânica e – para indicar o recuo. Portanto, caso seja informada a ação B-, significa que é desejado que a carga mecânica B recue (BOLTON, 1997).

O impacto de uma determinada decisão certamente tem efeito sobre as demais fases do projeto, pelo conceito da ES. No caso da técnica de representação, o impacto é tanto na eficiência com que o conhecimento coletado é inserido na BC quanto nas ferramentas de SE a serem escolhidas, que devem previamente ter a possibilidade de representar o conhecimento segundo a técnica de representação selecionada.

Uma das técnicas de representação do conhecimento optada é a combinação *regras e fatos*, que é o formato tradicionalmente adotado nos SE.

Para manipulações de dados simples são utilizadas variáveis, que apesar de oferecerem poucos recursos, satisfazem plenamente os requisitos operacionais básicos de representação.

O conhecimento processado em projetos de sistemas pneumáticos pode ser considerado como estruturado, pois exige conhecimento aprofundado e detalhado sobre o domínio do problema, ao contrário do que ocorre, por exemplo, em aplicações de diagnóstico, classificação e interpretação, nas quais é exigido um conhecimento abrangente, porém não tão detalhado quanto ao motivo que correlaciona as causas e os efeitos (GONZALEZ; DANKEL, 1993). Pelo formato estruturado do conhecimento em projetos de sistemas pneumáticos, é utilizado no protótipo também a técnica de representação baseada na *orientação a objetos* (OO), que conforme já apresentado no Capítulo 3, possui muitas vantagens, que no caso do protótipo pode oferecer outros benefícios como:

- Semelhança entre a técnica de estruturação *todo-parte* em relação à dependência existente entre os sistemas, circuitos e componente. O mesmo ocorre na estrutura *generalização-especialização*, referente aos diversos tipos de *generalizações* de circuitos e componentes e suas respectivas *especializações*;
- Simplificação na representação do conhecimento devido às características fundamentais dos SE como *abstração dos atributos* que são relevantes ao protótipo, *encapsulamento* onde todos os *atributos* relevantes estão reunidos formando as *classes*, e a *herança* que simplifica a passagem de *atributos* e *serviços* para as demais *classes* dependentes;
- Modularidade e flexibilidade possibilitando alterações, ampliações ou restrições dos *atributos* e *serviços* das *classes*, que estendem-se a todos os *objetos*;
- Melhor gerenciamento dos dados que são conduzidos pelos *objetos*.

A combinação de *regras, fatos* e OO na representação do conhecimento caracteriza a necessidade de um SE com suporte híbrido. A estrutura de *classes* adotada inicialmente é fundamentada no modelo de SE desenvolvido por Silva (1998) para projetos hidráulicos, que

apesar de ser uma aplicação distinta da pneumática, possui muitas características semelhantes. Com relação às cargas mecânicas a serem movimentadas, na modelagem OO são considerados todos os *atributos* utilizados por Silva (1998), isto é:

- Domínio: que indica se a carga mecânica é do tipo linear ou rotacional
- Posição: que indica se o deslocamento da carga mecânica é no eixo horizontal ou vertical;
- Reação: indicando a interação que existe da carga mecânica sobre o dispositivo que está movimentando a mesma;
- Controle: que indica a forma com que a carga mecânica deve ser deslocada ao longo da sua trajetória;
- Descrição: campo destinado ao usuário realizar um breve detalhamento sobre a carga mecânica.

Como inicialmente foi estabelecida a atuação do protótipo em projetos seqüenciais, na complementação dos itens considerados na modelagem das cargas mecânicas, é incluso o *atributo* seqüência, que descreve a inter-relação existente entre a seqüência com que as cargas mecânicas devem ser movimentadas.

Nos itens de saída do projeto é utilizado a estrutura de relação entre sistemas, circuitos e componentes adotada por Silva (1998) e apresentado na Figura 4.3. Adicionalmente para a *classe* de *generalização* formada pelos circuitos, são consideradas novas *classes* de *especialização*, que fazem parte da operacionalização da seqüência de movimentação das cargas mecânicas.

5.1.3.2 – Método de inferência

Como a técnica de representação do conhecimento inicialmente optada é baseada em *regras*, com a manipulação de *fatos* e *objetos*, existe a possibilidade de adoção do encadeamento *direto* e *reverso* das *regras*. A opção do encadeamento é fundamentada no tipo de DC e no procedimento que o EH adota para chegar até a solução.

Em situações em que o EH para obter a solução de um problema inicialmente coleta todas as informações sobre o mesmo para por fim chegar a uma conclusão e indicar um princípio de solução, Durkin (1994) recomenda o encadeamento *direto* das *regras*, sendo este o encadeamento a ser adotado no protótipo.

Na resolução do *conflito* entre *regras* do protótipo é adotada a *busca em profundidade*, conforme apresentado na Figura 3.11. Como o protótipo destina-se não somente a propor uma solução principal mas também a sugerir soluções alternativas, as demais ramificações da *rede*

de inferência devem também ser percorridas na busca de todos os princípios de solução que sejam adequados aos dados de entrada do projeto a ser realizado. A classificação prioritária na adoção dos circuitos como princípio de solução ocorre conforme a seqüência com que estes são criados, sendo os primeiros predominantes sobre os demais.

5.1.3.3 – Ferramentas

Dentre as ferramentas computacionais utilizadas para construir o protótipo, pode-se dividi-las em *hardware* e *software*. Classificado como *hardware*, estão todos os itens físicos responsáveis pelo processamento binário, unidades de memória e periféricos. Desde a evolução das primeiras máquinas mecânicas binárias, passando pelos equipamentos de processamento binários eletrônicos até os atuais *computadores pessoais* (PC), o desempenho e recursos destas máquinas teve grande crescimento, em proporção inversa ao custo.

No contexto da IA, o resultado desta evolução também tem influência sobre as áreas de aplicação dos SE como constata-se na Figura 3.3, que no período da década de 80 teve grande crescimento, repetindo esta grande evolução também na década de 90 como relata Durkin (1994), que atribui isto à mudança de *hardware*, passando das estações de trabalho para os já populares e difundidos PC como apresentado na Figura 5.1, referente ao ano de 1992. Para o protótipo, esta indicação sugere a possibilidade de aplicar o padrão PC como *hardware* para a criação de SE e utilização pelos usuários finais, ampliando assim a gama de potenciais usuários.

Referente às ferramentas de *software*, a flexibilização gerada pelos sistemas *shell* segundo Durkin (1994), teve também influência na evolução dos SE nas décadas de 80 e 90, representando quase metade dos programas computacionais utilizados na construção dos SE, conforme apresentado na Figura 5.2, em dados de 1992.

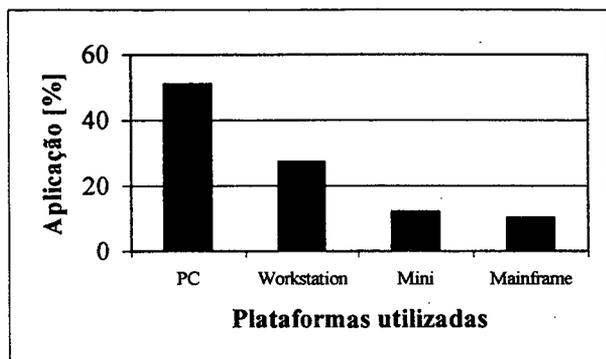


Figura 5.1 – Plataformas computacionais utilizadas na construção de SE (DURKIN, 1994).

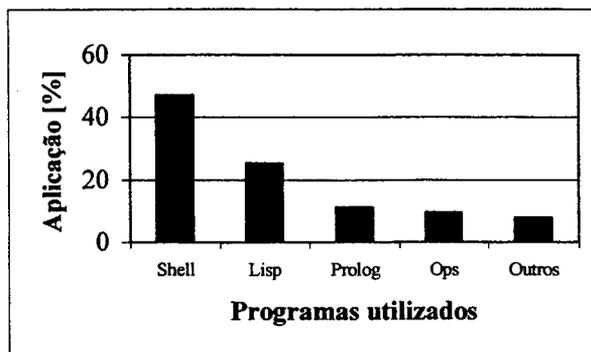


Figura 5.2 – Programas computacionais utilizados na construção de SE (DURKIN, 1994).

Analisando alguns sistemas *shell* pesquisados por Durkin (1994), e conciliando as características até então definidas como forma de representação utilizando *regras*, *fatos* e OO, com o encadeamento entre *regras*, sendo optado pelo tipo *direto*, chega-se à Tabela 5.2. Complementando esta tabela, é incluso o *shell* intitulado CLIPS¹, que foi utilizado por Silva (1998) em seu protótipo, tendo bons resultados.

Tabela 5.2 – Opções de sistemas *shell* para o protótipo (DURKIN, 1994).

Nome	Representação do conhecimento	Encadeamento entre <i>regras</i>	Requisitos de <i>hardware</i>	Custo
Goldworks	<i>Regras</i> , frames e OO	<i>Direto e reverso</i>	PC, Macintosh e Sun	Entre \$7.900 e \$9.900
Kappa-PC	<i>Regras</i> , frames e OO	<i>Direto e reverso</i>	PC	\$495
KEE	<i>Regras</i> , frames e OO	<i>Direto e reverso</i>	PC, Micro-Explorer workstations	Entre \$9.900 e \$30.000
ProKappa	<i>Regras</i> , frames e OO	<i>Direto e reverso</i>	HP 9000, Sun	\$14.450
ReMind	<i>Regras</i> , frames e OO	<i>Direto e reverso</i>	PC	\$3.000
CLIPS	<i>Regras</i> e OO	<i>Direto</i> ²	PC	<i>Freeware</i>

Como o CLIPS satisfaz previamente a todas as tomadas de decisão feitas até o momento, em conjunto com o êxito em tantas outras aplicações como citado na referência de Clips (1997), aliado à grande quantidade de material bibliográfico disponível na *internet* e ao

¹ CLIPS – SE desenvolvido pela Software Technology Branch (STB), NASA/Lyndon B. Johnson Space Center (GIARRATANO, 1998)

² Gonzalez e Dankel (1993)

custo zero de aquisição, inicialmente é feita a opção pela sua utilização. Outra vantagem que enfatiza a preferência pelo CLIPS é a possibilidade de interfaceamento do mesmo com outros *software* de programação, possibilitando o trabalho em conjunto com outros programas computacionais, ou até mesmo tornando-se parte de um determinado projeto que requeira a atuação com outros *software* em conjunto.

Caso sejam necessários outros *software* neste projeto, é dada a preferência por aqueles que são *freeware*¹ ou *shareware*², por não terem custo de aquisição.

Não havendo alternativas e sendo preciso utilizar algum *software* que implique em custo, dá-se preferência por aqueles que já estão disponíveis por conta de aquisições em circunstâncias anteriores. São considerados também a facilidade e o prévio conhecimento do programador e EC sobre estes *software* a serem utilizados.

5.1.3.4 – Time de desenvolvimento

Finalizando a etapa de *projeto preliminar* para SE, são definidas as pessoas que fazem parte do ciclo de vida de desenvolvimento do protótipo.

Conforme apresentado na Figura 3.1, pode-se identificar os recursos humanos exigidos para as seguintes funções: EC, EH, programador, auxiliar de escritório e usuários.

Na função de EC, que é exercida pelo pesquisador deste trabalho, são agregadas as atribuições de programador e auxiliar, devido às proporções limitadas a que este protótipo destina-se a atingir, sendo portanto possível esta junção de funções. Como requisitos para atuar como EC, Gonzalez e Dankel (1993) sugerem que o candidato tenha conhecimento básico sobre o DC, sendo este requisito suprido pelo presente pesquisador, que possui formação base em engenharia mecânica e com disciplinas cursadas nas áreas de hidráulica e pneumática, junto à *Universidade Federal de Santa Catarina (UFSC)* pelo *Programa de Pós-Graduação em Engenharia Mecânica (POSMEC)*.

Como este trabalho propõe-se a construir apenas um SE protótipo, com profundidade e abrangência limitada sobre o DC, a fonte de conhecimento a ser consultada tem origem na bibliografia técnica existente sobre projetos de sistemas pneumáticos. A participação do EH é priorizada nas etapas finais do protótipo, durante os testes.

¹ *Freeware*: programa computacional disponibilizado para utilização pelo desenvolvedor do mesmo sem que seja aplicado qualquer tipo de ônus ao usuário.

² *Shareware*: programa computacional disponibilizado para utilização pelo desenvolvedor do mesmo, geralmente por um período determinado podendo haver limitações em alguns recursos.

Para os usuários finais resta a função de teste, assim como os EH, porém com enfoque sobre as interfaces do SE protótipo. Estes usuários têm como característica algum tipo de conhecimento técnico em hidráulica ou pneumática, porém sem a profundidade do EH. Neste trabalho, a atividade de teste do protótipo por usuários não especialistas é auxiliada por estudantes dos cursos de tecnologia em *Automação Industrial e Eletromecânica da Sociedade de assistência aos trabalhadores do carvão (SATC¹)*.

5.1.4 – Prototipagem inicial

Com a definição de todos os parâmetros referentes ao protótipo, pode-se iniciar a etapa de implementação da *prototipagem rápida*. O resultado final desta etapa é justamente o protótipo que, apesar de algumas limitações, serve para nortear o restante do trabalho.

Gonzalez e Dankel (1993) definem a etapa de implementação como a que mais consome tempo. Portanto, as etapas de projeto que antecedem a implementação, assim como as que sucedem, como na avaliação dos resultados até então atingidos, devem ser criteriosamente percorridas, pois apesar do pouco tempo destinado a estas etapas, elas tem grande impacto sobre os resultados da implementação.

Ao fim desta etapa, tem-se um breve protótipo, executado no ambiente CLIPS, com interface via linha de comando para entrada dos dados do projeto e saída com alguns objetos representando circuitos e sistemas criados a partir das entradas, como apresentado no Apêndice A.

5.1.5 – Projeto detalhado

Na última etapa da *prototipagem rápida*, o projeto *detalhado* destina-se a adicionar, excluir ou alterar: os requisitos do protótipo, técnicas de representação, *software* utilizados dentre outras modificações que sejam necessárias.

Avaliando os resultados da *prototipagem rápida* em relação às especificações iniciais do protótipo, tendo como origem as necessidades dos *clientes*, conclui-se haver necessidade de ajustes no projeto *preliminar* do protótipo. As próximas seções descrevem e justificam detalhadamente as decisões e métricas tomadas no desenvolvimento do protótipo.

¹ SATC – Criciúma/SC, onde o presente pesquisador e EC é professor nas disciplinas de *Eletro-hidro-pneumática e Projetos de sistemas hidráulicos e pneumáticos*.

5.2 – Estrutura funcional do protótipo

Uma das deficiências apresentadas até então pelo protótipo é a falta de recursos na interface com o usuário, sendo o *shell* CLIPS desprovido de ferramentas próprias para esta tarefa, apesar do bom desempenho referente a manipulação simbólica, encadeamento entre regras e estruturação OO.

Para atender as necessidades inicialmente identificadas com relação à interação com o usuário, a dedicação sobre a interface do protótipo é de extrema importância, como constata Gonzalez e Dankel (1993), sendo os SE que possuem uma interface inadequada cabíveis de recusa na utilização por parte dos usuários, mesmo que possuam uma BC extremamente valiosa e elaborada. Dym e Levitt (1991) lançam o exemplo do *Dipmeter Advisor*, SE criado para realizar análises geológicas, onde de todos os esforços de programação, 42% são destinados ao interfaceamento com o usuário, como apresentado na Tabela 5.3, exigindo maior dedicação que outras tarefas, como na formação da BC.

Tabela 5.3 – Dedicação sobre os módulos de SE (DYM; LEVITT, 1991).

Partes do SE	Número de linhas de código [%]
<i>Máquina de inferência</i>	8 %
Base de conhecimento	22 %
Deteção de caracteres	13 %
Interface com o usuário	42%
Ambiente de suporte	15 %

Resultado semelhante foi atingido com o *Pride*, SE para projeto mecânico de subsistemas de manipulação de papel em fotocopiadoras, que possui aproximadamente 40% do código implementado destinado à interface com o usuário (DYM; LEVITT, 1991). Portanto, observa-se a complexidade existente em não somente obter a solução para um problema, mas como também em haver uma perfeita harmonia entre o SE e o potencial usuário do sistema.

Para a solução da deficiência existente na demonstração da interface de saída de esquemas pneumáticos, com a disposição e interação entre os componentes, conforme simbologias normatizadas, é adotada a solução utilizada por Silva (1998), aplicando o formato hipertexto no padrão HTML¹, que possibilita a oferta dos esquemas através de interface

¹ HTML – *Hypertext markup language* ou linguagem gráfica de marcação.

gráfica, com flexibilidade e interatividade que o usuário desejar por seguir o formato hipertexto, sendo este um padrão amplamente difundido.

Assim, é possível satisfazer as exigências de que os SE devem oferecer as justificativas para as saídas indicadas. Estando o computador conectado à *web*, pode-se também criar *links* específicos, limitando o acesso de informações pelo usuário conforme seja necessário (PETROUTSOS, 1999). Para ter acesso as saídas dos projetos pneumáticos, o usuário necessita de um programa com suporte para leitura do formato HTML.

Para a interface de entrada, a utilização de ambientes gráficos orientados a eventos, que é utilizado em *software* tradicionais para ambiente *Windows*, por exemplo, oferecem recursos que tornam o protótipo próximo das necessidades dos usuários. A inserção da interface gráfica torna-se plausível devido à possibilidade de operação do CLIPS com outras linguagens de programação.

A estrutura global do protótipo pode ser representada conforme apresentado na Figura 5.3, sendo possível realizar a entrada de dados tanto através da interface CLIPS padrão, como pela interface gráfica orientada a eventos. Isto torna possível avaliar o desempenho de cada uma destas interfaces em separado e facilita o trabalho nas etapas de formação da BC, que com a interface CLIPS padrão é mais simples devido a disponibilização de recursos existentes apenas nesta versão. Complementando, para simplificar a utilização do protótipo pelos usuários, o ambiente integrado de trabalho, com funções orientadas a eventos, possibilita que, em apenas um único programa, seja possível oferecer os dados de entrada do projeto, verificar a coerência dos mesmos e analisar as saídas do projeto. Todos estes recursos são assistidos pelo sistema *shell* e pela BC que o protótipo possui.

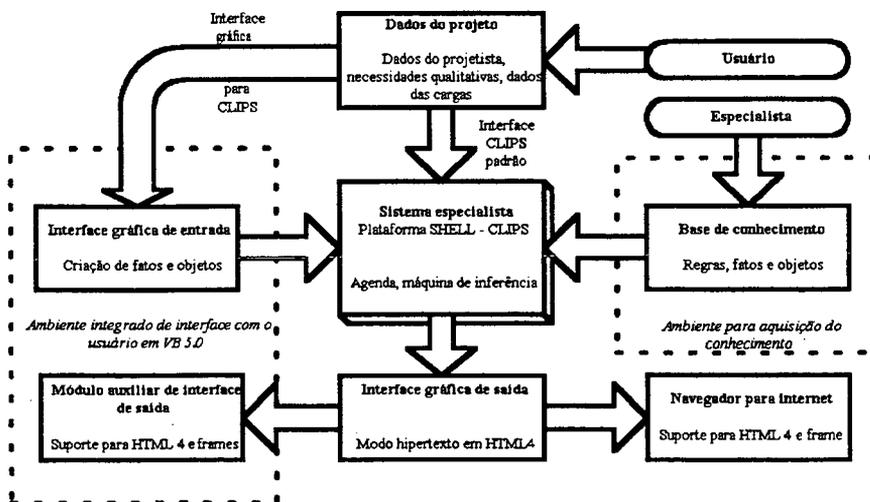


Figura 5.3 – Estrutura global do protótipo.

Todos estes itens do protótipo são gerenciados pela estrutura do sistema *shell* CLIPS, que coleta os dados do projeto a ser realizado, consulta a BC, identifica os possíveis princípios de solução, combina estes princípios de solução e por fim oferece os esquemas pneumáticos, segundo simbologia correspondente. Neste transcurso constata-se a dificuldade em chegar ao esquema pneumático, que além disto deve ser legível, com *layout* organizado e com correlação entre componentes devidamente estabelecida, para atender a funcionalidade inicialmente informada pelo usuário.

A BC durante a sua formação, independente da interface adotada e do *shell* empregado, deve ser construída em um módulo independente do restante do protótipo, pois esta é uma das características que diferencia os SE dos *software* convencionais. Nesta condição a BC independe de qual ambiente de interface é utilizado, pois a BC simplesmente descreve o conhecimento sobre o DC.

Nesta formação, evidencia-se a função do usuário e a do especialista, que apesar de possuírem interações com o protótipo, não estão inseridos nas atividades internas de desenvolvimento do mesmo. Os demais itens descritos na Figura 5.3, apesar de não serem citados, são de ação e responsabilidade do EC.

A fluência de dados e a relação entre todos os módulos do protótipo devem também ser adequadas para garantir a robustez durante a operação, pois um *software* com deficiências durante a operação e dito *inteligente* pode cair no descrédito pelo usuário.

5.3 – Domínio de conhecimento do protótipo

Como definido anteriormente, a atuação do protótipo está limitada a projetos do tipo seqüenciais assíncronos. Porém, como os projetos do tipo seqüenciais na sua estrutura são todos formados por circuitos de *simples comando*, como apresentado na Figura 4.5, utilizando componentes de comando e atuação, é disponibilizado no protótipo a opção de projetos de *simples comando*, que é representado pela supressão dos dados referentes aos elementos de sinais e processamento de sinais, existentes nos projetos pneumáticos seqüenciais.

Dentre as possíveis formas de disponibilizar os projetos seqüenciais, as estruturas metodológicas como o passo-a-passo proporcionam a simplificação dos projetos pneumáticos, que apesar de possuírem uma seqüência de elaboração dos esquemas previamente estruturada, ainda permitem pequenas modificações, para uma melhor otimização à condição a ser atendida.

Dentre as opções metodológicas existentes, inicialmente é explorada a demonstração da viabilidade da técnica de SE em projetos no domínio PP. Devido à limitação de tempo para

explorar todas as metodologias existentes, porém para poder demonstrar a possibilidade de utilizar os SE em outros domínios da pneumática, o campo de atuação do protótipo é expandido para atuar também na *eletropneumática* (EP). As metodologias a serem seguidas são o passo-a-passo, como anteriormente já iniciada a pesquisa, e o método seqüência máxima, que possibilita grande representatividade dentre as possibilidades metodológicas existentes.

A abordagem intuitiva, conforme apresentada no Capítulo 4, é desconsiderada neste protótipo, pois a elaboração de projetos pneumáticos segundo esta concepção é baseada na experiência, em conjunto com processos exaustivos de *tentativa e erro*, e portanto não existe disponível conhecimento devidamente estruturado conforme é esperado pelos SE, e que possa ser coletado pelo EC e posteriormente repassado para a BC do protótipo, pois fundamenta-se em grande parte na aleatoriedade da combinação entre componentes até atingir a funcionalidade desejada.

Devido à grande abrangência de aplicações dos projetos pneumáticos seqüenciais, com funções diversas, como: seqüências paralelas, alternativas, saltos, *looping* entre outras, é limitada a opção de entrada de dados do projeto a condições simples de operação, sem nenhuma das opções mais complexas acima citadas, para inicialmente demonstrar a possibilidade de construir um SE que gere integralmente um projeto pneumático, o que não seria possível caso seja ampliado demasiadamente o campo de atuação do protótipo, pois corre-se o risco de não ser possível construir o mesmo, de acordo com o tempo disponível.

Com esta limitação nas funcionalidades possíveis de serem realizadas, a representação da seqüência de acionamento das cargas mecânicas pode ser realizada no *diagrama trajeto-passo*, que apesar de sua simplicidade, supre por completo as necessidades de representação.

Aplicações utilizando *diagramas funcionais*, como apresentado no Capítulo 4, também são possíveis, porém a possibilidade de representar seqüências de movimentações simples de cargas mecânicas não justifica a sua aplicação, pois do ponto de vista gráfico, o mesmo é mais complexo de representar, quando comparado ao *diagrama trajeto-passo*, requerendo muito tempo de atividade computacional de implementação, do total despendido com o protótipo.

Todas estas limitações do protótipo são estipuladas para que seja possível conciliar a amplitude e a complexidade dos projetos de sistemas pneumáticos possíveis de serem realizados com o protótipo, pois a priorização de um destes itens inevitavelmente afeta o andamento dos demais. Portanto, os trabalhos sobre o protótipo são ponderados para tornar possível demonstrar a aplicação da técnica de SE em projetos de sistemas pneumáticos. Porém, com a atuação de outras frentes de pesquisa sobre este DC, é possível atingir um

campo maior do universo de aplicações dos projetos de sistemas pneumáticos.

Apesar de não ser o enfoque principal deste trabalho, futuramente atenção especial deve ser dedicada à pneumática. A não inclusão da pneumática no protótipo deve-se às limitações de tempo demandada na formação da BC, sendo esta opção possível de ser implementada em trabalhos futuros.

Deve-se em conjunto analisar que, com a evolução da pneumática, algumas aplicações podem tornar-se obsoletas ou limitadas. Independente da solução proposta pelo protótipo, este deve ser analisado conforme os princípios de solução que previamente encontram-se disponíveis na BC, não sendo portanto possível considerar que a solução proposta seja a mais adequada para uma determinada situação, pois as respostas que o protótipo pode oferecer são limitadas ao conhecimento existente na BC.

5.3.1 – Circuitos de gerenciamento

Na operacionalização de projetos pneumáticos seqüenciais, para garantir alta robustez e facilidades na operação, o circuito de gerenciamento é de extrema importância, pois é a interface existente entre o equipamento projetado e o usuário do mesmo.

As funcionalidades tradicionais que os projetos seqüenciais podem exigir, devem satisfazer inúmeras exigências, sendo algumas delas apresentadas na Figura 4.5.

Por serem os circuitos de gerenciamento construídos para satisfazerem as necessidades específicas que cada circunstância exige, são consideradas como entrada para os projetos de gerenciamento pneumáticos os comandos tradicionais comumente utilizados, que apesar de não satisfazerem de forma personalizada as exigências que o usuário deseja sobre o projeto, podem porém atender alguns possíveis requisitos do projeto.

Ampliações futuras da BC podem ser direcionadas a personalização dos comandos existentes nos circuitos de gerenciamento, conforme seja requerido pelo usuário, através da exclusão de comandos não necessários ao projeto sendo, portanto, conduzida apenas a adequação do circuito aos itens excluídos. Dá-se preferência a simplificação e exclusão de comandos ao invés da adição de novos, pois este procedimento é mais simples que a reordenação geral, tanto funcional quanto do *layout* dos esquemas, simplificando assim trabalhos futuros.

5.3.2 – Modularidade nos circuitos

Com a abrangência do protótipo atuando tanto em domínios PP, como EP, na construção de saídas em HTML, os circuitos propiciam alta modularidade, pois oferecem a funcionalidade desejada, sem que para isto seja preciso processar as unidades elementares, que são os componentes.

Analisando as possibilidades de combinações de saída até então definidas pelos tipos de projeto que devem ser realizados, constata-se a existência de inúmeras possibilidades de circuitos, que se forem considerados como um bloco único, elevam a especialização dos mesmos tornando-os muito específicos, o que exige, portanto, que inúmeros circuitos sejam criados, um para cada situação.

A solução é a ponderação na abrangência com que os componentes devem ser unidos, sem que para isto seja necessário construir um agrupamento de componentes que seja:

- Extremamente específico e especializado, sendo adequado para apenas uma única circunstância específica;
- Extremamente abrangente sendo, portanto, exigido a manipulação de cada componente separadamente até formar um circuito.

Circunstâncias como, por exemplo, dos circuitos de comando e atuação, exigem que para simplificar e reduzir as inúmeras combinações de circuitos, os componentes sejam divididos em sub-agrupamentos.

Isto ocorre devido ao fato de que componentes, como as *válvulas direcionais* (VD), devem ser inseridos nos circuitos especificamente para cada condição. Como existe uma grande diversidade de VD, exigindo tipos diferentes de alimentação e escape, diversificados tipos de comando para comutação, aliado a uma grande diversidade de combinações entre componentes para atender a determinados tipos de controle, constata-se a grande quantidade de combinações possíveis, sendo inviável construir um circuito para cada situação, pois é mais conveniente realizar agrupamentos entre componentes, que quando unidos formam o circuito desejado. No protótipo, estes agrupamentos descritos atendem os componentes localizados antes da VD, após a VD, e a VD em si.

5.3.3 – Nomenclatura dos componentes

A simplificação na designação e identificação de componentes segundo a norma ISO 1219 parte 2 (ISO 1219-2, 1995) permite a ordenação dos mesmos, proporcionando a rápida e segura identificação, sendo extremamente importante em circunstâncias como na montagem ou manutenção, em que componentes com características semelhantes, porém exercendo funções distintas no sistema devam ser designados e diferenciados.

Assim, cada componente recebe uma identificação única, na diferenciação em relação aos demais, através do preenchimento de campos que compõem o código do componente, conforme apresentado na Figura 5.4. O primeiro campo a ser preenchido é referente ao *número da instalação* a qual o componente pertence, iniciando a contagem progressiva a partir do numeral 1.

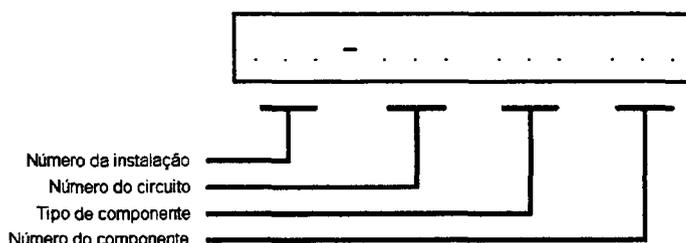


Figura 5.4 – Campos do código dos componentes.

Posteriormente é inserido um hífen, prosseguindo com o preenchimento dos demais campos, que na seqüência indicam o *número do circuito* ao qual o componente pertence, com contagem progressiva a partir de 1, sendo em especial utilizada a notação 0 para o caso dos acessórios. Na continuidade, tem-se a designação do *tipo do componente*, através das notações da Tabela 5.4.

Tabela 5.4 – Designação do tipo de componente.

Tipo de componente	Designação
Bombas e compressores	P
Atuadores	A
Motor de acionamento primário	M
Sensores	S
Válvulas	V
Todos os demais tipos de componentes	Z ou as demais letras não citadas

Por fim, o último campo do código dos componentes indica a contagem progressiva a partir de 1 da quantidade de componentes existentes, especificamente para cada *circuito e tipo de componente*. No caso da existência de apenas um único componente de cada tipo em um determinado circuito, o último campo do código pode ser descartado, como apresentado em exemplos na norma ISO (ISO 1219-2, 1995), apesar do texto da norma não descrever nada sobre a obrigatoriedade deste procedimento de exclusão. Todas estas informações que compõem o código do componente dentro da instalação devem estar localizadas ao lado do respectivo componente, grafados com uma moldura retangular para destacá-los.

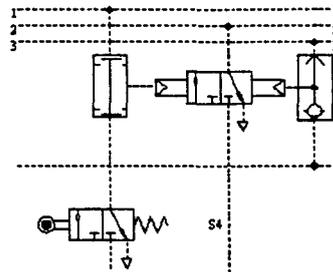
Na transcrição destes detalhes da norma ISO 1219-2 para o protótipo algumas simplificações são necessárias devido a limitações computacionais, sendo a designação do código dos componentes ao lado do mesmo sobre o esquema muito elaborada para ser expressa no formato HTML, tornando-se portanto inviável, impossibilitando seguir rigorosamente o que a norma solicita.

De modo alternativo, a descrição do código de cada componente que compõe um circuito é apresentada no formato de tabela, no detalhamento de cada circuito que forma o sistema, o que é mais simples de ser representado.

Outro detalhe a ser ponderado na geração dos códigos dos componentes refere-se a definição das fronteiras de cada circuito, pois esta delimitação é subjetiva, podendo um componente pertencer possivelmente a mais de um circuito, afinal todos os componentes estão inter-conectados. Geralmente no estabelecimento das fronteiras dos circuitos são adotados conceitos referentes à união de componentes destinados a exercerem a mesma sub-função dentro da funcionalidade global do sistema.

O mesmo ocorre com o número da instalação a qual o circuito pertence. Para tornar mais otimizado o processamento dos dados pelo protótipo, os circuitos são gerados e posteriormente combinados, podendo os circuitos pertencerem a mais de um sistema a ser proposto como solução. Neste sentido, o primeiro campo na designação do código dos componentes não é preenchido pelo protótipo, sendo este parâmetro possível de ser informado posteriormente, com a definição do número da instalação. Na Figura 5.5 é apresentado um circuito que exemplifica as simplificações adotada em relação a norma ISO 1219-2 (ISO 1219-2, 1995) na notação dos códigos de campo apresentados pelo protótipo.

Circuit 5



Lista de Componentes

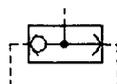
Componente	Tipo	Grupo	Código
Component V 4	valvula 3/2 nf	componente_v	#-5V1
Component V 5	valvula condicional logica tipo e	componente_v	#-5V2
Component V 6	valvula condicional logica tipo ou	componente_v	#-5V3
Component S 2	rolete fim de curso na posicao s4	componente_s	#-5S1

Figura 5.5 – Exemplo de esquema de circuito fornecido pelo protótipo.

Destaque para o elemento sensor de fim de curso da Figura 5.5, sendo considerado este componente integrante do circuito de memória, apesar do acionamento do mesmo ser realizado pela movimentação dos elementos de atuação. Por isso, este componente poderia também ser considerado como integrante dos circuitos de comando e atuação, como é o caso de alguns exemplos existente na norma ISO (ISO 1219-2, 1995).

Como no protótipo é utilizado o formato hipertexto, o detalhamento de cada componente pode ser visualizado conforme o usuário desejar. Na Figura 5.6 tem-se o exemplo do detalhamento de um dos componentes existente no circuito da Figura 5.5.

Component_V 6



- **Nome:** valvula condicional logica tipo ou
- **Origem:** Circuit 5
- **Código:** #-5V3
- **Descricao do Componente:** Este componente tem a funcao de realizar uma logica do tipo OU entre os sinais provenientes do comando de RESET das cargas e RESET realizado pelo passo posterior

Figura 5.6 – Detalhes sobre componentes dos circuitos.

Com as informações da Figura 5.6 pode-se obter detalhes diversos, como o circuito ao qual o componente faz parte, o código e a breve descrição da função que o componente exerce no circuito.

5.4 – Estrutura de classes

No DC de projetos de sistemas pneumáticos no campo de atuação seqüencial das cargas mecânicas, após a conclusão da *prototipagem rápida* é possível ter o refinamento da estrutura de *classes* para o caso específico deste protótipo. Para os circuitos de carga são incluídos os seguintes atributos:

- **Tipo_comando**: define o tipo de sinal de comando para acionar a válvula de comando, podendo ser PP ou EP;
- **Atuação**: representa todos os componentes que são conectados após a válvula de comando, como atuadores ou válvulas de controle de vazão ou pressão;
- **Elemento_comando**: indica qual tipo de válvula de comando a ser adotada;
- **Alimentação**: indica as alimentações e escapes que o circuito possui conforme o tipo de válvula de comando.

Estes três últimos tópicos descrevem o agrupamento parcial de componentes conforme já descrito na seção 5.3.2.

A abrangência do DC do protótipo possibilita a utilização de componentes de diversos tipos como: atuador, sensor e válvula. Por isso, complementando os circuitos de carga mecânica são estabelecidos os seguintes atributos:

- **Componente_A**: indica quantos componentes do tipo *atuador* existem dentro do circuito;
- **Componente_S**: indica quantos componentes do tipo *sensor* existem dentro do circuito;
- **Componente_V**: indica quantos componentes do tipo *válvula* existem dentro do circuito;

Estas designações da quantidade de componentes existentes nos circuitos conforme o tipo, auxilia na definição dos nomes que os componentes possuem, como apresentado na seção 5.3.3.

Como o protótipo realiza também projetos seqüenciais, os circuitos de carga mecânica para esta aplicação possuem algumas diferenças, sendo a *abstração* deste circuito expandida para possibilitar a inclusão de novos parâmetros exclusivos para a aplicação seqüencial, sendo criada a subclasse de *especialização* `circuito_carga_seqüencial` da *classe* de *generalização* `circuito_carga`, com os seguintes atributos adicionais:

- **Comando_avanço**: indica quais são os passos em que a válvula de comando deve ser comutada para obter o movimento das cargas mecânicas para o avanço,

conforme a seqüência previamente definida pelo usuário;

- Comando_recuo: indica em que circunstâncias a válvula de comando deve ser comutada para realizar o recuo do atuador do circuito desta carga mecânica.

Para os circuitos seqüenciais formados pelos elementos de sinais e processamento, são criadas *especializações* distintas, tanto para os projetos pelo método passo-a-passo, como para o método seqüência máxima. Como para cada um destes tipos de metodologias utilizam-se componentes de origem distinta como a pneumática, que é o caso do método passo-a-passo, ou elétrica, caso do método seqüência máxima, cada um destes métodos possui subclasses distintas, conforme o domínio de energia que os componentes manipulam. Destaque para a *classe* de componentes elétricos, que possui nova estrutura *todo-parte*, onde é possível identificar as partes que formam os componentes elétricos. O refinamento na estrutura de *classe* e seus *atributos* é apresentado na Figura 5.7, com alguns dos *serviços* utilizados.

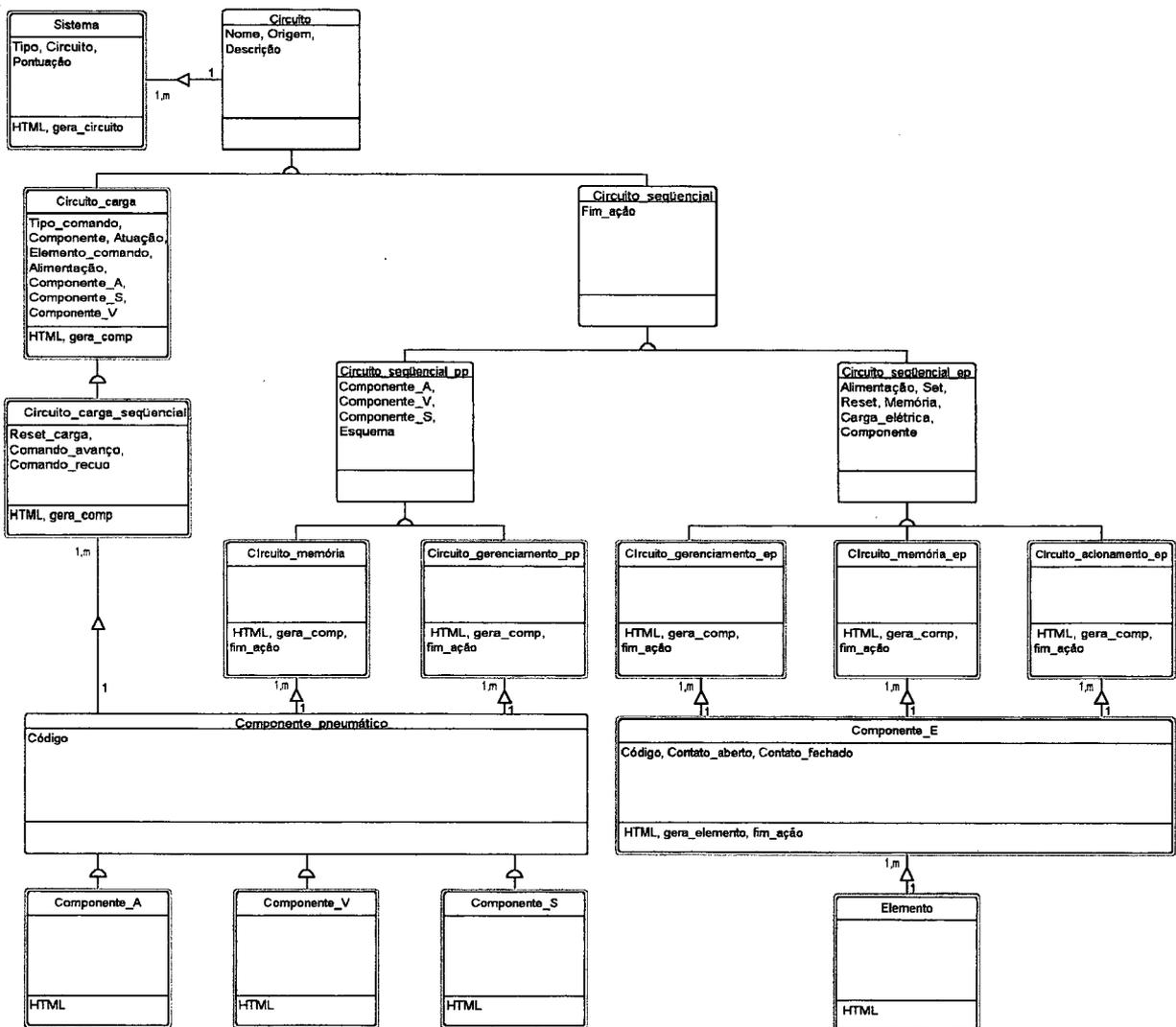


Figura 5.7 – Estrutura refinada das classes do protótipo.

Internamente ao protótipo, o processamento do projeto tem início com a prévia definição do tipo de projeto que deve ser realizado, seja ele PP ou EP. Esta seleção pode tanto ser conduzida pelo próprio usuário ou através do conhecimento existente na BC, que auxilia na definição de qual método é mais adequado. Após esta etapa, é gerado o *sistema* que desencadeia o restante dos procedimentos, com a seleção das partes com que este é formado, passando pelos *circuitos* e chegando aos *componentes*, e no caso específico dos componentes elétricos, atinge-se o grau de detalhamento das partes ou *elementos* com que este é formado.

No protótipo construído, a heurística utilizada nos projetos pneumáticos pode então ser representada pela estrutura de classes da Figura 5.7, que simula o entendimento utilizado no projeto de sistemas pneumáticos sobre este DC, com as partes que o compõem e a relação entre as mesmas. A heurística encontra-se também inserida na relação dos componentes durante a formação dos esquemas, pois não basta indicar quais são os componentes necessários para realizar o projeto, mas é preciso também indicar como estes são interconectados.

Situações em que devem ser considerados critérios adicionais na realização do projeto, como apresentado na seção 4.5.2, a priorização na utilização dos circuitos pneumáticos possíveis de serem aplicados pode ser conduzida através da busca pela melhor solução, quando existe mais de uma opção que satisfaça as condições técnicas mínimas de operação.

Nestes termos, Back (1983) sugere a adoção de uma *função critério*, que em situações em que tem-se pouca quantidade de informação durante as etapas iniciais do projeto, pode ser conduzida através da valoração de aspectos técnicos das soluções alternativas viáveis, buscando estabelecer valores numéricos para quantificar estas soluções. Esta *função critério* pode ser estabelecida segundo a adoção de *valores e pesos*, segundo alguns *critérios* relevantes e pertinentes ao projeto.

Os requisitos do projeto abordados como *critérios* a serem explorados pelos circuitos pneumáticos podem ter origens diversas. De modo geral, Pahl e Beitz (1995) sugerem para os projetos, considerações como funcionalidade, número de peças, montagem, transporte, operação, reciclagem, custo, entre outros. Novos fatores direta ou indiretamente correlacionados com os projetos pneumáticos podem também ser inseridos.

Na descrição dos *critérios* da *função critério* de um projeto, Back (1983) sugere que estes possam ser estabelecidos através dos requisitos de projeto, devendo estes atender algumas condições:

- Os *critérios* devem englobar todos os requisitos relevantes e as condições possíveis, de modo que nenhum ponto de vista seja esquecido na valoração da solução;
- Os *critérios* individuais, pelos quais as soluções são valoradas, devem ser independentes uns dos outros, de forma que as medidas adotadas para elevar o valor de uma variante relativa a um *critério* não devem influenciar o valor de outro *critério*;
- As propriedades do sistema a ser avaliado, em relação aos *critérios* devem ser, na medida do possível, identificadas em forma quantitativa ou, ao menos, qualitativamente.
- Os *critérios* devem ser estabelecidos positivamente ou em uma direção que leve a melhorar a solução, priorizando a adoção de parâmetros como, por exemplo, silencioso ao invés de não ruidoso.

Back (1983) também indica que estes *critérios* devem ser estabelecidos conforme os propósitos da valoração.

Com o estabelecimento dos *critérios*, pode-se prosseguir para a etapa posterior que compreende o confronto dos *critérios* com os princípios de soluções tecnicamente viáveis de serem aplicados, segundo uma escala de valoração única para todos.

Porém, cada situação em que está sendo realizado um projeto deve ser abordado de forma específica. Por isto, a relevância dos *critérios* sobre o projeto pneumático deve ser determinada a partir da definição das condições que devem ser atendidas. Isto pode ser feito através do estabelecimento da importância de cada *critério* ao projeto, através de *pesos* que priorizam alguns *critérios* que possuem maior relevância em detrimento de outros que não são tão relevantes, para uma determinada situação a ser atendida.

Estes *pesos* devem ser posteriormente correlacionados com o *valor* que os princípios de soluções em circuitos pneumáticos possuem, para todos os *critérios* explorados, a partir do somatório dos produtos destes parâmetros.

Assim a *função critério* que quantifica, segundo alguns *critérios*, o *valor* que um princípio de solução possui em relação à realidade imposta ao projeto através de *pesos*, pode ser representada pelo seguinte somatório:

$$F_j = \sum_{i=1}^n p_i v_{ij}$$

Equação 5.1 – Somatório para a descrição da *função critério*.

, onde o índice *j* representa a *função critério* para um determinado princípio de solução a ser analisado; o índice *i* representa o *critério* a ser analisado, totalizando *n* *critérios*; *p_i* indica os *pesos* considerados para o *critério* de índice *i*; *v_{ij}* indica o *valor* que o princípio de solução de índice *j* possui, segundo o *critério* de índice *i*.

Após a realização do somatório da *função critério* de cada princípio de solução, é possível determinar a solução mais conveniente a ser aplicada, através da *função critério* que possui maior valor.

Estes princípios de solução considerados neste trabalho vão desde os circuitos de comando simples listados na Tabela 4.2, até os métodos seqüenciais possíveis de serem utilizados e listados na Tabela 4.3. Para a finalidade deste trabalho, foram considerados os *critérios* listados na Figura A.10, segundo parâmetros descritos na seção 4.5.2.

Porém, devido a limitações de tempo, a valoração destes princípios de solução em relação aos *critérios* considerados não foi realizada, mas foram todos implementados no protótipo de forma parametrizada, sendo facilmente possível a inserção destes dados em propostas futuras, que deve ser realizada com o auxílio de um EH, devido a essência heurística contida nesta tarefa.

No próximo capítulo são apresentados detalhes referentes ao processo de teste do protótipo, que assim como as demais etapas de construção descritas neste capítulo, também possui procedimentos metodológicos. Com o teste pode-se ter a constatação final da adequação do protótipo às necessidades dos *clientes do software*, bem como a robustez e a adequação da BC, ou em caso contrário, determinar correções ou mudanças de paradigma a serem aplicados.

Capítulo 6 – Teste do protótipo

Finalizando o ciclo de desenvolvimento do protótipo em *sistema especialista* (SE), chega-se a etapa de teste, conforme o prosseguimento evolutivo descrito na Figura 3.14. Em um ciclo contínuo de melhorias, aprimoramentos e ampliações do campo de atuação sobre o *domínio de conhecimento* (DC), a etapa de teste busca realimentar o procedimento contínuo e incremental na elaboração do protótipo.

Neste capítulo são descritas as características da etapa de teste, a forma como deve ser conduzida e os responsáveis pela execução. Ao término é apresentado o teste do protótipo e os resultados obtidos.

6.1 – Importância da etapa de teste em SE

Muitas áreas de projeto utilizam os benefícios que os testes oferecem ao desenvolvimento de produtos. Em projetos mecânicos, por exemplo, os testes propiciam a obtenção de dados sobre o produto antes mesmo deste estar pronto, utilizando modelos simplificados para simulações de operação, tanto em condições reais quanto em ambientes computacionais, através de *software* de CAE. Ao término dos testes, os resultados obtidos auxiliam na determinação da adequação do projeto às especificações, direcionando o retrabalho dos projetistas ou simplesmente concluindo a etapa de projeto.

Com a mesma finalidade, porém com procedimentos específicos para adequação as circunstâncias particulares, os testes são também aplicados ao ciclo de evolução de programas computacionais. Porém, diferenças na abordagem sobre o DC em que o protótipo atua, indicam a necessidade de atuação diferenciada na etapa de teste, caso este dos SE.

A importância do teste no ciclo de evolução dos SE é exigido, pois erros cometidos por um programa derivado da *inteligência artificial* (IA), sendo portanto considerado *inteligente*, podem resultar em perda de credibilidade pelos usuários do protótipo, tanto em circunstâncias de operação rotineiras, como em aplicações extremas envolvendo risco de vida.

Os erros mais comuns cometidos em SE são (GONZALEZ; DANKEL, 1993):

- Falta de especificações ou simplesmente não utilização das especificações inicialmente estabelecidas no seu projeto;
- Erro de operação do programa computacional, conhecidos como *bugs*;
- Representação incorreta do conhecimento, gerando soluções incorretas ou impossibilidade de chegar à solução desejada.

Na abordagem incremental de evolução dos SE, em que a formação da *base de conhecimento* (BC) é realizada em etapas progressivas, os testes são extremamente importantes, pois indicam a possibilidade de continuação do trabalho ou a necessidade de um reprojeto com mudanças de paradigma sobre o SE.

Com a antecipação na identificação dos erros sobre o SE, que pela abordagem de *engenharia simultânea* (ES) agrega qualidade ao trabalho, pois isto ocorre ainda nas etapas iniciais, sendo exigido o retrabalho apenas na parte do conhecimento em formação, e não em toda a BC realizada até a identificação do possível erro.

6.2 – Diferenças no teste de programas computacionais convencionais e SE

Em programas computacionais que não seguem a linha dos SE, geralmente realiza-se o processamento de dados de forma algorítmica e devidamente estruturada. Portanto, é totalmente transparente e previsível a conexão entre as entradas e as saídas. Esta característica simplifica o teste nestes tipos de *software*, pois basta correlacionar as respostas de um problema conhecido com os dados acessados, utilizando um *teste de caso*. Havendo a correlação entre entradas e saídas, o *software* é considerado testado e adequado para a utilização.

Porém, esta mesma abordagem de teste de *software* não pode ser aplicada na íntegra aos SE devido a não existência de uma estrutura rígida e algorítmica no processamento de dados, devido justamente à atuação da *agenda* e da *máquina de inferência* como descrito na Figura 3.6, que em conjunto priorizam os encadeamentos entre as *regras* até chegar a solução do problema.

Por estas circunstâncias, a abordagem de *teste de casos* não satisfaz por completo os requisitos de qualidade no teste dos SE. Isto deve-se ao fato de que os SE possuem como características (GONZALEZ; DANKEL, 1993):

- O conhecimento inserido na BC é subjetivo, pois é obtido de fontes em que é preciso a descompilação para posteriormente inserção na BC, portanto não havendo objetividade desde a entrada dos dados do problema até a solução;

- Existência de incerteza sobre o DC, sendo os SE imprecisos e inexatos, assim como os *especialistas humanos* (EH) também são. Dentro de certos limites, esta incerteza pode ser tolerada;
- Os modelos de sistemas inseridos em programas computacionais convencionais podem ser testados em campo, sendo a adequação do modelo constatada através da simples comparação direta em *testes de casos*. Nos SE os modelos são mais complexos de serem formados e testados, pois devem ser obtidos através do conhecimento do EH. Portanto, o teste deve ser realizado apenas por pessoas extremamente capacitadas, que possuam ampla vivência sobre o DC e que possam correlacionar as entradas e as saídas do SE.

Em suma, Durkin (1994) define o teste em SE tão impreciso, quanto são incertos os possíveis *testes de casos*, não sendo possível submeter o SE a um teste padrão que defina se os resultados do sistema são corretos, exigindo avaliadores capacitados e que possuam profundo conhecimento sobre o DC, como o caso dos EH.

6.3 – Estágio do teste de SE

Apesar da etapa de teste não ser a que requer mais horas de trabalho sobre o desenvolvimento de um SE, porém é com certeza a que requer maior atenção e que exige conhecimento especializado, tendo importância tão grande quanto as etapas iniciais de definição do SE, pois serve como retorno na constatação da adequação do trabalho até então feito sobre o SE.

Segundo Waterman (1986), a área de teste para os SE deve ser, na seqüência, o laboratório através de *testes de caso*, e depois em condições de campo nos *testes de campo*. Gonzalez e Dankel (1993) subdividem estas diferentes fases de teste em *verificação* e *validação* (VV). A passagem por estas fases que formam a etapa de teste, seguem métricas que realmente possam auxiliar na evolução do SE, como já detalhado nas etapas apresentadas anteriormente na Figura 3.14, com a utilização de metodologias que auxiliem na construção do SE. Detalhes sobre estas fases que formam o teste, bem como as diferenças no teste de SE e a relação existente no processo incremental de desenvolvimento do protótipo são descritos nas próximas subseções.

6.3.1 – Verificação em SE

A fase de verificação é a que o *engenheiro de conhecimento* (EC) tem maior atuação. Por ser realizado por uma pessoa que, na maioria das vezes, possui um conhecimento muito

superficial em relação ao DC, porém com relativo conhecimento em SE, a fase de verificação possui atuação predominante sobre detalhes computacionais dos SE.

A verificação destina-se a eliminação de erros inseridos durante a etapa de implementação, sendo os procedimentos muito semelhantes à verificação que é realizada em programas computacionais convencionais, sendo para o caso dos SE com atuação tanto na BC quanto na estrutura de *máquina de inferência* e *agenda* (GONZALEZ; DANKEL, 1993).

Para o caso de SE construídos utilizando sistemas *shell*, a princípio a preocupação com a verificação do *shell* pode ser eliminada, considerando que este trabalho já foi realizado pelos responsáveis do *shell*, e que o mesmo encontra-se pronto para a utilização, pois mesmo que seja identificada alguma falha, o EC tem poucas possibilidades de solucioná-la pelo fato de que o *shell* é apenas uma ferramenta a qual o EC não possui acesso ao código fonte (GONZALEZ; DANKEL, 1993).

A verificação pode ser conduzida primeiramente analisando a adequação do SE as especificações inicialmente estabelecidas no projeto do SE.

Em uma segunda fase, são identificados os erros computacionais mais comuns, como erros de sintaxe e semântica, que por possuir características meramente computacionais, são identificados também em programas computacionais convencionais, podendo este trabalho ser conduzido pelo próprio EC.

6.3.1.1 – Erros de sintaxe

Os erros de sintaxe são descritos como falhas decorrentes da entrada, por parte do programador, de comandos incorretos, devido a:

- Erros de digitação, na chamada dos comandos;
- Erros na chamada de funções definidas pelo programador que não são possíveis de serem identificadas, pois são solicitadas antes destas terem sido carregadas na BC ou por simplesmente não existirem.

A origem destes erros está na etapa de implementação. De modo geral, as ferramentas de desenvolvimento de *software*, em geral, possuem inserida a possibilidade de previamente identificar estes tipos de erros cometidos. Na grande maioria das situações, a correção de erros computacionais deste tipo não deixam indícios de sua realização, e não implicam em mudanças conceituais da abordagem sobre o DC, sendo o maior esforço dedicado na identificação dos erros ao longo do código e posterior correção.

6.3.1.2 – Erros de semântica

Ao contrário do que ocorre com os erros de sintaxe, os erros de semântica têm origem na formação incorreta da BC. Por este motivo, a constatação deste tipo de erro dificilmente pode ser evidenciada apenas pela própria depuração do SE, tendo que ser identificado pelo EC. Dentre os erros de semântica mais comuns, que na grande maioria das vezes está ligado à representação do conhecimento, tem-se (GONZALEZ; DANKEL, 1993):

- *Regras* redundantes, em que a mesma *regra* é inserida na BC mais de uma vez;
- *Regras* conflitantes, em que para as mesmas condicionais, duas ou mais *regras* chegam a conclusões divergentes;
- *Regras* derivadas de outras *regras*, em que uma *regra* com muitas condicionais a serem satisfeitas possui a mesma conclusão que outra *regra* com menos condicionais, evidenciando portanto que a *regra* que possui menos condicionais representa a essência do conhecimento;
- *Regras* com inferência circular, que ocorre quando as *regras* são mutuamente acionadas, originando os *loopings*;
- *Regras* com encadeamentos sem chegar a uma solução, nas quais após ser percorrido um determinado caminho na *rede de inferência*, o SE é incapaz de chegar a uma solução;
- *Regras* nunca satisfeitas, que ocorre quando a *regra* possui condicionais que nunca são possíveis de ocorrer;

A busca por estes tipos de erros passa pela análise minuciosa de toda a BC do SE pelo EC, sendo este um trabalho exaustivo. Mesmo após a passagem por uma detalhada busca por erros, tanto de sintaxe como de semântica, isto não impossibilita a existência de falhas no SE. Segue na próxima seção a descrição da fase de validação, que visa também a busca por possíveis erros no SE que não podem ser localizados pelo EC, devido às suas atribuições e limitações dentro do projeto do SE, pois esta deve ser realizada em condições externas ao ambiente de desenvolvimento.

6.3.2 – Validação em SE

Em uma fase mais avançada na busca por falhas nos SE, a validação busca identificar erros ou inadequações na operação do SE. Geralmente, a validação é realizada submetendo o SE a *testes de campo*, em condições fora do ambiente ao qual é desenvolvido.

O detalhamento das conclusões da validação pode ter a profundidade tanto apenas superficial, resultando em parâmetros *qualitativos* do desempenho do SE, quanto pode ser aprofundada, podendo descrever detalhes *quantitativos* das falhas cometidas. Em ambos os casos, os resultados devem ser devidamente descritos para que se possa obter maiores detalhes sobre a validação.

Gonzalez e Dankel (1993) sugerem como escala de valores para os diversos níveis de desempenho do SE, desde valores simples, quantificáveis através de resposta como *sim* e *não*, até a utilização de escalas de valores de 0 a 10. Para evitar problemas durante a validação do SE, esta deve ser assistida pelo EC ou deve-se utilizar uma interface que seja amigável com o validador e que não gere dúvidas.

Os validadores de um SE devem formar um público heterogêneo, referente ao conhecimento prévio sobre o DC, podendo ser formado tanto por EH, que possuam conhecimento especializado suficiente para realizar a validação, como também por pessoas que possuem um conhecimento superficial sobre o DC, mas que formam o público alvo do SE.

A validação para poder ser considerada como diretriz sobre os caminhos do SE, deve atender a um público com ampla representatividade e diversidade. Gonzalez e Dankel (1993) sugerem a aplicação de métodos estatísticos no auxílio dos resultados de testes que envolvam grandes amostragens de resultados. Segue na próxima seção a aplicação prática da validação do protótipo descrito no Capítulo 5.

6.4 – VV do protótipo

A aplicação do protótipo a testes de VV, submetendo-o a condições reais de operação, podem propiciar o levantamento de dados que constatem a adequação do mesmo às necessidades dos *clientes* estabelecidas inicialmente no Capítulo 5, bem como identificar pontos fracos que devem ser melhorados, fechando assim o ciclo de desenvolvimento *incremental*, o qual pode ser novamente iniciado para a realização de novos trabalhos sobre o protótipo, sendo contudo norteados pelos resultados da VV.

6.4.1 – Verificação do protótipo com *teste de caso*

A submissão do protótipo a *testes de casos* busca submeter o mesmo a situações previamente estabelecidas, em que já se tem o conhecimento do resultado, o qual pode ser comparado com o resultado que o protótipo oferece como saída.

Como apresentado inicialmente neste capítulo, o *teste de caso* não apresenta alta eficiência como método de constatação da adequação do SE, porém com algumas limitações pode-se obter com ele:

- Identificação de erros grosseiros;
- Verificar erros de interface;
- Verificar erros de operação ou *bugs*.

6.4.1.1 – Definição dos dados do *teste de caso*

O *teste de caso* adotado no protótipo visa prioritariamente constatar as adequações das saídas do protótipo em relação às funcionalidades que os métodos pneumáticos seqüenciais oferecem. Como a finalidade principal deste protótipo computacional é oferecer uma solução pneumática para projetos do tipo seqüencial, e sendo o projeto realizado apenas com base no procedimento seqüencial desejado, independente da finalidade a que se destina esta seqüência de movimentações, a proposta de *teste de caso* deve ter atenção prioritária para tentar descrever uma grande variedade de situações de projeto possíveis de serem solicitados.

Portanto, o *teste de caso* é baseado em cargas mecânicas do tipo *sem controle*. São consideradas apenas duas cargas mecânicas, sendo uma delas na posição recuada no início do ciclo seqüencial e outra avançada. Os passos são estabelecidos como sendo três, havendo em um destes o acionamento simultâneo das duas cargas. Segue abaixo o *diagrama trajeto-passo* da seqüência de movimentação das cargas mecânicas que o protótipo oferece, representando exatamente a funcionalidade de entrada, indicando a adequação do mesmo na entrada dos dados.

Diagrama Trajeto Passo - Projeto Sequencial

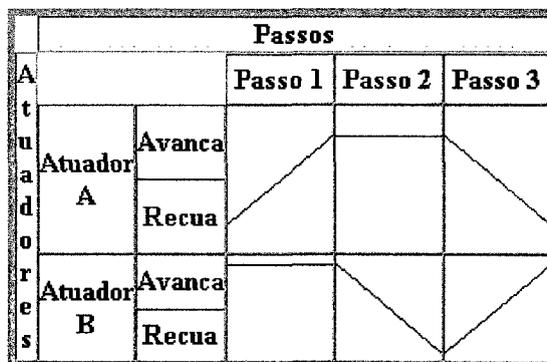


Figura 6.1 – *Diagrama trajeto-passo* do teste de caso.

Prosseguindo o projeto com estes dados de entrada, inicialmente é selecionado o método passo-a-passo da *pneumática pura* (PP), cujo o esquema de saída oferecido pelo protótipo é apresentado na Figura 6.2.

Projeto Pneumatico - Metodo Passo a Passo

System 1

Lista de Circuitos

Circuito	Nome	Grupo
1	Circuit 2	circuito_de_carga_sequencial
2	Circuit 1	circuito_de_carga_sequencial
3	Circuit 6	circuito_passo_a_passo
4	Circuit 5	circuito_passo_a_passo
5	Circuit 4	circuito_passo_a_passo
6	Circuit 3	circuito_de_gerenciamento_passo_a_passo

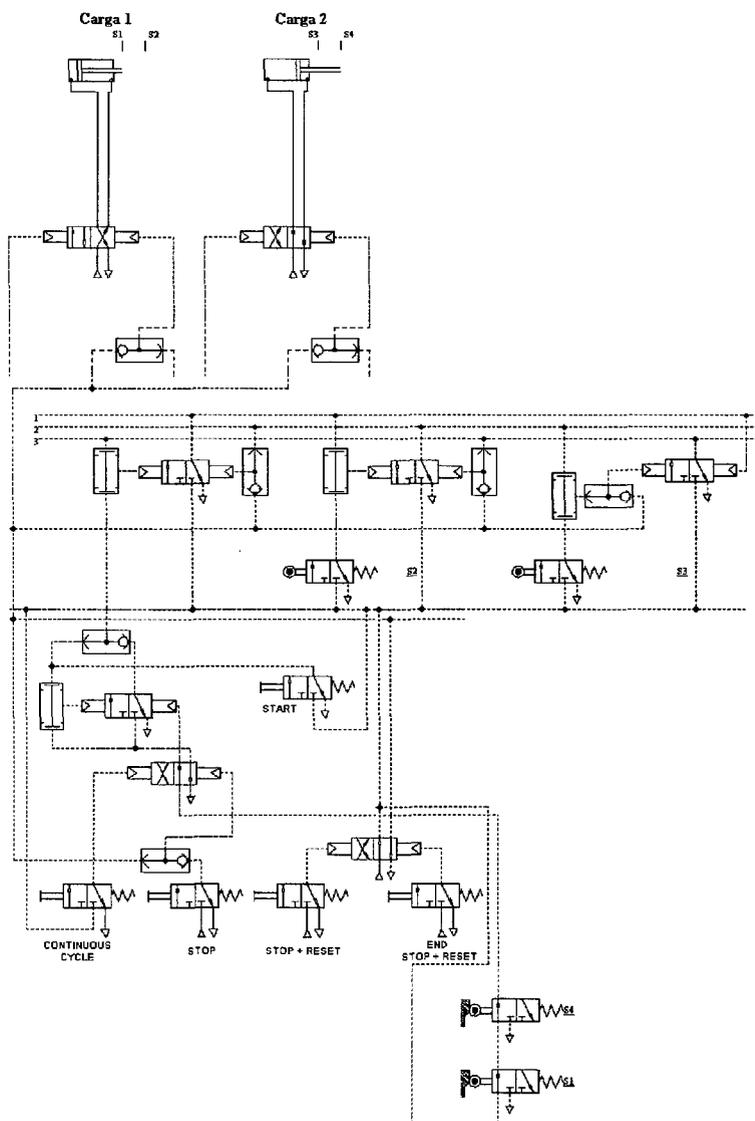


Figura 6.2 – Esquema do método passo a passo para o teste de caso.

Para o mesmo teste de caso, quando selecionado o método seqüência máxima, o protótipo oferece como saída o esquema apresentado na Figura 6.3.

Projeto Pneumatico - Metodo Sequencia Maxima
System 1

Lista de Circuitos

Circuito	Nome	Grupo
1	Circuito 2	circuito de carga sequencial
2	Circuito 1	circuito de carga sequencial
3	Circuito 17	circuito de potencia electrico
4	Circuito 16	circuito de potencia electrico
5	Circuito 15	circuito de potencia electrico
6	Circuito 14	circuito de accionamento electrico
7	Circuito 13	circuito de accionamento electrico
8	Circuito 12	circuito de accionamento electrico
9	Circuito 11	circuito de accionamento electrico
10	Circuito 10	circuito de memoria electrica
11	Circuito 9	circuito de memoria electrica
12	Circuito 8	circuito de memoria electrica
13	Circuito 7	circuito de gerenciamiento electrico
14	Circuito 6	circuito de gerenciamiento electrico
15	Circuito 5	circuito de gerenciamiento electrico
16	Circuito 4	circuito de gerenciamiento electrico
17	Circuito 3	circuito de gerenciamiento electrico

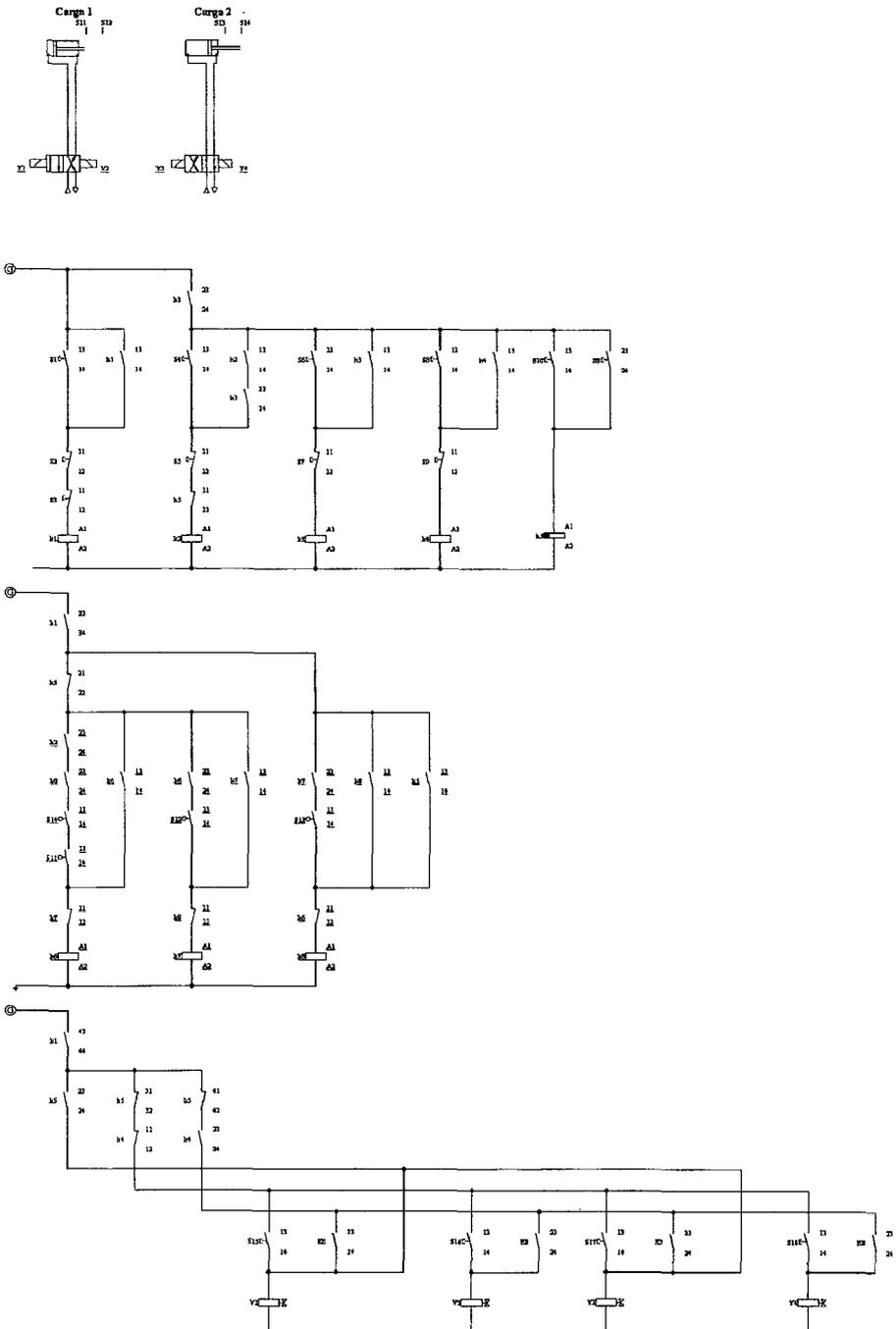


Figura 6.3 - Esquema do método seqüência máxima para o teste de caso.

Durante a evolução desta breve verificação, o protótipo não apresenta nenhum tipo de *bug* ao longo da operação. Apesar da representação nas figuras anteriores ilustrarem os sistemas pneumáticos na íntegra, como foi utilizada a estrutura de hipertexto, é possível obter detalhes de partes do sistema, que são os circuitos. Havendo a necessidade de maior profundidade, pode-se prosseguir a pesquisa chegando aos componentes e aos elementos dos componentes, caso este dos componentes elétricos.

6.4.1.2 – Montagem do teste de caso no *Automation Studio*

Com os esquemas que o protótipo oferece como saída para o projeto, pode-se realizar a montagem dos mesmos no *Automation Studio* (AS), que é um *software* que possibilita a montagem de esquemas hidráulicos, pneumáticos, elétricos e eletrônicos, o qual é utilizado por alunos dos cursos de tecnologia da *Sociedade de assistência aos trabalhadores do carvão* (SATC) nas disciplinas de hidráulica e pneumática, onde estão disponíveis algumas versões registradas. Com o AS é possível realizar a verificação da consistência dos dados do projeto que o protótipo oferece como saída. Segue na Figura 6.4 o esquema seqüencial pneumático puro do método passo-a-passo, montado no AS.

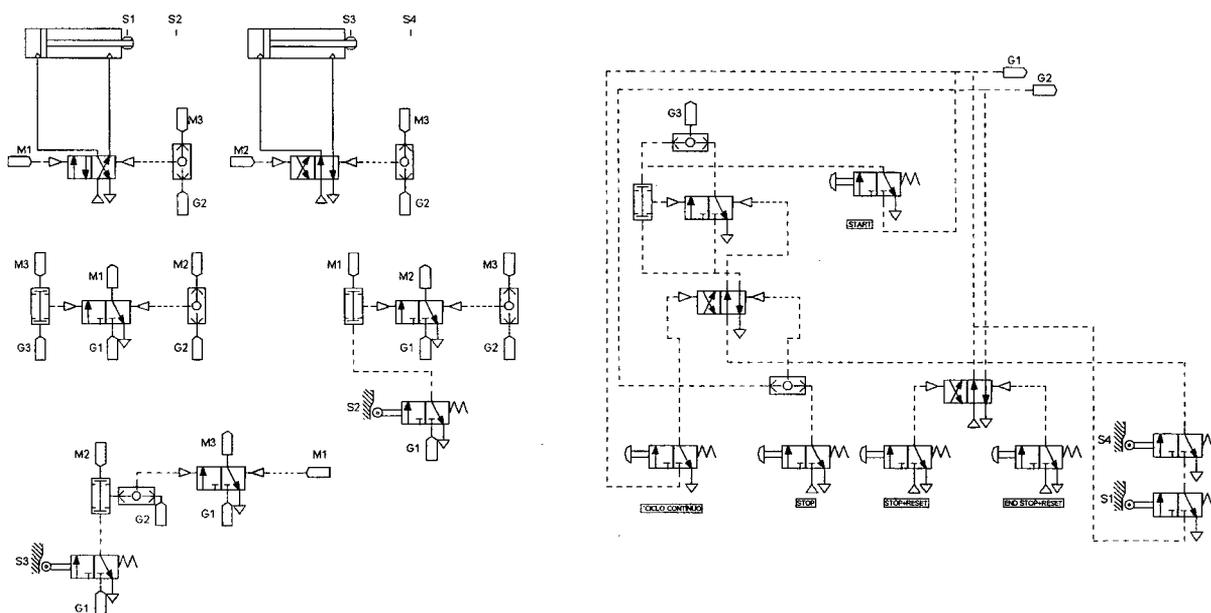


Figura 6.4 – Teste de caso do método passo-a-passo montado no AS.

Analisando o resultado da operação do esquema montado no AS, conclui-se a perfeita adequação dos resultados do protótipo ao objetivo esperado.

Prosseguindo os *testes de caso*, é realizado o mesmo procedimento para a esquema

seqüencial do método seqüência máxima para a eletropneumática, no qual obteve-se o resultado apresentado na Figura 6.5.

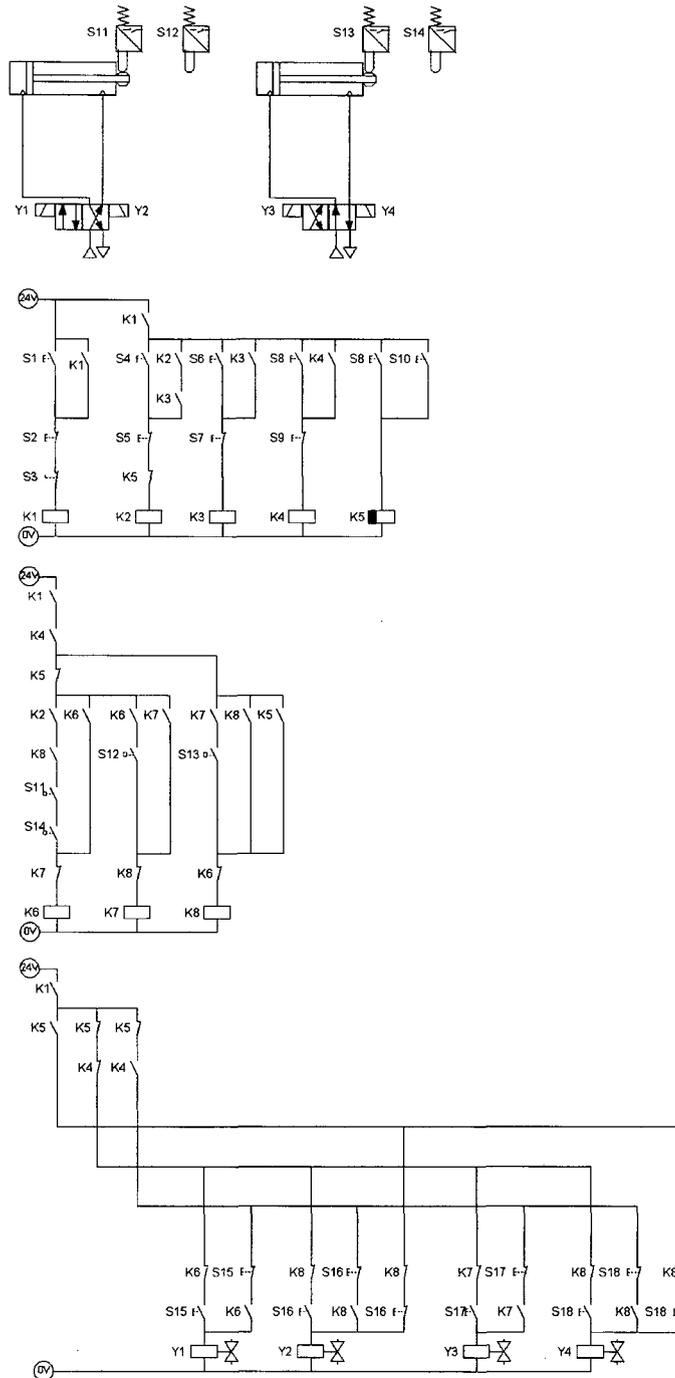


Figura 6.5 – Teste de caso do método seqüência máxima montado no AS.

O esquema da Figura 6.5 montado no AS na realidade representa o projeto que atende as funcionalidades inicialmente desejadas, não sendo portanto uma cópia fiel da saída do protótipo, conforme apresentado na Figura 6.3. Esta diferença entre os esquemas deve-se a

constatação de erros de operação que os projetos fornecidos pelo protótipo possuem, sendo o esquema da Figura 6.5 a adequação para atender os requisitos de operação. Esta falha ocorre no circuito elétrico, na parte responsável pela alimentação dos solenóides, onde inconsistências no isolamento de partes do circuito não são satisfeitas, sendo preciso a inclusão de novos comandos com a anulação dos demais acionamentos. Com a identificação deste erro, o protótipo foi aprimorado para oferecer as saídas conforme as funcionalidades esperadas.

6.4.1.3 – Montagem dos esquemas em bancada experimental

Complementando os ciclos de verificação e ampliando as possibilidades de teste do protótipo, a verificação no AS não pode ser considerada conclusiva, afinal o próprio AS, que é também um *software*, não está isento de possíveis erros. Nestas condições, a montagem dos esquemas propostos pelo protótipo em condições reais de operação ou em situações que representem parcialmente o projeto proposto pode propiciar um maior refinamento das saídas.

Nesta condição, as bancadas didáticas podem ser aplicadas, pois possuem uma grande variedade de componentes, possibilitando a montagem de diversos tipos de projetos. As bancadas utilizadas na verificação do protótipo fazem parte do material laboratorial da SATC.

A montagem é realizada parcialmente devido à grande quantidade de componentes necessários, não sendo possível construir todo o projeto. Na opção inicial da montagem, dá-se preferência pelos esquemas do método passo-a-passo, proposto pelo protótipo para o *teste de caso*, pois para este domínio da pneumática tem-se a disposição uma maior quantidade de componentes, apesar de não satisfazerem a totalidade de componentes desejados.

Com estas limitações, é dedicada maior atenção sobre o circuito de gerenciamento do projeto pneumático, que realiza a interface entre o usuário e o equipamento projetado. O resultado final da montagem é apresentado na Figura 6.6.

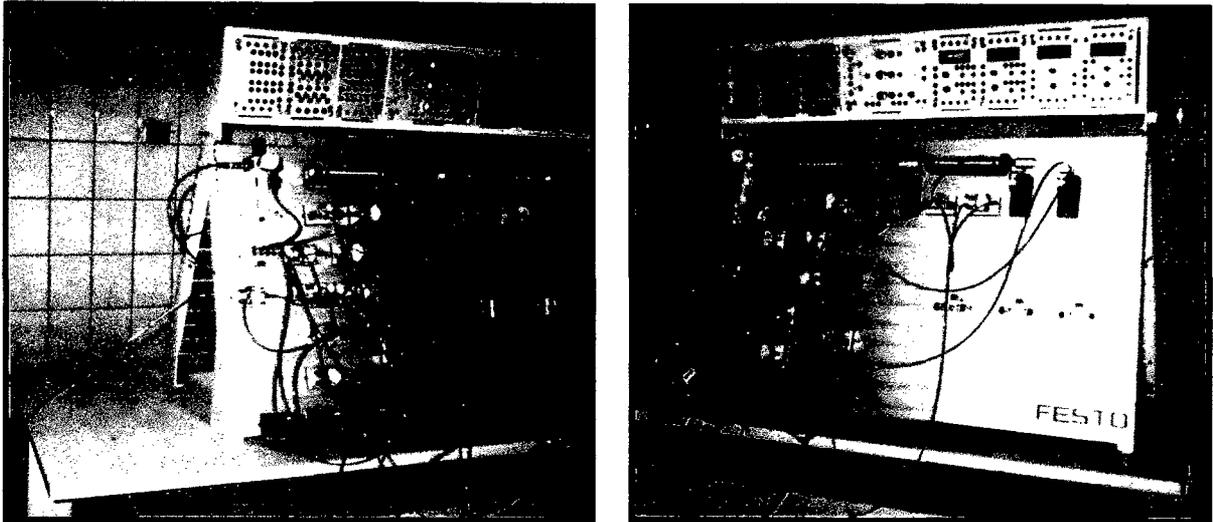


Figura 6.6 – Teste de caso do método passo-a-passo montado na bancada didática.

Analisando a funcionalidade da montagem, é constatado que o circuito realiza os procedimentos conforme é esperado. A verificação do esquema seqüência máxima na bancada didática não é realizada pois o projeto exige muitos componentes, não sendo possível realizar uma montagem simplificada das funções mais importantes, não sendo portanto possível a sua verificação.

Analisando a totalidade da verificação do protótipo, desde a implementação no AS até a montagem na bancada didática, são constatadas algumas deficiências nas saídas do projeto seqüencial pelo método seqüência máxima. No restante, o protótipo apresenta desempenho condizente com o resultado esperado. Apesar da constatação destas falhas, o protótipo é submetido a testes de validação, que servem para obter resultados mais aprofundados sobre o mesmo, através da submissão do protótipo a condições de *teste de campo*, em condições reais de operação junto aos seus usuários.

6.4.2 – Validação do protótipo

Em uma etapa mais aprofundada de testes sobre o protótipo, a validação visa submeter o protótipo a pessoas externas ao projeto do mesmo, para validar a sua utilização. Para coletar as opiniões dos validadores, é utilizada uma *ficha de avaliação*, que para facilitar a análise dos resultados é compartimentada e estruturada conforme os tipos de informações que deseja-se adquirir.

Como detalhes na *ficha de avaliação*, são compartimentados os seguintes parâmetros:

- Perfil do avaliador: onde pode-se analisar que profundidade de conhecimento o usuário possui, para que possa ser possível classificar a sua avaliação como sendo de um EH ou de um usuário não especialista;
- Interface de entrada, que indica a interação do protótipo com o usuário na entrada de dados sobre os projetos pneumáticos;
- Interface de saída, que descreve a facilidade com que o protótipo interage com o usuário quando das saídas do projeto;
- Base de conhecimento, que descreve o conhecimento especializado em projetos pneumáticos que o protótipo possui.

Para simplificar a avaliação do protótipo, a estrutura da *ficha de avaliação* pode ser acessada tanto através de cópia impressa como apresentado na Figura 6.7, ou através do próprio ambiente do protótipo como apresentado na Figura 6.8, sendo o resultado possível de ser destinado ao EC através de correio eletrônico, ampliando e possibilitando que todos os usuário ofereçam a sua opinião sobre o protótipo facilmente.

Perfil do avaliador – Dados pessoais

Formação nível técnico	Sim		Não		Em andamento	
Formação nível superior	Sim		Não		Em andamento	
Cursos em pneumática	Número de anos					
Trabalho em pneumática	Número de anos					
Trabalha atualmente com pneumática	Sim		Não			

Escala de valores

7 – Ótimo	6 – Muito Bom	5 – Bom	4 – Regular	3 – Ruim	2 – Muito Ruim	1 – Péssima
-----------	---------------	---------	-------------	----------	----------------	-------------

Interface de entrada	
Facilidade de entrada dos dados	
Interatividade de entrada dos dados	
Conhecimento exigido na entrada dos dados	
Interface de saída	
Facilidade na leitura dos dados de saída	
Interatividade dos dados de saída	
Apresentação dos sistemas, etc.	
Velocidade na obtenção das saídas	
Base de conhecimento	
Conhecimento inserido no programa	
Coerência entre as entradas e as saídas	
Capacidade de oferecer soluções alternativas	
Aplicação do programa em projetos reais	

Recomendações, sugestões e comentários adicionais:

Figura 6.7 – Ficha de avaliação do protótipo no formato impresso.

Figura 6.8 – Ficha de avaliação inserida no protótipo.

A escala de valores adotada na determinação do desempenho do protótipo sobre os tópicos existentes na *ficha de avaliação* é subdividida em 7 níveis. Para que estes níveis descrevam realmente a opinião do usuário, estes são acompanhados por termos como bom, regular, péssimo entre outros, para simplificar a atuação do avaliador e obter a real expressão de cada conceitos que o avaliador tenha sobre o protótipo.

Devido ao perfil de especialidade em pneumática ser diferenciado entre os validadores, estes são classificados em EH e em usuários não especialistas. Esta divisão propicia que detalhes sobre a avaliação do protótipo, específicos sobre um determinado tópico sejam analisados por pessoas capacitadas para a função.

Portanto, para a avaliação dos tópicos referentes à interface de entrada e saída, os usuários que não são classificados como especialistas são as pessoas indicadas, pois formam o público alvo do protótipo, sendo este o nicho que deseja-se predominantemente atender.

Apesar das interfaces do protótipo poderem ser avaliadas também por EH, como estas pessoas a princípio não formam o principal público alvo ao qual o protótipo busca atender, possíveis deficiências na interface do protótipo podem ser amenizadas pelo prévio conhecimento especializado que o validador possui, podendo assim ocultar falhas cometidas. Por este motivo, os resultados da validação pelos EH são priorizados com relação à BC formada. O resultado da avaliação do protótipo, segundo divisão conforme especialidade dos avaliadores é descrito nas próximas subseções.

6.4.2.1 – Usuários não especialistas

O auxílio na validação do protótipo é obtido com a cooperação dos estudantes dos cursos de *Tecnologia em Automação Industrial* e *Eletromecânica* da SATC, conforme descrito no Capítulo 5, podendo estes serem classificados como validadores não especialistas, por estarem ainda em formação.

Na totalidade, a validação nesta etapa teve a colaboração de 29 pessoas, subdivididas em 2 turmas. Apesar de todos estarem ainda em fase de formação, 69% declararam ter formação técnica, e 3%, 14% e 3% declaram respectivamente terem 1, 2 e 3 anos de experiência profissional em pneumática.

A validação do protótipo com os usuários estudantes seguiu-se inicialmente com uma breve apresentação do trabalho, o contexto ao qual está inserido e o que os estudantes poderiam fazer para auxiliar no projeto.

Com o consenso de todos, o protótipo em si foi inicialmente apresentado, sendo primeiramente utilizada a versão que possui a interface padrão do CLIPS, via linha de comando, para apenas posteriormente serem feitas as análises sobre a interface que utiliza a orientação a eventos, para com isso evitar a grande recusa que poderia ocorrer caso a versão CLIPS padrão fosse utilizada por último, podendo assim distorcer os resultados não sendo assim possível obter o real acréscimo de qualidade na transição destas versões. A comparação direta entre estas versões do protótipo são descritos no Apêndice D.

Para submeter o protótipo a testes pelos usuários, inicialmente ambas as versões foram apresentadas as opções existentes nos menus e posteriormente realizado um projeto que segue a estrutura dos dados do *teste de caso*, adotado na verificação. Com isto foi possível rapidamente treinar os usuário para a entrada dos dados do projeto que é desejado realizar no protótipo. Para cada uma das versões, os estudantes tiveram o equivalente a 2 horas-aula para testa-los. Sendo este tempo suficiente para uma breve apresentação inicial do protótipo, no tempo restante os estudantes tiveram a possibilidade de testarem livremente o protótipo e realizar a avaliação do mesmo na *ficha de avaliação*. Na totalidade, os estudantes tiveram 4 horas-aula de atuação com o protótipo.

Apesar de haver uma relação do tipo professor/aluno entre o EC e os validadores, o que poderia conduzir à uma avaliação tendenciosa por talvez haver uma possível intimidação dos estudantes, destaca-se que na *ficha de avaliação* nenhum dos campos a serem preenchidos descrevem dados estritamente pessoais e únicos como *nome*.

Segue na Tabela 6.1 os resultados obtidos na validação do protótipo, sendo listados apenas a versão com interface orientada a eventos.

Tabela 6.1 – Validação do protótipo com interface orientada a eventos por estudantes.

	Notas [%]							Nota média [1 a 10]
	7 – Ótimo	6 – Muito Bom	5 - Bom	4 - Regular	3 - Ruim	2 – Muito Ruim	1 - Pêssimo	
Interface de entrada								
Facilidade de entrada dos dados	31,0	44,8	17,2	6,9	0,0	0,0	0,0	8,6
Interatividade de entrada dos dados	34,5	48,3	6,9	10,3	0,0	0,0	0,0	8,7
Conhecimento exigido na entrada dos dados	20,7	55,2	17,2	6,9	0,0	0,0	0,0	8,4
Interface de saída								
Facilidade na leitura dos dados de saída	24,1	58,6	17,2	0,0	0,0	0,0	0,0	8,7
Interatividade dos dados de saída	31,0	44,8	24,1	0,0	0,0	0,0	0,0	8,7
Apresentação dos sistemas, etc.	37,9	41,4	6,9	13,8	0,0	0,0	0,0	8,6
Velocidade na obtenção das saídas	41,4	41,4	13,8	3,4	0,0	0,0	0,0	8,9
Base de conhecimento								
Conhecimento inserido no programa	34,5	44,8	17,2	3,4	0,0	0,0	0,0	8,7
Coerência entre as entradas e as saídas	27,6	37,9	34,5	0,0	0,0	0,0	0,0	8,5
Capacidade de oferecer soluções alternativas	13,8	41,4	31,0	13,8	0,0	0,0	0,0	7,9
Aplicação do programa em projetos reais	34,5	37,9	24,1	3,4	0,0	0,0	0,0	8,6
Média global								8,5

Na análise dos dados da Tabela 6.1, evidencia-se que o protótipo apresenta um bom desempenho, conforme a perspectiva dos usuários não especialistas, sendo o resultado da avaliação global representada pela nota 8,5 em uma escala de valores ponderados até 10. Pela perspectiva da dispersão dos resultados da validação, constata-se que os valores flutuam entre as notas absolutas 4 e 7, segundo os conceitos regular e ótimo.

Dentro de uma margem aceitável de resultados a serem alcançados, pois para qualquer avaliação que seja realizada é esperado que não seja possível atingir a totalidade do índice de satisfação máximo, constata-se que o protótipo apresenta resultados segundo a perspectiva dos validadores como sendo relativamente adequados à utilização.

Deve-se destacar os pontos fracos detectados que devem ser enfatizados em futuros aprimoramentos, como a *capacidade de oferecer soluções alternativas*, que obteve o pior resultado na avaliação, o que pode ser solucionado em futuras implementações com a ampliação da BC.

Podem ser consideradas também as recomendações complementares sugeridas pelos usuários, compiladas a seguir:

- Animação interativa com o usuário, assim como a realizada no AS;
- Proporcionar a seleção do modelo de componente;
- Aumentar a área de visualização dos esquemas.

6.4.2.2 – Usuários especialistas

Em uma etapa de validação mais aprofundada, existe a necessidade da submissão do protótipo a validação por EH que atuem na área do DC. Ao contrário da validação realizada pelos potenciais usuários dos *software*, em que a atuação é concentrada sobre a interação do mesmo com o seu público alvo, a validação com EH busca obter as deficiências que o *software* possui, referentes ao conhecimento inserido na BC, a forma com que o DC é abordado e a devida correlação entre as entradas e as saídas.

A tarefa de validação do protótipo por especialistas é auxiliada pelo professor Dr. Eng. Victor Juliano De Negri, pesquisador e membro do *Laboratório de Sistemas Hidráulicos e Pneumáticos* (LASHIP), do Departamento de Engenharia Mecânica, da *Universidade Federal de Santa Catarina* (UFSC).

A seguir tem-se o resultado da validação do protótipo, com os respectivos comentários do validador, seguido pelos comentários e justificativas do EC sobre o protótipo, utilizando a formatação em itálico na diferenciação.

Facilidade de entrada dos dados: Bom

A entrada de dados é bastante acessível para quem já projetou sistemas pneumáticos. Creio que apresentação de um sistema exemplo com as partes constituintes e os pontos onde o usuário será requisitado a inserir especificações seria fundamental. Como sistema especialista, seria importante contextualizar os sistemas pneumáticos no âmbito de sistemas de automação destacando a existência de sistemas de atuação (onde aparecem os circuitos de atuação) e de um subsistema de processamento de informações (onde surgem as realizações em pneumática pura, eletropneumática (em desuso) e programação).

Referente a possibilidade de inserção de um sistema exemplo, a princípio, o objetivo principal do protótipo destina-se a realização de projetos do tipo seqüenciais, sendo o ponto de partida do projeto apenas a descrição da seqüência com que deseja-se movimentar as cargas mecânicas. Como a forma com que pode-se solucionar um projeto seqüencial é muito

diversificada, podendo ser utilizadas soluções hidráulicas, elétricas, mecânicas ou pneumáticas, o conhecimento prévio do usuário sobre estes domínios deve ser suprido pelo software, não sendo assim necessário a preocupação por parte do usuário com relação aonde estas informações são necessárias. O que pode ser aprimorado é informar após o projeto ser concluído onde os dados informados pelo usuário foram utilizados pelo SE, que pode ser feito justamente como a proposta do EH de explicar as parte constituintes do projeto proposto.

Apesar de em projetos mais complexos as soluções pneumáticas pura e eletropneumáticas estarem em desuso, é optado pela implementação previa destes princípios de solução para a prévia busca do campo de atuação do protótipo, que um ciclo de melhoria continua pode ser expandido para outros DC.

Melhorias na descrição de detalhes sobre projetos de automação em pneumática podem também serem anexados futuramente ao help já existente no protótipo, podendo ser utilizadas partes deste trabalho.

O projeto pelos métodos seqüência máxima e passo a passo tradicional poderia ser substituído pelo método passo a passo generalizado, que conduz às três soluções alternativas: Pneumática pura, eletropneumática e software (CLP). Para esta seleção seriam requisitados outras informações como custo, capacidade técnica da equipe de manutenção em uma área ou outra, meio ambiente etc.

O contexto com que os projetos são realizados é justamente baseado na inserção prévia dos requisitos que o projeto deve atender, para posteriormente passar para a etapa de busca pelas soluções, tanto através dos detalhes comportamentais a serem atendidos, como com relação a detalhes qualitativos esperados sobre o protótipo, conforme também implementado por Silva (1998). Porém, por limitação de tempo, o direcionamento dos projetos pneumáticos segundo detalhes qualitativos são considerados como aprimoramentos futuras do protótipo.

O diagrama trajeto-passo é bastante limitado para a especificação comportamental de sistemas reais. Por exemplo, este perde a facilidade de representação em sistemas com passos alternativos ou loops e também quando são empregados vários outros sensores além daqueles que detectam fim de curso de cilindros.

Como apresentado no texto, algumas limitações foram estabelecidas sobre o protótipo para possibilitar a busca por resultados sobre a aplicação dos SE em projetos pneumáticos.

Futuramente descrições comportamentais adicionais podem ser oferecidas no protótipo.

É necessário possibilitar ao usuário a inserção de outras entradas como escolha de condições de operação (automática, manual, passo-a-passo etc), sensores complementares etc.

Devido à grande complexidade existente entre coletar os requisitos do projeto, agrupar os componentes que formam os circuitos, que por sua vez formam os sistemas, de tal forma que sejam atendidos os requisitos do projeto, e que por fim sejam apresentados no formato gráfico, com uma diagramação que seja possível de ser acessada pelo usuário, optou-se por inicialmente oferecer as funções mais elementares, que futuramente em ampliações do protótipo podem suprir requisitos específicos de projeto.

Interatividade de entrada dos dados: Muito bom

Dentre os dados hoje inseridos, a associação entre estes parece adequada.

Conhecimento exigido na entrada dos dados: Muito bom

Facilidade de leitura dos dados de saída: Muito bom

Interatividade dos dados de saída: Muito bom

Apresentação dos sistemas, etc.: Regular

Para dar flexibilidade ao projeto, é importante que os circuitos gerados possam ser convertidos para algum software de desenho como AutoCad ou Visio.

É necessária uma análise crítica de todo o sistema. Observando o circuito 15 - circuito de gerenciamento elétrico e a descrição do circuito apresentada percebe-se que esta última é mais abrangente que o diagrama de circuito apresentado.

Como os circuitos que formam o bloco de componentes responsáveis pelo gerenciamento dos projetos do método seqüência máxima possuem grande interação entre si, opta-se não somente pela descrição do mesmo, mas como também dos demais circuitos, facilitando a análise do conjunto. A descrição detalhada específica de cada circuito pode porém, também ser realizada e disponibilizada em versões futuras do protótipo.

A notação para identificação dos componentes não está de acordo com a ISO 1219-2.

Velocidade na obtenção das saídas: Ótimo

Conhecimento inserido no programa: Bom

Coerência entre as entradas e as saídas: Muito bom

Dentre as entradas e saídas fornecidas na versão atual, há necessidade de evidenciar a utilização dos dados qualitativos.

O EH foi capaz de identificar este detalhe, pois a entrada dos dados qualitativos esperados sobre o projeto e a utilização destes como parâmetros no projeto ainda não estão implementados.

Capacidade de oferecer soluções alternativas: Ruim

Não há suporte ao projetista para a seleção do sistema pneumático dentre as soluções propostas.

Mesmo o protótipo sendo capaz de oferecer diversos sistemas como propostas de solução, estas ainda não estão classificadas conforme prioridade de aplicação justamente pela não inserção dos parâmetros qualitativos do projeto.

Aplicação do programa em projetos reais: Muito bom

O conceito foi atribuído considerando a potencialidade do sistema em desenvolvimento.

Creio que a inserção de regras para escolha do método de projeto e também do princípio de solução a ser adotado gerará um software de grande utilidade para empresas que desenvolvem equipamentos e processos com automação pneumática. Desta forma estará sendo aliada à criação do diagrama, o apoio à seleção da forma mais adequada de implementação.

Capítulo 7 – Conclusões e recomendações futuras

7.1 – Conclusões sobre o trabalho

Apesar de que nos objetivos inicialmente estabelecidos buscava-se a construção de um *software* que realiza-se projetos de sistemas pneumáticos, já era previamente esperado algumas *limitações* de atuação bem como possíveis *deficiências* de operação.

As *deficiências* de operação podem ser consideradas, muitas vezes, como uma característica inerente de qualquer sistema, seja ele de qualquer domínio, pois contínuas evoluções são sempre necessárias. No contexto do protótipo, a definição das *deficiências* pode ser considerada muito subjetiva, como constata-se nos resultados da pesquisa com usuários não especialistas, o qual observa-se uma dispersão nos dados, atingindo conceitos entre regular e ótimo e deficiente para as mesmas condições de teste, evidenciando previamente a impossibilidade de atender a todas as possíveis necessidades dos usuários.

Para a análise das *limitações* do protótipo, que utiliza a técnica de *sistemas especialistas* (SE), pode-se utilizar a definição de Dym e Levitt (1991, p. 364) que descrevem “os SE não podem fazer o impossível, como exemplo curar o câncer [...] os SE não podem fazer o extraordinário, como exemplo obter dinheiro continuamente na bolsa de valores”, evidenciando assim as *limitações* existentes em qualquer SE, não impedindo contudo que melhorias sejam realizadas para minimizar estas *limitações*.

Isto é condizente com os objetivos de evolução contínua do protótipo adotado nas metodologias incrementais de desenvolvimento. Destaca-se que no processo incremental as melhorias podem estender-se também a ampliação do campo de abrangência da *base de conhecimento* (BC), que no caso de projetos de sistemas pneumáticos, o protótipo foi capaz de atingir apenas uma pequena parcela da totalidade de opções de aplicação da pneumática.

Considerando a totalidade do trabalho realizado, em conjunto com os resultados obtidos das validações, pode-se considerar que o protótipo atende de modo satisfatório aos objetivos estabelecidos, o que pode ser considerado um bom resultado, pois no trabalho não ocorreram novos ciclos de implementação para a correção das falhas diagnosticadas, devido à falta de tempo para o prosseguimento do mesmo.

No contexto de projetos de sistemas pneumáticos, pode-se constatar que as ferramentas computacionais, em destaque os SE, são de grande valia, inserindo o conhecimento especializado nas etapas iniciais do projeto, situação esta extremamente benéfica para a garantia da obtenção de soluções consistentes nas etapas iniciais do projeto, sendo este o diferencial.

Analisando sob a perspectiva dos elementos existentes externamente à fronteira do protótipo, o mesmo demonstra-se suficientemente preparado para poder ser utilizado por usuários com diversos níveis de especialidade, sendo contudo limitado a exigência de um prévio conhecimento básico sobre pneumática, como simbologias, características de funcionamento dos componentes pneumáticos, entre outras informações.

Isto porém não impossibilita que com a atual configuração o protótipo não seja possível de ser utilizado como complemento a atividades didático-pedagógica, como no caso do *teste de campo* com os alunos, pois este foi inserido no conteúdo previamente estabelecido de projetos seqüenciais existente nas disciplinas curriculares.

A constatação da adequação da técnica de SE em aplicações de projeto, como o caso dos sistemas pneumáticos demonstra-se ser adequada, pois nas tomadas de decisão sobre o direcionamento do projeto, tem-se o emprego de conhecimentos de origem heurística, sendo o modelo dos SE adequado ao tipo de processamento de informações que é feito nos projetos de sistemas pneumáticos.

Sobre as etapas de evolução no desenvolvimento do protótipo, a análise de viabilidade inicialmente conduzida foi extremamente benéfica com a constatação, ainda nas fases iniciais, da adequação das possíveis métricas a serem utilizadas na seqüência de elaboração do mesmo, sendo algumas já evidenciadas a partir de trabalhos anteriores, como o realizado por Silva (1998), tornando assim possível obter bons resultados, apesar de ter sido feito apenas um ciclo de implementação e teste.

Além dos benefícios dos SE em projetos, o formato adotado na representação do conhecimento também teve influência no resultado final obtido. Com a aplicação das técnicas de *orientação a objetos* (OO), pode-se simplificar o trabalho de *engenharia do conhecimento*, que para o usuário pode ser imperceptível, pois o mesmo não tem interação direta com este formato. Porém, a qualidade do resultado final obtido passa a ser algo perceptível pelo usuário, através de simplificações decorrentes da melhor organização, inserção e gerenciamento do conhecimento dentro da BC, proporcionando resultados com menor tempo e possibilitando ampliações decorrentes da modularidade que a OO proporciona.

No domínio do conhecimento de projetos de sistemas pneumáticos, o formato da OO pode simplificar na formação da estrutura de inter-relacionamento entre os itens manipulados, sendo processados como *objetos*, propiciando a adoção das características inerentes a OO, como *abstração*, *encapsulamento* entre outros, na composição da dependência existente entre as parte que formam os projetos, como dos circuitos em relação ao sistema, dos componentes em relação aos circuitos e dos elementos que formam os componentes, sendo este último caso específico aos componentes elétricos.

Com a diversidade nas possibilidades de projetos possíveis de serem realizados, a OO possibilitou a reutilização de *classes* destinadas a tipos específicos de aplicações, como nos circuitos de carga mecânica, que devido à *especialização* exigida neste tipo de circuito, foi possível adotar a *generalização* das cargas mecânicas utilizadas nos projetos de *simples comando*, com a inserção de novos *atributos* específicos ao circuito que se deseja modelar, como no caso dos circuitos de carga mecânica para projetos seqüenciais. Este mesmo procedimento pode ser utilizado em outras correlações entre *classes*.

Mesmo não sendo possível oferecer todas as funções previamente desejadas, como o caso da inserção de parâmetros qualitativos no projeto, todos estes procedimentos foram construídos de forma a poderem ser inclusos com grande facilidade de implementação, pois todos estes dados já estão parametrizados, restando apenas a adaptação dos mesmos ao *domínio de conhecimento* (DC).

As formas com que o usuário pode interagir com o protótipo revela que, mesmo com um SE que possua um BC com grande *amplitude* de atuação e com grande *profundidade* de especialização, a interface pode limitar a forma de comunicação entre o usuário e a BC. Avaliando em conjunto os dados das versões do protótipo, pode-se evidenciar um pequeno acréscimo na aceitação da utilização da versão com interface orientada a eventos por parte dos usuários. Isto ocorre pois o protótipo que utiliza interface CLIPS padrão já possui interface gráfica na saída, sendo portanto esta versão satisfazendo de forma condizente os recursos básicos de interface desejada.

7.2 – Recomendações futuras

A área de pneumática demonstra ser amplamente aplicada e as possibilidades de estudo são diversas, podendo ser realizados estudos sobre diversas perspectivas, a partir do trabalho até então realizado sobre o protótipo, dentre as quais cita-se:

- Inserção de módulos na BC destinados à função didático-pedagógica, em que seja possível realizar o treinamento dos potenciais usuários até atingirem um nível mínimo para poderem compreender a operação do estágio atual que o protótipo possui. Esta ampliação pode ser estruturada utilizando o formato hipertexto, segundo o padrão HTML ou então através da estrutura de *help*, existente em *software* diversos, destinado ao nivelamento dos usuários com relação a um determinado assunto. Para o caso do usuário que esteja em treinamento, pode-se também ampliar a BC para avaliar o desempenho do seu aprendizado e, com isso, saber as suas deficiências e poder direcionar o trabalho de formação;
- Possibilitar a realização de projetos seqüenciais pelos métodos Cascata e seqüência mínima, e também propor soluções na pneutrônica;
- Aprimorar a BC para oferecer soluções derivadas dos métodos já conhecidos como, por exemplo, utilizando atuadores ou válvulas direcionais de comando com retorno por mola;
- Realizar o projeto de unidades de produção e distribuição de ar comprimido, habilitando assim a função previamente inserida no protótipo, porém não implementada;
- Expansão do protótipo para a realização de projetos combinatórios, propondo assim soluções para as aplicações pneumáticas, eletropneumáticas e pneutronicas;
- Expansão para os domínios de aplicação com a utilização de apenas componentes elétricos, além da busca pelo principio de solução mais conveniente a ser adotado, isto é, pneumático, elétrico ou hidráulico, tanto para a finalidade de geração de trabalho mecânico, bem como na coleta e processamento de sinais;

- Ampliação da BC para a atuação sobre o domínio de diagnóstico e correção de falhas em sistemas pneumáticos, sendo este campo de aplicação considerado como tendo muito mais conhecimento a ser explorado, quando comparado com campos como planejamento e projeto (DYM; LEVITT, 1991). Uma das limitações existentes na área do conhecimento de diagnóstico é a obtenção do conhecimento especializado sobre os esquemas pneumáticos, pois muitas vezes o conhecimento especializado é atuante sobre um sistema pneumático já previamente existente e conhecido. Para a construção de um SE para esta finalidade tem-se as limitações decorrentes da interpretação por parte do SE do esquema pneumático que deseja-se diagnosticar, sendo portanto na grande maioria das vezes construído um SE que atua especificamente sobre apenas um tipo de esquema. No caso deste protótipo isto é amenizado pois a formação do esquema pneumático em estudo já encontra-se inserido na BC, o que não inviabiliza a possibilidade de propor um SE para a detecção de falhas segundo perspectiva generalista sem que a BC seja conduzida a atuar sobre alguns esquemas pneumáticos previamente conhecidos; porém esta perspectiva generalista limitará a possibilidade de exploração detalhada dos conjuntos, atuando especificamente sobre os componentes;
- Assim como na finalidade de manutenção, destaque pode ser dado também para módulos destinados à montagem inicial dos esquemas pneumáticos, que além de requererem informações mais detalhadas, podem utilizar o conhecimento existente para o diagnóstico de operação incorreta, mesmo no início de operação do conjunto e com isso identificar possíveis falhas de montagem ou componentes previamente danificados;
- Possibilitar a interação com outros *software*, seja para oferecer os esquemas montados em formato que possa ser interpretado pelos *software* tradicionais de CAD, seja através da interação com *software* de simulação como adotado por Silva (1998) ou através de *software* destinados apenas a análise superficial do esquema proposto, como o *Automation Studio* (AS);
- Melhorar a interface com o usuário, disponibilizando assim a entrada de novos parâmetros de projeto, além dos já existentes, tornando a coleta de necessidades do usuário próxima das condições a serem atendidas pelo projeto pneumático;

- Oferecimento de detalhes mais profundos sobre o projeto, obtidos nas etapas preliminares e detalhadas, com a descrição e especificação dos componentes a serem utilizados bem como procedimentos a serem adotados na montagem dos mesmos;

No geral, o protótipo construído abriu novos caminhos, sendo o horizonte da pneumática amplamente utilizado em contextos diversos, acreditando que com o prosseguimento desta dissertação sejam possíveis futuros trabalhos, como o realizado por Silva (1998), que dentre as considerações de expansão propostas, cita o campo da pneumática, o qual deu origem ao presente trabalho.

Referências bibliográficas

ABHP **Coletânea de artigos técnicos**. [S.I.]: ABHP, 1995.

ALVES, Guilherme Dionízio **Sistema especialista protótipo para diagnóstico de falhas em um sistema hidráulico naval**. 2001. 161 f. Dissertação (Mestrado em Engenharia Mecânica) – Departamento de Engenharia Mecânica, Programa de Pós-Graduação em Engenharia Mecânica, Universidade Federal de Santa Catarina, Florianópolis.

ASI. **Actuators sensors interface**. Disponível em: < <http://www.as-interface.net/docs/en/vor.xml> >. Acesso em: 7 out. 2001.

ATTIÉ, Sérgio Sarquis **Automação hidráulica e pneumática empregando a teoria de sistemas a eventos discretos**. 1998. 191 f. Dissertação (Mestrado em Engenharia Mecânica) – Departamento de Engenharia Mecânica, Programa de Pós-Graduação em Engenharia Mecânica, Universidade Federal de Santa Catarina, Florianópolis.

BACK, Nelson; FORCELLINI, Fernando A. **Projeto de produto**: apostila do curso de pós-graduação em engenharia mecânica da UFSC. [199-?].

BORGES, Joel Brasil; SILVA, Jonny Carlos da **Projeto pneumático – uma nova forma de criação**. In CONGRESSO BRASILEIRO DE ENGENHARIA MECÂNICA, 16., 2001, Uberlândia. Anais... Uberlândia: ABCM, 2001. p. 211 – 218.

BOLTON, W. **Pneumatic and hydraulic systems**. Oxford: Butterworth-Heinemann, 1997. 248 p.

BULLINGER, Hans-Jörg; WARSCHAT, Joachim **Concurrent simultaneous engineering systems: the way to successful product development**. London: Springer-Verlag, 1996. 394 p.

BUCHANAN, Bruce G.; SHORTLIFFE, Edward H. **Rule-based expert systems: the MYCIN experiments of the Stanford heuristic programming project**. Reading: Addison-Wesley, 1984. 748 p.

CHAPMAN, William L.; BAHILL, A. Terry; WYMORE, A. Wayne **Engineering modeling and design**. Boca Raton: CRC, 1992. 371 p.

CHORAFAS, Dimitris N. **Knowledge engineering: knowledge acquisition, knowledge representation, the role of the knowledge engineer, and domains fertile to IA implementation**. New York: Van Nostrand Reinhold, 1990. 380 p.

CLIPS application abstract, 1st Nov. 1997. Disponível em: <<http://www.ghg.net/clips/download/documentation/abstract.pdf>>. Acesso em: 18 dez. 1998.

COAD, Peter; YOURDON, Edward **Análise baseada em objetos**. 2. ed. rev. Rio de Janeiro: Campus, 1997. 225 p.

_____. **Projeto baseado em objetos**. Rio de Janeiro: Campus, 1993. 195 p.

DE NEGRI, Victor Juliano **Estruturação da modelagem se sistemas automáticos e sua aplicação a um banco de testes para sistemas hidráulicos**. 1996. 157 f. Tese (Doutorado em Engenharia Mecânica) – Departamento de Engenharia Mecânica, Programa de Pós-Graduação em Engenharia Mecânica, Universidade Federal de Santa Catarina, Florianópolis.

DURKIN, John **Expert systems: design and development**. Englewood Cliffs: Prentice-Hall, 1994. 800p.

DYM, Clive L.; LEVITT, Raymond E. **Knowledge based systems in engineering**. New York: McGraw-Hill, 1991. 404 p.

FESTO DIDATIC **Técnicas de automação industrial III: SPA3**. [S.I.]: FESTO DIDATIC, 1992. 97 p.

FESTO DIDATIC-BRASIL **Análise e montagem de sistemas pneumáticos**. São Paulo: FESTO DIDATIC-BRASIL, 1995. 142 p.

FRENCH, Michael J. **Conceptual design for engineers**. London: Design council, 1985. 226 p.

GIARRATANO, Joseph **CLIPS: User's Guide**. 5th, Aug. 1998. Disponível em: <<http://www.ghgcorp.com/clips/download/documentation/usrguide.pdf>>. Acesso em: 18 dez. 1998.

GONZALEZ, Avelino J.; DANKEL, Douglas D. **The engineering of knowledge-based systems: theory and practice**. Englewood Cliffs: Prentice-Hall, 1993. 521p.

HAMMER, Michael; CHAMPY, James **Reengenharia: revolucionando a empresa em função dos clientes, da concorrência e das grandes mudanças da gerência**. Rio de Janeiro: Campus, 1994. 189 p.

HARTLEY, John R. **Engenharia simultânea: um método para reduzir prazos, melhorar a qualidade e reduzir custos**. Porto Alegre: Bookman, 1998. 266 p.

HASEBRINK, J. P.; KLOBER, R. **Técnica de comando 1: fundamentos de pneumática/eletropneumática**. [S.I.]: Festo, 1975. 212 p.

_____. **Projeto de sistemas pneumáticos: P122**. 3. ed. [S.I.]: FESTO DIDATIC-BRASIL, 1988. 143 p.

HENEY, Paul **Fluid power: 2000 and beyond a blueprint for the future**. **Hydraulics & pneumatics**, United States of America : Penton publishing, v.51, n.3, p.68-80, mar. 1998.

HUBKA, Vladimir; EDER, W. Ernst **Design Science: Introduction to the needs, scope and organization of engineering design knowledge**. 2. rev. ed. Springer-Verlag: London 1996. 251 p.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. ISO 1219-1: Fluid power systems and components – Graphic symbols and circuit diagrams – Part 1 Graphic symbols. Switzerland, 1991. 38 p.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. ISO 1219-2: Fluid power systems and components – Graphic symbols and circuit diagrams – Part 2 Circuit diagrams. Switzerland, 1995. 21 p.

LUCENA, C. **Inteligência artificial e engenharia de software**. Rio de Janeiro: Jorge Zahar. 1987. 305 p.

MADEIRA, Mauro Notarnicola **Orientação por objetos aplicada à automação industrial: objeto-imagem**. 1995. 135 f. Dissertação (Mestrado em Ciências da Computação) – Programa de Pós-Graduação em Ciências da Computação, Universidade Federal de Santa Catarina, Florianópolis.

MEIXNER, H.; KLOBER R. **Especialização e projeto de sistemas pneumáticos: P131**. 3. ed. [S.I.]: FESTO DIDATIC-BRASIL, 1986. 139 p.

MILLER, Landon C. G. **Concurrent Engineering Design: integrating the best practices for process improvement**. Dearborn: SME, 1993. 319 p.

PAHL, Gerhard.; BEITZ, Wolfgang **Engineering Design: A systematic approach**. 2nd ed. London: Springer Verlag, 1999. 544 pp.

PARSAEI, Hamid R (Ed.); SULLIVAN, Willian G. (Ed.) **Concurrent engineering: comtemporary issues and modern design tools**. London: Chapman&Hall, 1993. 497 p.

PETROSKI, Henry **Design paradigms: case histories of error and judgment in engineering**. New York: Press Syndicate of Cambridge, 1994. 209 p.

PETROUTSOS, Evangelos **Dominando o visual basic 6: a bíblia**. São Paulo: Makron, 1999. 1126 p.

PRINCÍPIOS básicos; produção; distribuição e condicionamento. [S.I.: s.n.], 19--?. 115 p.

RICH, Elaine **Inteligência artificial**. São Paulo: McGraw-Hill, 1988. 503 p.

RILEY, Gary; GIARRATANO, Joseph C. **Expert systems: principles and programming**. 2nd ed. Boston: PWS, 1994. 644 p.

ROOZENBURG, N. F. M.; EEKELS, J. **Product design: fundamentals and methods**. Chichester: Wiley, 1995. 408 p.

SCHNEIDER, Richard T.; HITCHCOX, Alan L. Fluid power technology shapes industry worldwide. **Hydraulics & pneumatics**, United States of America : Penton publishing, v.51, n.3, p.59-122, mar. 1998.

SILVA, Jonny Carlos da **Expert system prototype for hydraulic system design focusing on concurrent engineering aspects**. 1998. 185 f. Tese (Doutorado em Engenharia Mecânica) – Departamento de Engenharia Mecânica, Programa de Pós-Graduação em Engenharia Mecânica, Universidade Federal de Santa Catarina, Florianópolis.

THIBAUT, R. **Automatismo: pneumáticos e hidráulicos**. Rio de Janeiro: Livros técnicos e científicos, 1979. 61 p.

Bibliografia

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 6023: Informação e documentação – Referências – Elaboração. Rio de Janeiro, 2000. 22 p.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 10520: Informação e documentação – Apresentação de citações em documentos. Rio de Janeiro, 2001. 4 p.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 14724: Informação e documentação – Trabalhos acadêmicos – Apresentação. Rio de Janeiro, 2001. 6 p.

CLIPS: Reference Manual. 5th, Aug. 1998. Disponível em: <<http://www.ghgcorp.com/clips/download/documentation/bpg.pdf>>. Acesso em: 12 jul. 1999.

CORRÊA, Rosa **Publicação eletrônica**: revisão norma NBR10520 [mensagem eletrônica]. Mensagem recebida por <joelbr@zipmail.com.br> de <rosamcorrea@uol.com.br> em 9 jan 2002.

_____. **Publicação eletrônica**: revisão norma NBR10520 [mensagem eletrônica]. Mensagem recebida por <joelbr@zipmail.com.br> de <rosamcorrea@uol.com.br> em 14 jan 2002.

CSE html, versão 4.0521: validator professional. [S.I.]: AI Internet Solutions, 2000. Disponível em: <<http://www.htmlvalidator.com>>. Acesso em: 17 jul. 2000.

DICIONÁRIO Aurélio Eletrônico, versão 3.0: século XXI. [S.I.]: Lexikon, 1999.

FÜRSTENAU, Eugênio **Novo dicionário de termos técnicos inglês-portugues**. São Paulo: Globo, 1999. 1413 p.

HALVORSON, Michael **Microsoft visual basic 6.0 profissional passo a passo**. São Paulo: Makron, 1999. 427 p.

HOUAISS, Antônio; CARDIM, Ismael **Novo webster's: dicionário universitário**. Rio de Janeiro: Record, 1998. 914 p.

LEMAY, Laura **Aprenda em 1 semana HTML 4**. Rio de Janeiro: Campus, 1998. 631 p.

MICROSOFT visual basic 5.0: programmer's guide. Redmond: Microsoft press, 1997. 898 p.

VARELA, José Roberto **Iniciando no visual basic 5.0: desenvolva rapidamente aplicações windows com o visual basic**. 2. ed. São Paulo: Érica, 1997, 542 p.

Apêndice A – Telas do protótipo no ambiente CLIPS padrão

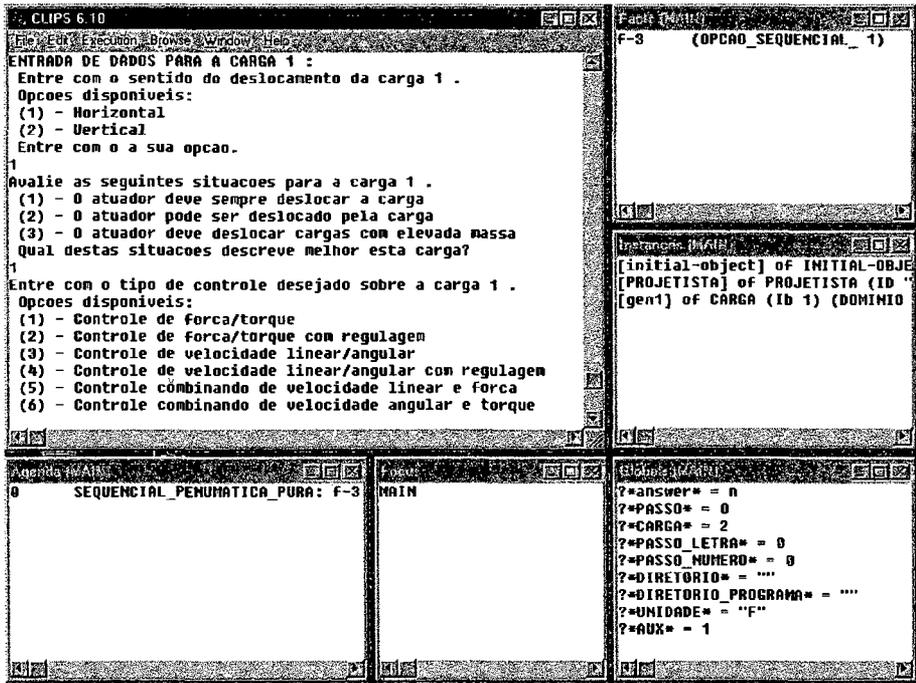


Figura A.1 – Tela do protótipo na interface padrão do CLIPS.

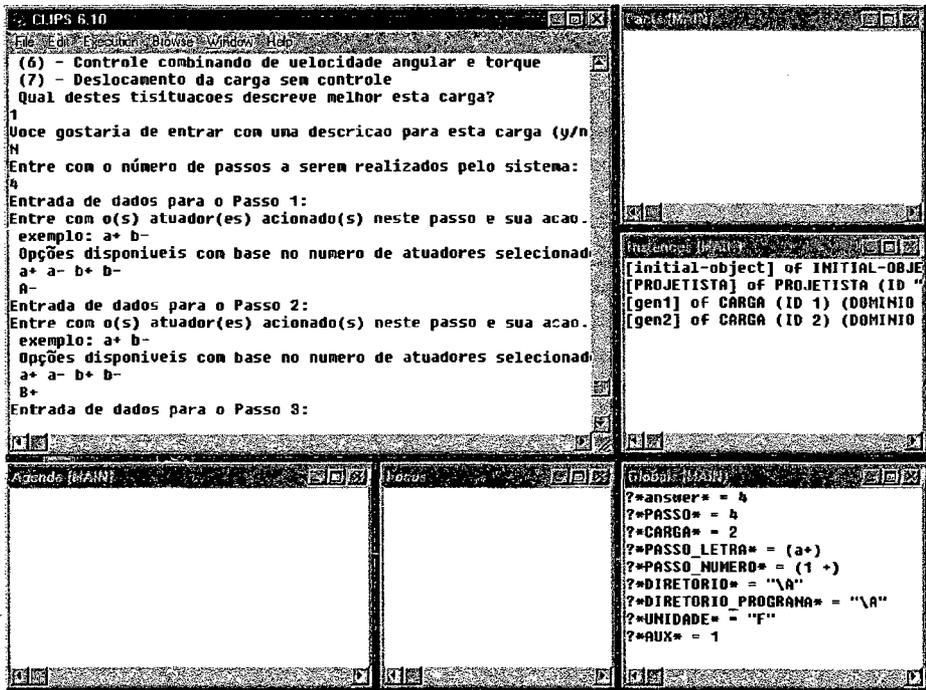


Figura A.2 – Tela do protótipo na interface padrão do CLIPS.

Nestas figuras, demonstra-se as telas que o protótipo possui, quando a *base de conhecimento* (BC) utilizada é operada através da interface CLIPS padrão, onde além da janela principal do programa, pode-se constatar as janelas auxiliares, onde são descritos as características de operação do protótipo, enquanto é utilizado o *software*, como *agenda das regras*, foco referente a que parte da BC está sendo utilizada, variáveis e seus valores no momento, lista de *objetos* existentes em conjunto com os valores dos *atributos* e por fim os *fatos* existentes.

Apêndice B – Telas do protótipo no ambiente orientado a eventos

No início da utilização do protótipo, o usuário tem a possibilidade de obter as informações iniciais básicas necessárias sobre alguns detalhes do *software*, como ilustra a Figura A.3.

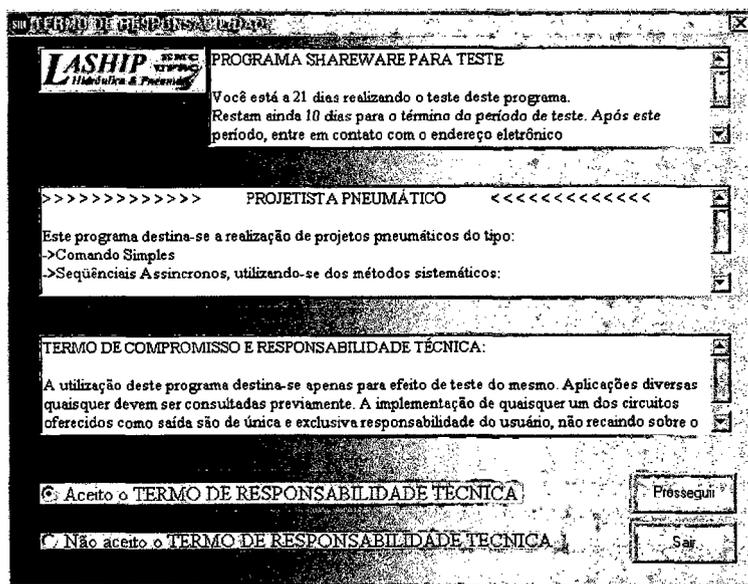


Figura A.3 – Tela de entrada do protótipo.

Estando o usuário ciente das condições de utilização do protótipo, pode-se prosseguir com a utilização do mesmo, que inicia com uma seleção prévia do sistema operacional utilizado como base para executar o protótipo, compondo informações que descrevem a habilitação de funções, conforme limitações de cada sistema operacional, segundo apresentado na Figura A.4.

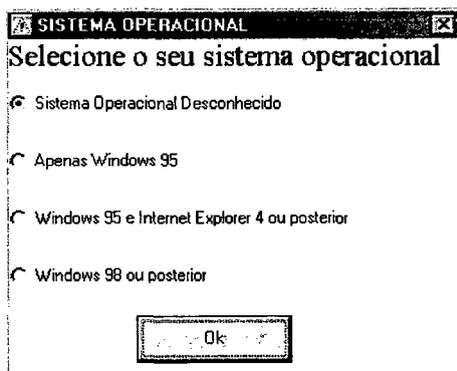


Figura A.4 – Seleção do sistema operacional.

Após a personalização do protótipo aos recursos computacionais existentes, pode-se prosseguir para a tela principal do protótipo, cujo menu principal é ilustrado na Figura A.5.

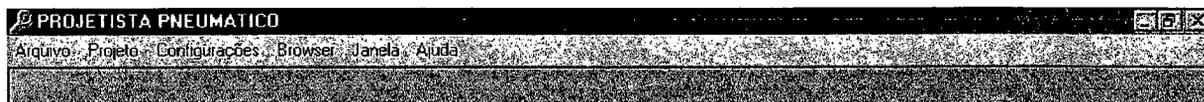


Figura A.5 – Menu de opções do protótipo.

Dentre as opções de projeto existentes, o usuário pode selecioná-las conforme ilustrado na Figura A.6, sendo porém algumas das opções existentes ainda não implementadas, podendo ser disponibilizadas em trabalhos futuros.

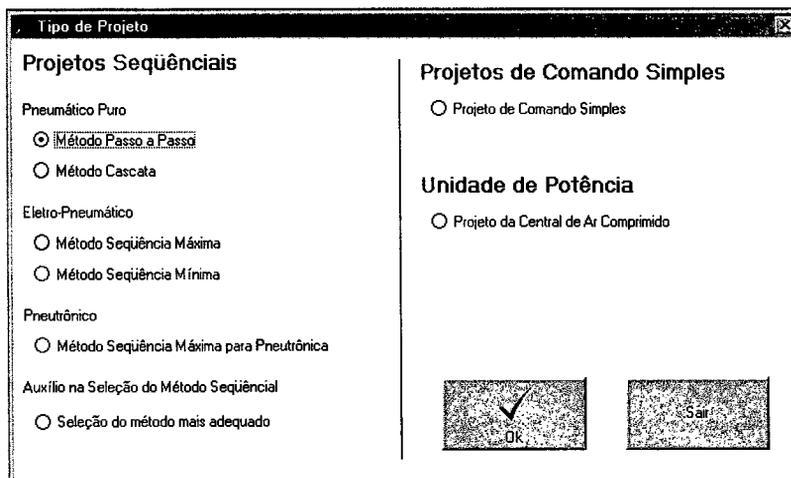


Figura A.6 – Seleção do tipo de projeto que deseja-se realizar.

Conforme o tipo de projeto, são solicitados os parâmetros que devem ser descritos na entrada do projeto, que para os projetos seqüências realizados, tanto pelos métodos passo-a-passo como pelo método seqüência máxima, seguem a mesma formação, pois o que diferencia estes métodos é a forma de construir o projeto, sendo possível, para ambos os casos, utilizar o mesmo formato de entrada de dados. A diferença significativamente existe quando da seleção dos projetos de *simples comando*, no qual não são requisitados dados para a modelagem comportamental das cargas mecânicas. Segue na Figura A.7 a tela exemplo para o projeto a ser realizado pelo método seqüência máxima.

Apesar de haverem muitos comandos disponíveis na tela da Figura A.7, todos estes estão intercorrelacionados, de forma a identificar e evitar a entrada de informações errôneas

ou inconsistentes sobre o projeto descrito pelo usuário, como caracteres inválidos, modelagem comportamental das cargas inconsistentes, entre outras falhas. Com grande interatividade, o usuário pode também constatar as informações indicadas sobre o projeto, através do diagrama *trajeto-passo*.

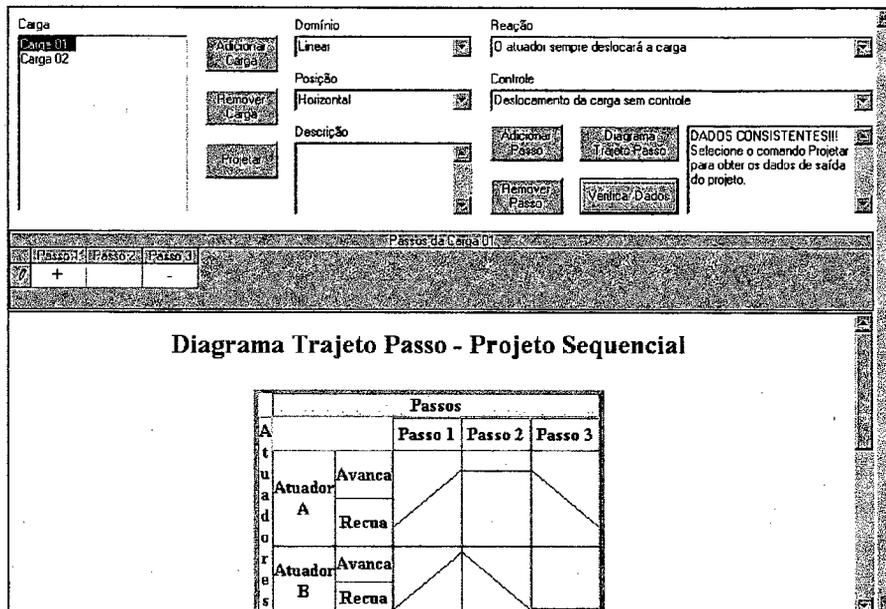


Figura A.7 – Seleção do sistema operacional.

Após a constatação da coerência nos dados informados pelo usuário, é possível prosseguir com o projeto, sendo estes dados inseridos na *base de conhecimento* (BC) para originar os esquemas do projeto. Os dados podem ser visualizados em um *browser* para HTML, com suporte para a versão 4.0. Outra opção é analisar os resultados do projeto no próprio protótipo, o qual possui internamente suporte para apresentação dos esquemas do projeto pneumático, como ilustrado na Figura A.8, onde as funções tradicionais de um *browser* são disponibilizadas.

Como a disposição dos componentes segue uma estrutura de hipertexto, a Figura A.8 descreve uma das telas do protótipo, a qual é dinâmica e pode ser conduzida conforme as necessidades de informações que o usuário desejar.

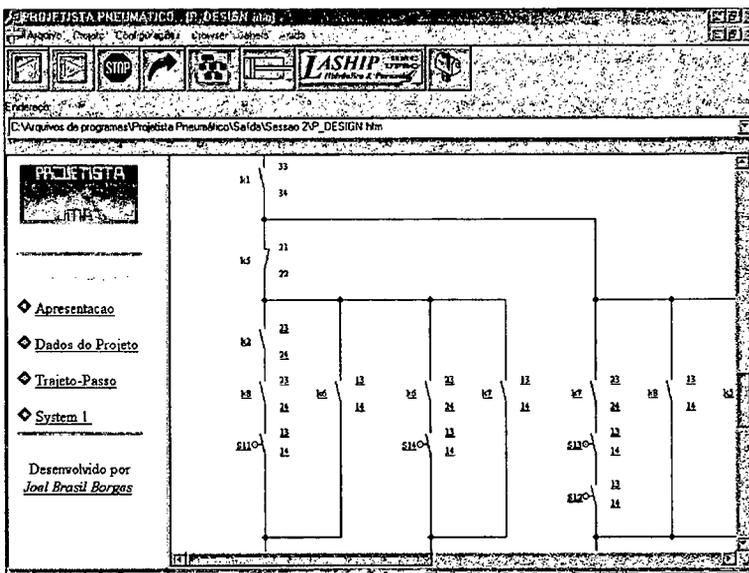


Figura A.8 – Browser do protótipo.

Informações complementares sobre o projeto podem ser utilizadas, como na descrição de dados do projetista e detalhes qualitativos desejados sobre o protótipo, conforme utilizado por Silva (1998), e apresentado na Figura A.9 e Figura A.10, apesar de algumas destas opções não estarem ainda totalmente implementadas, como os dados qualitativos sobre os projetos, que resta ainda realizar a coleta do conhecimento especializado para ser inserido na BC, o que pelo ponto de vista computacional pode ser feito sem maiores complicações, pois todos os procedimentos para que isto seja possível já estão implementados. Para a obtenção de uma versão de teste do protótipo, consulte o endereço eletrônico <http://www.pndr.hpg.com.br>.

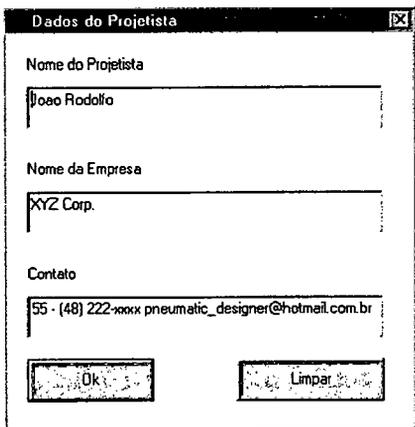


Figura A.9 – Tela de entrada de dados do projetista.

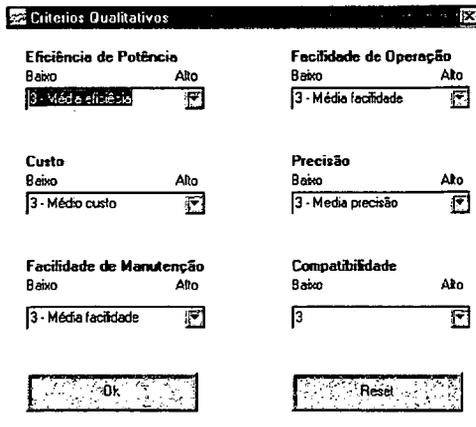


Figura A.10 – Tela de entrada de dados qualitativos sobre o projeto.

Apêndice C – Considerações computacionais sobre o protótipo

Apesar de não ser o objetivo principal deste trabalho realizar um trabalho de pesquisa meramente computacional, pois a essência do mesmo é abordar a *engenharia do conhecimento* no domínio de projeto de sistemas pneumáticos, em conjunto com a aplicação da metodologia de projeto no desenvolvimento de protótipos computacionais em SE, vale destacar algumas considerações computacionais adotadas ao projeto do protótipo, que proporcionaram atingir o resultado final, através de uma breve descrição dos procedimentos adotados, que podem auxiliar na documentação da totalidade do protótipo, inclusive da parcela computacional.

Inicialmente descreve-se a adoção da linguagem *Visual Basic* (VB) 5.0, pois a sua utilização é simples e encontra-se disponível uma versão oficial, requisito para poder gerar e distribuir versões do executável do protótipo para usuários externos ao trabalho, sem apresentar problemas com relação aos direitos autorais sobre o VB. Como motivos adicionais na utilização do VB 5.0 tem-se: velocidade compatível com o grau de exigência requerido no interfaceamento, robustez na transferência de dados, possibilidade de criação de discos de instalação do protótipo e simplicidade na utilização, facilitando o trabalho do EC, pois é ele que utiliza esta ferramenta na construção das interfaces do protótipo com o usuário.

Dentre os formatos HTML existentes para a representação dos esquemas de saída, a versão 4.0 é adotada como padrão por ser a mais recente e aceita pela grande maioria dos *browser* para internet, além de operar em conformidade com o *ambiente integrado de saída* e ser o padrão que suporta a opção de *frames*. Para garantir a adequação das saídas HTML, para que estas não apresentem erros durante a apresentação, o auxílio de validadores de HTML é de grande auxílio durante a evolução do protótipo, simplificando o trabalho do programador e reduzindo o tempo de depuração. Como opção, foi utilizado o *software* de validação de HTML titulado *CSE HTML validator professional* versão 4.05, o qual é uma versão *shareware*, sem custos ao projeto.

Como o *software* CLIPS possui deficiências na formação do código com que o protótipo manipula, para a implementação do protótipo, durante a formação da BC, foram analisados vários editores de auxílio à programação, sendo optado pelo *software* TextPad 4.0, que possui ferramentas de auxílio na identificação da inserção de comandos em rotinas, possibilidade de trabalhar com vários arquivos de código simultaneamente, além de ser *shareware*.

Apesar da efemeridade destas ferramentas, pois estas estão em contínua evolução, é importante destacar que a seleção de ferramentas adequadas para determinadas tarefas é de extrema importância devido à simplificação do trabalho e otimização dos recursos.

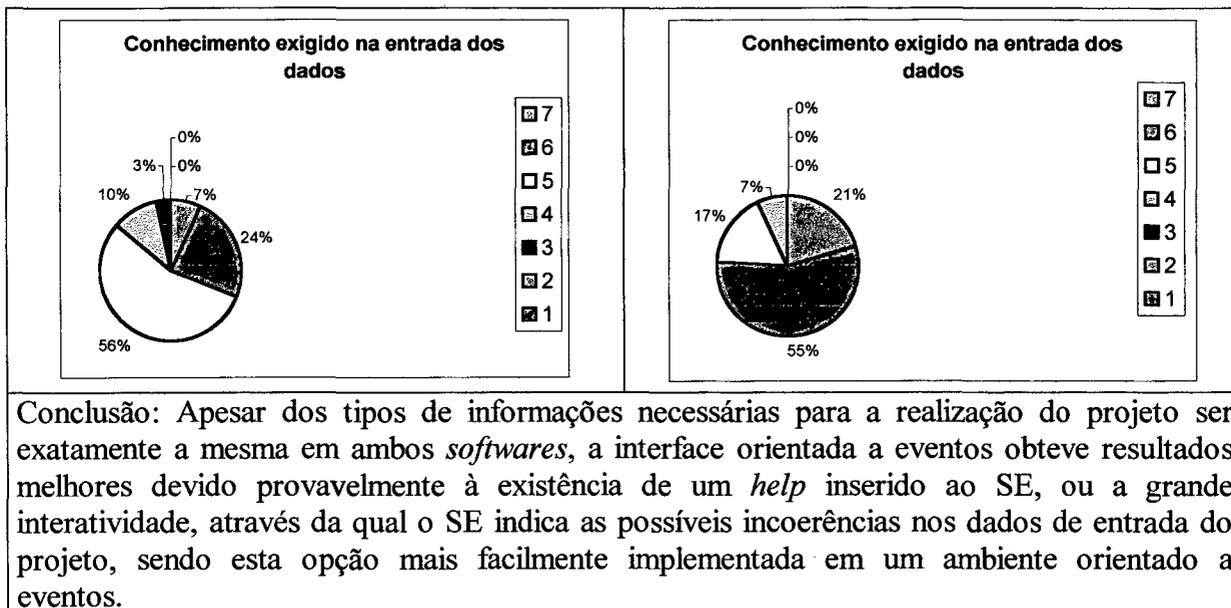
Apêndice D – Comparação entre as versões do protótipo

A seguir, descreve-se os resultados da pesquisa com os validadores não especialistas em conjunto com a comparação e as conclusões sobre estas duas versões do protótipo, descrito nas demais tabelas existentes nesta seção.

Tabela A.1 – Validação da interface de entrada por não especialistas.

Interface de entrada																																	
Com interface CLIPS padrão	Com interface gráfica orientada a eventos																																
<p>Facilidade de entrada dos dados</p> <table border="1"> <caption>Facilidade de entrada dos dados - Interface CLIPS padrão</caption> <thead> <tr><th>Nota</th><th>Porcentagem</th></tr> </thead> <tbody> <tr><td>1</td><td>42%</td></tr> <tr><td>2</td><td>38%</td></tr> <tr><td>3</td><td>17%</td></tr> <tr><td>4</td><td>3%</td></tr> <tr><td>5</td><td>0%</td></tr> <tr><td>6</td><td>0%</td></tr> <tr><td>7</td><td>0%</td></tr> </tbody> </table>	Nota	Porcentagem	1	42%	2	38%	3	17%	4	3%	5	0%	6	0%	7	0%	<p>Facilidade de entrada dos dados</p> <table border="1"> <caption>Facilidade de entrada dos dados - Interface gráfica orientada a eventos</caption> <thead> <tr><th>Nota</th><th>Porcentagem</th></tr> </thead> <tbody> <tr><td>1</td><td>45%</td></tr> <tr><td>2</td><td>31%</td></tr> <tr><td>3</td><td>17%</td></tr> <tr><td>4</td><td>7%</td></tr> <tr><td>5</td><td>0%</td></tr> <tr><td>6</td><td>0%</td></tr> <tr><td>7</td><td>0%</td></tr> </tbody> </table>	Nota	Porcentagem	1	45%	2	31%	3	17%	4	7%	5	0%	6	0%	7	0%
Nota	Porcentagem																																
1	42%																																
2	38%																																
3	17%																																
4	3%																																
5	0%																																
6	0%																																
7	0%																																
Nota	Porcentagem																																
1	45%																																
2	31%																																
3	17%																																
4	7%																																
5	0%																																
6	0%																																
7	0%																																
<p>Conclusão: A adoção da interface orientada a eventos simplifica a entrada de dados, que pode ser justificada pela: flexibilidade na entrada dos dados na seqüência com que o usuário deseja, facilidade de correção de algum destes dados e o prévio conhecimento dos usuários em relação aos menus existentes nos aplicativos convencionais para <i>Windows</i>, ao invés dos <i>softwares</i> que utilizam <i>linha de comando</i>.</p>																																	
<p>Interatividade de entrada dos dados</p> <table border="1"> <caption>Interatividade de entrada dos dados - Interface CLIPS padrão</caption> <thead> <tr><th>Nota</th><th>Porcentagem</th></tr> </thead> <tbody> <tr><td>1</td><td>35%</td></tr> <tr><td>2</td><td>31%</td></tr> <tr><td>3</td><td>28%</td></tr> <tr><td>4</td><td>3%</td></tr> <tr><td>5</td><td>3%</td></tr> <tr><td>6</td><td>0%</td></tr> <tr><td>7</td><td>0%</td></tr> </tbody> </table>	Nota	Porcentagem	1	35%	2	31%	3	28%	4	3%	5	3%	6	0%	7	0%	<p>Interatividade de entrada dos dados</p> <table border="1"> <caption>Interatividade de entrada dos dados - Interface gráfica orientada a eventos</caption> <thead> <tr><th>Nota</th><th>Porcentagem</th></tr> </thead> <tbody> <tr><td>1</td><td>49%</td></tr> <tr><td>2</td><td>34%</td></tr> <tr><td>3</td><td>10%</td></tr> <tr><td>4</td><td>7%</td></tr> <tr><td>5</td><td>0%</td></tr> <tr><td>6</td><td>0%</td></tr> <tr><td>7</td><td>0%</td></tr> </tbody> </table>	Nota	Porcentagem	1	49%	2	34%	3	10%	4	7%	5	0%	6	0%	7	0%
Nota	Porcentagem																																
1	35%																																
2	31%																																
3	28%																																
4	3%																																
5	3%																																
6	0%																																
7	0%																																
Nota	Porcentagem																																
1	49%																																
2	34%																																
3	10%																																
4	7%																																
5	0%																																
6	0%																																
7	0%																																
<p>Conclusão: Apesar da interface CLIPS padrão oferecer bom desempenho, a interface orientada a eventos tem resultado melhor, que pode ser justificado pela possibilidade de correções e adequações ao projeto.</p>																																	

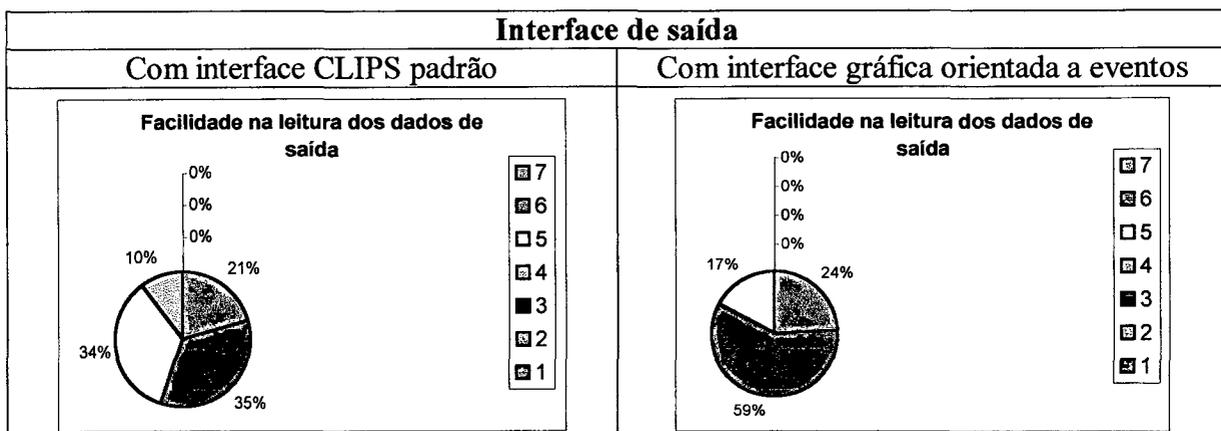
Tabela A.1 – Validação da interface de entrada por não especialistas (continuação).



Apesar da dificuldade de analisar integralmente os resultados da avaliação do protótipo, pode-se determinar, a partir dos dados da Tabela A.1, que ambos os SE possuem boa adequação para utilização, segundo opinião dos usuários, sendo o SE que utiliza interface orientada a eventos o de maior aceitação.

Na continuidade da avaliação dos SE, na Tabela A.2 são apresentados os resultados da avaliação do SE com relação a interface de saída.

Tabela A.2 – Validação da interface de saída por não especialistas.



Conclusão: Algumas flutuações da representatividade das notas 5 e 6 podem ser constatadas, com influência positiva para o SE com interface orientada a eventos. Isto pode ser considerado como correlacionado ao ambiente integrado do SE, que em um *software* o usuário tem acesso tanto a entrada de dados, como a obtenção dos dados de saída do projeto, o que não ocorre com o SE que utiliza a interface CLIPS padrão, que necessita do auxílio de outro *software* com possibilidade de acesso ao formato HTML, exigindo do usuário a utilização de dois programas simultaneamente.

Tabela A.2 – Validação da interface de saída por não especialistas (continuação).

Interface de saída																																	
Com interface CLIPS padrão	Com interface gráfica orientada a eventos																																
<p>Interatividade dos dados de saída</p> <table border="1"> <caption>Interatividade dos dados de saída - Interface CLIPS padrão</caption> <tr><th>Nota</th><th>Porcentagem</th></tr> <tr><td>1</td><td>52%</td></tr> <tr><td>2</td><td>38%</td></tr> <tr><td>3</td><td>7%</td></tr> <tr><td>4</td><td>3%</td></tr> <tr><td>5</td><td>0%</td></tr> <tr><td>6</td><td>0%</td></tr> <tr><td>7</td><td>0%</td></tr> </table>	Nota	Porcentagem	1	52%	2	38%	3	7%	4	3%	5	0%	6	0%	7	0%	<p>Interatividade dos dados de saída</p> <table border="1"> <caption>Interatividade dos dados de saída - Interface gráfica orientada a eventos</caption> <tr><th>Nota</th><th>Porcentagem</th></tr> <tr><td>1</td><td>45%</td></tr> <tr><td>2</td><td>31%</td></tr> <tr><td>3</td><td>24%</td></tr> <tr><td>4</td><td>0%</td></tr> <tr><td>5</td><td>0%</td></tr> <tr><td>6</td><td>0%</td></tr> <tr><td>7</td><td>0%</td></tr> </table>	Nota	Porcentagem	1	45%	2	31%	3	24%	4	0%	5	0%	6	0%	7	0%
Nota	Porcentagem																																
1	52%																																
2	38%																																
3	7%																																
4	3%																																
5	0%																																
6	0%																																
7	0%																																
Nota	Porcentagem																																
1	45%																																
2	31%																																
3	24%																																
4	0%																																
5	0%																																
6	0%																																
7	0%																																
<p>Conclusão: Neste critério de avaliação, a interface orientada a eventos demonstra a maior interatividade com o usuário, que novamente pode ter origem no ambiente integrado do protótipo, pois após analisar os resultados de saída, rapidamente em um único <i>software</i> o usuário pode realizar as modificações nos dados de entrada que ele considere necessária e conveniente.</p>																																	
<p>Apresentação dos sistemas, etc.</p> <table border="1"> <caption>Apresentação dos sistemas, etc. - Interface CLIPS padrão</caption> <tr><th>Nota</th><th>Porcentagem</th></tr> <tr><td>1</td><td>31%</td></tr> <tr><td>2</td><td>24%</td></tr> <tr><td>3</td><td>17%</td></tr> <tr><td>4</td><td>28%</td></tr> <tr><td>5</td><td>0%</td></tr> <tr><td>6</td><td>0%</td></tr> <tr><td>7</td><td>0%</td></tr> </table>	Nota	Porcentagem	1	31%	2	24%	3	17%	4	28%	5	0%	6	0%	7	0%	<p>Apresentação dos sistemas, etc.</p> <table border="1"> <caption>Apresentação dos sistemas, etc. - Interface gráfica orientada a eventos</caption> <tr><th>Nota</th><th>Porcentagem</th></tr> <tr><td>1</td><td>41%</td></tr> <tr><td>2</td><td>38%</td></tr> <tr><td>3</td><td>14%</td></tr> <tr><td>4</td><td>7%</td></tr> <tr><td>5</td><td>0%</td></tr> <tr><td>6</td><td>0%</td></tr> <tr><td>7</td><td>0%</td></tr> </table>	Nota	Porcentagem	1	41%	2	38%	3	14%	4	7%	5	0%	6	0%	7	0%
Nota	Porcentagem																																
1	31%																																
2	24%																																
3	17%																																
4	28%																																
5	0%																																
6	0%																																
7	0%																																
Nota	Porcentagem																																
1	41%																																
2	38%																																
3	14%																																
4	7%																																
5	0%																																
6	0%																																
7	0%																																
<p>Conclusão: Como as saídas em ambos os SE é a mesma, o que pode justificar a pequena diferença positiva para o SE que utiliza interface orientada a eventos é a integração de todas as funções que o SE possui em um único <i>software</i>, em que após a entrada dos dados do projeto, o usuário já pode constatar os resultados do projeto.</p>																																	
<p>Velocidade na obtenção das saídas</p> <table border="1"> <caption>Velocidade na obtenção das saídas - Interface CLIPS padrão</caption> <tr><th>Nota</th><th>Porcentagem</th></tr> <tr><td>1</td><td>31%</td></tr> <tr><td>2</td><td>39%</td></tr> <tr><td>3</td><td>24%</td></tr> <tr><td>4</td><td>3%</td></tr> <tr><td>5</td><td>0%</td></tr> <tr><td>6</td><td>3%</td></tr> <tr><td>7</td><td>0%</td></tr> </table>	Nota	Porcentagem	1	31%	2	39%	3	24%	4	3%	5	0%	6	3%	7	0%	<p>Velocidade na obtenção das saídas</p> <table border="1"> <caption>Velocidade na obtenção das saídas - Interface gráfica orientada a eventos</caption> <tr><th>Nota</th><th>Porcentagem</th></tr> <tr><td>1</td><td>42%</td></tr> <tr><td>2</td><td>41%</td></tr> <tr><td>3</td><td>14%</td></tr> <tr><td>4</td><td>3%</td></tr> <tr><td>5</td><td>0%</td></tr> <tr><td>6</td><td>0%</td></tr> <tr><td>7</td><td>0%</td></tr> </table>	Nota	Porcentagem	1	42%	2	41%	3	14%	4	3%	5	0%	6	0%	7	0%
Nota	Porcentagem																																
1	31%																																
2	39%																																
3	24%																																
4	3%																																
5	0%																																
6	3%																																
7	0%																																
Nota	Porcentagem																																
1	42%																																
2	41%																																
3	14%																																
4	3%																																
5	0%																																
6	0%																																
7	0%																																
<p>Conclusão: Em ambos os SE, a velocidade de execução do projeto é muito semelhante, pois a passagem pela BC para a obtenção dos dados de saída do projeto é a mesma, sendo por isso os resultados muito semelhantes, nos quais a pequena diferença preferencial para o SE com interface orientada a eventos pode ser atribuída à agilização que o ambiente integrado de projeto propicia.</p>																																	

Na totalidade, os SE possuem uma interface de saída dos projetos adequada. Apesar das saídas em ambos os SE ser a mesma, o SE que utiliza a interface orientada a eventos pode ser considerado como mais conveniente, como constata os dados da Tabela A.2. Isto pode ser correlacionado ao maior ajuste desta interface com o usuário.

Como o objetivo principal da validação do protótipo com usuários não especialistas é obter resultados sobre o desempenho do SE com relação à interface existente, os resultados da validação podem findar-se neste estágio. No intuito de complementar os resultados da validação, é descrito na Tabela A.3 o resultado sobre a avaliação da BC.

Tabela A.3 – Validação da BC por não especialistas.

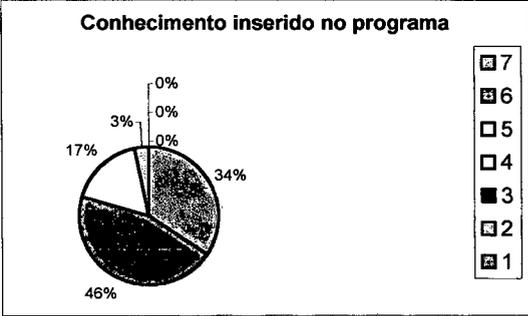
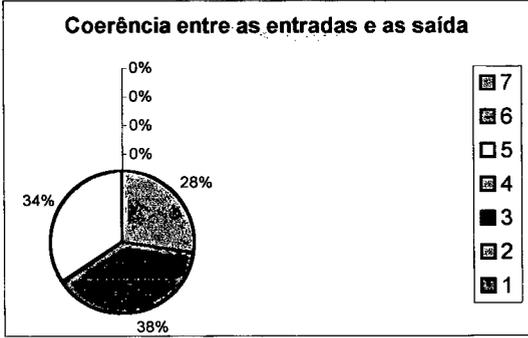
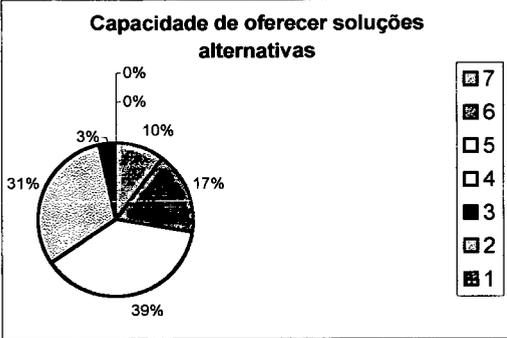
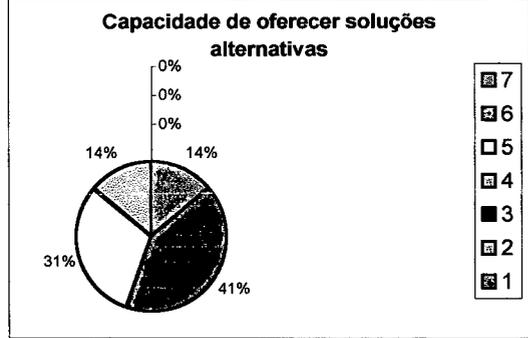
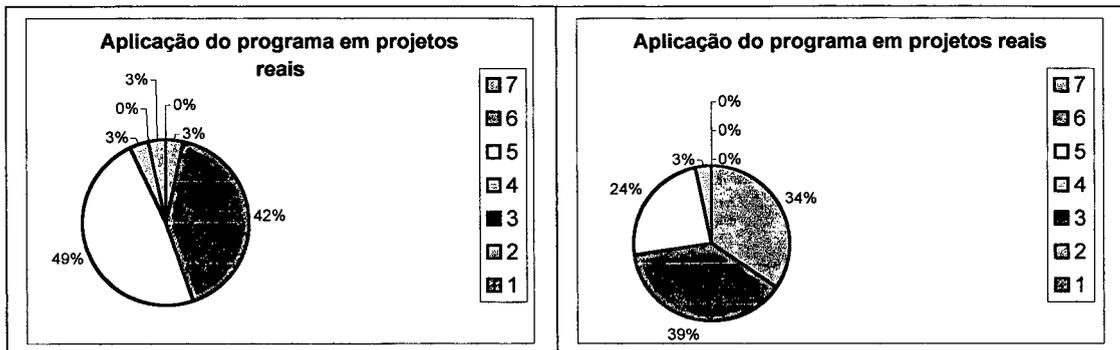
Base de conhecimento															
Com interface CLIPS padrão	Com interface gráfica orientada a eventos														
<p>Conhecimento inserido no programa</p>  <table border="1"> <tr><td>7</td></tr> <tr><td>6</td></tr> <tr><td>5</td></tr> <tr><td>4</td></tr> <tr><td>3</td></tr> <tr><td>2</td></tr> <tr><td>1</td></tr> </table>	7	6	5	4	3	2	1	<p>Conhecimento inserido no programa</p>  <table border="1"> <tr><td>7</td></tr> <tr><td>6</td></tr> <tr><td>5</td></tr> <tr><td>4</td></tr> <tr><td>3</td></tr> <tr><td>2</td></tr> <tr><td>1</td></tr> </table>	7	6	5	4	3	2	1
7															
6															
5															
4															
3															
2															
1															
7															
6															
5															
4															
3															
2															
1															
<p>Coerência entre as entradas e as saída</p>  <table border="1"> <tr><td>7</td></tr> <tr><td>6</td></tr> <tr><td>5</td></tr> <tr><td>4</td></tr> <tr><td>3</td></tr> <tr><td>2</td></tr> <tr><td>1</td></tr> </table>	7	6	5	4	3	2	1	<p>Coerência entre as entradas e as saída</p>  <table border="1"> <tr><td>7</td></tr> <tr><td>6</td></tr> <tr><td>5</td></tr> <tr><td>4</td></tr> <tr><td>3</td></tr> <tr><td>2</td></tr> <tr><td>1</td></tr> </table>	7	6	5	4	3	2	1
7															
6															
5															
4															
3															
2															
1															
7															
6															
5															
4															
3															
2															
1															
<p>Capacidade de oferecer soluções alternativas</p>  <table border="1"> <tr><td>7</td></tr> <tr><td>6</td></tr> <tr><td>5</td></tr> <tr><td>4</td></tr> <tr><td>3</td></tr> <tr><td>2</td></tr> <tr><td>1</td></tr> </table>	7	6	5	4	3	2	1	<p>Capacidade de oferecer soluções alternativas</p>  <table border="1"> <tr><td>7</td></tr> <tr><td>6</td></tr> <tr><td>5</td></tr> <tr><td>4</td></tr> <tr><td>3</td></tr> <tr><td>2</td></tr> <tr><td>1</td></tr> </table>	7	6	5	4	3	2	1
7															
6															
5															
4															
3															
2															
1															
7															
6															
5															
4															
3															
2															
1															

Tabela A.3 – Validação da BC por não especialistas (continuação).

Como a BC é a mesma para os dois SE testados, é de se esperar que os resultados sejam muito semelhantes para todos os critérios de avaliação. No entanto, ocorrem algumas flutuações com relação aos resultados destes critérios avaliados, apresentando sempre um melhor resultado o SE que utiliza interface orientada a eventos. Este fato pode estar correlacionado aos resultados positivos conseguidos pelo SE na totalidade, que indiretamente atuaram sobre o resultado da avaliação da BC.

Analisando o resultado global desta pesquisas referente às interfaces do protótipo, constata-se a melhor aceitação do protótipo quando o mesmo é analisado utilizando a interface orientada a eventos.