

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

**PROJETO DE SISTEMAS AUTOMÁTICOS COM MODELAGEM E CONTROLE DA
COMUNICAÇÃO COM O AMBIENTE EXTERNO**

Dissertação submetida à

UNIVERSIDADE FEDERAL DE SANTA CATARINA

para a obtenção do grau de

MESTRE EM ENGENHARIA MECÂNICA

RODRIGO BARBOSA SOUTO

Florianópolis, Março de 2005

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

**PROJETO DE SISTEMAS AUTOMÁTICOS COM MODELAGEM E CONTROLE DA CO-
MUNICAÇÃO COM O AMBIENTE EXTERNO**

RODRIGO BARBOSA SOUTO

Esta dissertação foi julgada adequada para a obtenção do título de

MESTRE EM ENGENHARIA

ESPECIALIDADE ENGENHARIA MECÂNICA

sendo aprovada em sua forma final.

Victor Juliano De Negri, Dr. Eng. - Orientador

José Eduardo Ribeiro Cury, Dr. d'Etat – Co-orientador

Antônio Bellini da Cunha Neto, Dr - Coordenador do Curso

BANCA EXAMINADORA

Acires Dias, Dr. Eng.

Max Hering de Queiroz, Dr. Eng.

Fernando Antônio Forcellini, Dr. Eng.

Eduardo Alves Portela Santos, Dr. Eng.

“[...] science is not done in a vacuum. It is done in a social context, and the results of science have important implications for society, even if it is simply providing a general understanding of how we humans fit into the cosmos.

Thus, simply producing new knowledge, without making any attempt to help disseminate it and explain it, is not enough. I think one cannot expect every scientist to spend time on the effort to explain science. But in a society in which the science is of vital importance and also in which many forces are trying to distort the results of science, it is crucial that some of us speak out.”

(SCIENTIFIC AMERICAN, 2004)

**Lawrence M. Krauss, Ph.D.
Case Western Reserve University, Cleveland, Ohio.**

“[...] a ciência não é feita no vácuo. Ela é feita em um contexto social e seus resultados têm aplicações importantes para a sociedade, mesmo quando ela simplesmente promove um entendimento geral de como nós seres humanos nos encaixamos no cosmos.

Assim, produzir novos conhecimentos simplesmente, sem fazer nenhum esforço para ajudar a disseminá-lo e explicá-lo, não é suficiente. Acho que não se pode esperar que cada cientista gaste tempo e esforço para explicar a ciência. Entretanto, em uma sociedade na qual a ciência é de vital importância e também na qual muitas forças estão tentando distorcer seus resultados, é crucial que alguém de nós dê sua opinião.”

(SCIENTIFIC AMERICAN, 2004 – tradução nossa)

**Lawrence M. Krauss, Ph.D.
Case Western Reserve University, Cleveland, Ohio.**

**Para minha filha Laura.
Para minha esposa Genice,
companheira sempre.**

AGRADECIMENTOS

Primeiramente agradeço aos meus orientadores, Prof. Victor J. De Negri e Prof. José Eduardo R. Cury, por acreditarem e apoiarem este trabalho de maneira paciente diante de um orientado muitas vezes teimoso. Através de seus conselhos, observações e condução experiente senti-me sempre seguro e norteado no caminho correto para meu melhor aprendizado técnico e pessoal.

Em auxílio a este estudo, também estavam os colegas da pós-graduação e do LASHIP que, através do companheirismo e convívio sadio e harmonioso, contribuíram para o engrandecimento técnico e equilíbrio emocional deste autor, fazendo com que o cansaço desaparecesse diante de demonstrações de bom humor e solidariedade. Obrigado a todos.

Agradeço a UFSC e ao PosMec por mais esta oportunidade de crescimento dentro de um ambiente rico em conhecimento e cultura.

Agradeço a CAPES pelo fomento deste trabalho, sem o qual ele não poderia ser realizado.

A minha esposa, agradeço pela paciência sem fim e o apoio nos momentos de cansaço, mesmo quando seu esforço para nos presentear com nossa filha foi muito maior que o meu para concluir este trabalho.

Aos meus demais familiares, agradeço o apoio e a compreensão pelo meu breve afastamento de todos para realizar esta tarefa.

Mais uma das infinitas vezes agradeço a Deus por mais este feito – peço-lhe que este seu filho possa ser merecedor de muitas outras conquistas ainda por vir.

SUMÁRIO

Lista de figuras.....	xiii
Lista de tabelas.....	xix
Simbologias e abreviaturas.....	xxi
Resumo.....	xxiii
Abstract.....	xxv
1. Introdução.....	1
1.1 Cenário atual.....	1
1.2 Objetivos.....	3
1.3 Estrutura do documento.....	5
2. Modelo consensual.....	7
2.1 Projeto Informacional.....	8
2.1.1 Planejar o projeto informacional.....	9
2.1.2 Pesquisar informações sobre o problema de projeto.....	9
2.1.2.1 Análise do problema de projeto.....	9
2.1.2.2 Pesquisa de informações necessárias ao trabalho de projeto.....	10
2.1.2.3 Pesquisa de produtos concorrentes ou similares.....	10
2.1.3 Definir ciclo de vida e clientes do produto.....	10
2.1.4 Identificar os requisitos dos clientes do produto.....	12
2.1.5 Definir os requisitos de projeto do produto.....	13
2.1.6 Definir as especificações de projeto do produto.....	13
2.2 Projeto conceitual.....	14
2.2.1 Verificação do problema.....	15
2.2.2 Análise funcional.....	15
2.2.3 Pesquisa por princípios de solução.....	16
2.2.4 Geração, seleção desenvolvimento e avaliação das variantes de concepção.....	16
2.3 Projeto preliminar.....	17
2.4 Projeto detalhado.....	18
2.5 Comentários.....	18
3. Modelagem e controle de sistemas automáticos.....	21
3.1 Rede Canal/Agência e o modelo estrutural e funcional de sistemas automáticos.....	22
3.2 Teoria de linguagem e autômatos e o modelo comportamental de sistemas automáticos.....	26
3.2.1 Linguagens.....	26
3.2.2 Autômatos.....	27
3.3 Teoria de controle supervisorio modular local.....	28

3.3.1 Controle supervisorio monolítico e modular	28
3.3.2 Representação por sistema produto (RSP)	29
3.3.3 Planta local.....	30
3.3.4 Supervisor modular local.....	31
3.4 Estrutura de controle e implementação.....	32
3.5 Exemplo de controle supervisorio	37
3.6 Projeto conceitual de SMMA.....	40
3.7 Comentários	44
4. Características do projeto de sistemas automáticos.....	45
4.1 Introdução	45
4.2 A influência da metodologia de projeto na qualidade dos resultados	45
4.2.1 O processo de projeto	46
4.2.2 O conhecimento no projeto	47
4.2.3 Qualidade do projeto e de seus resultados.....	48
4.2.4 Metodologia.....	49
4.2.5 Comentários	50
4.3 Particularidades entre sistemas técnicos e automáticos	50
4.3.1 Sistemas técnicos	51
4.3.2 Sistemas automáticos	51
4.4 Método de projeto de sistemas automáticos.....	53
4.5 Comentários	54
5. Modelagem da comunicação com o ambiente externo de um sistema automático.	57
5.1 Motivação	58
5.2 Comunicação simples entre um sistema automático e seu ambiente externo	59
5.3 Modelo funcional e estrutural para a comunicação com o ambiente externo ..	61
5.3.1 Observações finais.....	69
5.4 Modelo comportamental para a comunicação com o ambiente externo.....	70
5.4.1 Modelos para os pedidos do ambiente externo	71
5.4.2 Modelos para as respostas ao ambiente externo	73
5.4.2.1 Respostas relacionadas aos pedidos.....	74
5.4.2.2 Respostas não relacionadas aos pedidos.....	75
5.4.3 Observações finais.....	76
6. Aspectos de implementação em CLP	79
6.1 Programação em CLP.....	79
6.2 Métodos e ferramentas de cálculo do controle supervisorio	81
6.3 Estrutura de implementação de autômatos em diagrama escada	83
6.4 Supervisores	84

6.5	Estrutura de desabilitação de eventos controláveis.....	85
6.6	Estrutura de <i>reset</i> das variáveis de controle e variáveis de sinalização de ocorrência de eventos.....	85
6.7	Sistema produto.....	86
6.7.1	Transições não-controláveis	86
6.7.2	Transições controláveis	87
6.8	Interfaces de entrada e saída e seqüências operacionais.....	88
6.9	Observações importantes	90
7.	Projeto de um sistema exemplo.....	93
7.1	Introdução	93
7.2	Procedimento de projeto	93
7.3	Especificações de projeto	94
7.4	Modelagem do sistema energético/material	94
7.4.1	Modelagem estrutural e funcional	94
7.4.2	Modelagem comportamental da planta livre	95
7.4.3	Identificação de princípios de solução	98
7.4.4	Especificação do comportamento controlado	99
7.5	Modelagem da comunicação com o ambiente externo.....	100
7.5.1	Modelagem estrutural e funcional	100
7.5.2	Modelagem de comportamento relacionada aos pedidos	102
7.5.3	Modelagem de comportamento relacionada às respostas	103
7.5.4	Identificação de princípios de solução para pedidos e respostas.....	104
7.5.5	Especificação do comportamento controlado das respostas.....	105
7.5.6	Realização de especificações complementares	106
7.6	Cálculo do controlador	106
7.6.1	Gerar arquivos de modelos e especificações	106
7.6.2	Obter representação por sistema produto	107
7.6.3	Calcular e reduzir os supervisores modulares	107
7.7	Implementação em CLP.....	109
7.8	Resultados e conclusões	109
8.	Projeto de uma unidade de potência hidráulica.....	110
8.1	Introdução	110
8.2	Especificações de projeto	113
8.3	Modelagem do sistema energético/material	114
8.3.1	Modelagem funcional e estrutural	114
8.3.2	Modelagem comportamental.....	117
8.3.3	Especificação do comportamento controlado	120
8.4	Modelagem da comunicação com o ambiente externo.....	124

8.4.1 Modelagem de comportamento relacionada aos pedidos.....	125
8.4.2 Modelagem de comportamento relacionada às respostas.....	126
8.4.3 Especificação do comportamento controlado das respostas.....	129
8.5 Cálculo do controlador	135
8.5.1 Gerar arquivos de modelos e especificações	135
8.5.2 Obter representação por sistema produto.....	136
8.5.3 Calcular e reduzir os supervisores modulares	137
8.6 Implementação em CLP.....	139
8.7 Resultados e conclusões	140
8.7.1 Resultados	140
8.7.2 Conclusões	140
9. Conclusões	144
9.1 Perspectivas.....	145
Referências bibliográficas.....	148
Apêndices	152
1. Modelagem e operações simbólicas de SED, utilizando a ferramenta GRAIL....	154
2. Exemplos de uso do CTCT e ferramentas de convesão Grail-CTCT	156
3. Exemplos de utilização do CGLI	158
1. Anexo - Controle Supervisório Modular de Sistemas de Manufatura Discretos. .	162

LISTA DE FIGURAS

Figura 1.1 - Plataforma hidráulica proporcional.....	4
Figura 2.1 - Modelo de fases do projeto de produto (Modelo Consensual)(FORCELLINI, 2002).	7
Figura 2.2 - Projeto Informacional (FORCELLINI, 2002).....	8
Figura 2.3 - Proposta mínima de classificação de tipos de produto (FONSECA, 2000).	9
Figura 2.4 - Classificação de tipos de projeto (CONDOOR et al., 1992, apud FONSECA, 2000).	10
Figura 2.5 - Espiral de desenvolvimento (FONSECA, 2000).....	11
Figura 2.6 - Classificação resumida dos atributos do produto (FONSECA, 2000).....	11
Figura 2.7 - Ciclo de vida e suas necessidades (FONSECA, 2000).	12
Figura 2.8 - Conversão de necessidade em requisitos de cliente (FORCELLINI, 2002).	13
Figura 2.9 - Projeto Conceitual (adaptado de FORCELLINI, 2002).	14
Figura 2.10 - Tarefas e processos envolvidos na análise funcional (adaptado de FORCELLINI, 2002).	15
Figura 2.11 - Estabelecimento da estrutura de funções (PAHL e BEITZ, 1984 apud FERREIRA, 1997).	16
Figura 2.12 - Diagrama complexidade <i>versus</i> concreticidade do Projeto conceitual (FERREIRA, 1997).	17
Figura 2.13 - Etapas do projeto preliminar de sistemas hidráulicos de controle de posição (FURST e DE NEGRI, 2002).....	18
Figura 3.1 - Rede Canal/Agência (Rede C/A) (DE NEGRI, 1996).....	23
Figura 3.2 - Mecanismo de refinamento e condensação de redes C/A (DE NEGRI, 1996)..	23
Figura 3.3 - Rede C/A de um grupo Turbina-Gerador genérico (adaptado de PAES e DE NEGRI, 2002).	24
Figura 3.4 - Modelo funcional e estrutural, em rede C/A , de um sistema automático - (a) Condensado, (b) Refinamento em primeiro nível (DE NEGRI, 1996).	25
Figura 3.5 - Exemplo de autômato para uma máquina qualquer.	28
Figura 3.6 - Esquema de controle monolítico (QUEIROZ e CURY, 2000).	29
Figura 3.7 - Esquema de controle modular (QUEIROZ e CURY, 2000).....	29
Figura 3.8 - Relação dos conjuntos dos alfabetos (eventos) de um sistema composto de sub sistemas M1 á M6 e os módulos do sistema produto Md1, Md2 e Md3.	30
Figura 3.9 – Geração das plantas locais, composição dos módulos do sistema produto de acordo com as especificações (Ea, Eb e Ec).....	31
Figura 3.10 - Esquema de controle modular local (QUEIROZ e CURY, 2000).	31
Figura 3.11 - Estrutura básica de um sistema de controle (QUEIROZ et al, 2001).....	33
Figura 3.12 - Seqüência operacional modelada através de um SFC.	34
Figura 3.13 - Estrutura de interpretação de um Autômato em Diagrama Escada.....	35

Figura 3.14 - Interpretação de dois ou mais estados que levam a um estado.	35
Figura 3.15 - Representação do sistema produto onde $\alpha 1$ é controlável e $\beta 1$ é não controlável.	35
Figura 3.16 - Diagrama escada para o supervisor e estrutura de desabilitação.	36
Figura 3.17 - Exemplo de diagrama escada de uma seqüência operacional onde, (a) – Bloco de passos e (b) - Bloco de saídas.	37
Figura 3.18 - Sistema exemplo.	37
Figura 3.19 - Modelo das furadeiras por sistema produto (M1 e M2) e da especificação (E).	38
Figura 3.20 - Planta local.	38
Figura 3.21 - Supervisor modular local.	39
Figura 3.22 - Diagrama escada para o estado A1 e a desabilitação de $\alpha 1$	39
Figura 3.23 - Supervisor reduzido para o exemplo das duas furadeiras.	39
Figura 3.24 - Diagrama escada do sistema produto das furadeiras M1 e M2.	40
Figura 3.25 - Exemplo de SO para uma furadeira simples, em SFC.	40
Figura 3.26 - Projeto conceitual para SMMS (adaptado de SANTOS, 2003).	41
Figura 3.27 - Exemplo de modelo comportamental de uma agência (SANTOS, 2003).	42
Figura 5.1 - Comunicação entre usuário e um sistema controlado por computador.	58
Figura 5.2 - Modelo funcional/estrutural condensado de um sistema automático.	59
Figura 5.3 – Exemplo de seqüência operacional implementada em diagrama de funcionamento (SFC) – Entradas e saídas ao ambiente externo.	60
Figura 5.4 - Modelo em C/A de um sistema automático sem ambiente externo.	61
Figura 5.5 - Diagrama de blocos da estrutura de controle para a comunicação do sistema de informação e o ambiente externo.	63
Figura 5.6 – Modelo estrutural e funcional do sistema de informação para o ambiente externo.	64
Figura 5.7 – Modelo estrutural e funcional de um sistema automático estendido – caso de um usuário.	65
Figura 5.8 – Diagrama de blocos da estrutura de controle estendida ao ambiente externo.	66
Figura 5.9 – Modelo estrutural e funcional do sistema de informação – Processamento de informações.	67
Figura 5.10 – Modelo macro do usuário.	69
Figura 5.11 - Modelo comportamental para o pedido - duas opções independentes.	71
Figura 5.12 - Modelo comportamental para o pedido - duas opções mutuamente exclusivas.	72
Figura 5.13 - Modelo para o pedido - três opções independentes.	72
Figura 5.14 - Modelo para o pedido - três opções mutuamente exclusivas.	73

Figura 5.15 - Modelo da respostas ao ambiente externo relacionadas aos pedidos - três mensagens distintas.....	74
Figura 5.16 - Modelo da respostas ao ambiente externo relacionadas aos pedidos - duas mensagens distintas.....	75
Figura 5.17 - exemplos de modelos de respostas não relacionadas aos pedidos.....	75
Figura 6.1 - Bancada de simulação do controle de uma unidade de potência hidráulica (adaptada de SILVA NETO, 2004).	79
Figura 6.2 - Padronização da programação de CLP (FREY e LITZ, 2000).	80
Figura 6.3 - Canal/Agência do processo de projeto de controle lógico em CLP (adaptado de FREY e LITZ, 1998, 2000).	80
Figura 6.4 - Níveis de abstração de aplicações em CLP (MADER e WUPPER, 2000. tradução nossa).	81
Figura 6.5 - Estrutura do programa em CLP para o controle supervísório proposto.	83
Figura 6.6 – (a) Diagrama escada genérico para implementação de supervisores modulares originalmente em (b) autômatos.....	84
Figura 6.7 - Diagrama escada para a estrutura de desabilitação.....	85
Figura 6.8 - Diagrama escada para a estrutura de <i>reset</i>	86
Figura 6.9 – Diagrama-escada para implementação de transição não-controlável entre dois estados diferentes.	86
Figura 6.10 - Diagrama escada para implementação de transição não controlável - Autolaço (self-loop).....	87
Figura 6.11 - Diagrama escada para implementação de transição controlável.....	88
Figura 6.12 - SFC relativo a entrada de sinais de uma chave - (a) Informa quando a chave é ligada, (b) informa quando a chave é desligada.....	89
Figura 6.13 - Diagrama escada da interface de entrada de uma chave comum.....	90
Figura 7.1 - Representação em C/A do sistema energético/material.	95
Figura 7.2 - C/A refinado do sistema exemplo.	95
Figura 7.3 - Modelos comportamentais das sub funções de Ag1.....	96
Figura 7.4 - Autômato do comportamento global de Ag1.....	96
Figura 7.5 - Especificação preliminar do comportamento de Ag1.	97
Figura 7.6 - Novo modelo de comportamento da agência 1.....	97
Figura 7.7 - Modelo de comportamento da agência 2.	97
Figura 7.8 - Princípios de solução para os modelos de sub funções.	98
Figura 7.9 - Princípios de solução para Ag1'	98
Figura 7.10 - Especificações de acionamento de Ag1 e Ag2, relativas às escolhas dos usuários.	100
Figura 7.11 - Especificações de acionamento de Ag1 e Ag2, relativas aos comandos de ligar dos usuários.	100

Figura 7.12 - C/A do sistema de informação do projeto teste.....	101
Figura 7.13 - Modelos de comportamento dos pedidos dos usuários 1 e 2 - escolha de agências.	102
Figura 7.14 - Modelos de comportamento dos pedidos dos usuários 1 e 2 - comando liga/desliga.....	102
Figura 7.15 - Modelos de comportamento das respostas aos usuários u1 e u2.	104
Figura 7.16 - Especificações para as respostas a u1 e u2 - respostas em caso de pedido.	105
Figura 7.17 - Especificações para as respostas a u1 e u2 - determina o tipo de resposta.	105
Figura 7.18 - Especificações complementares para evitar acionamento parcial.....	106
Figura 8.1 - Desenho da unidade hidráulica de potência.	111
Figura 8.2 - Circuito hidráulico da unidade de potência.....	112
Figura 8.3 - Esquema do painel de comando para as bancadas (adaptado de SOUZA, 2003).....	113
Figura 8.4 - Modelo em rede C/A condensada da UPCH	115
Figura 8.5 - Entidade do circuito hidráulico importantes ao controle.	116
Figura 8.6 - Rede C/A refinada da UPCH.....	117
Figura 8.7 - Autômatos do conjunto de bombas e do filtro.	118
Figura 8.8 - Autômato do direcionador da bomba variável.....	118
Figura 8.9 - Autômato do direcionador da bomba fixa.....	118
Figura 8.10 - Autômato do direcionador 2.	119
Figura 8.11 - Autômato do direcionador 3.	119
Figura 8.12 – Autômato do acumulador.....	120
Figura 8.13 - Especificações da planta sem a comunicação com o ambiente externo.	121
Figura 8.14 - Especificação para preferência de BV no enchimento do acumulador.....	121
Figura 8.15 - Especificações geral e para a bomba variável.....	122
Figura 8.16 – Especificações para direcionador da bomba fixa e direcionador 2.	123
Figura 8.17 - Especificação do direcionador 3 e do acumulador.....	124
Figura 8.18 - Modelos dos pedidos do usuário 1.....	125
Figura 8.19 - Modelos das respostas relacionadas aos pedidos do usuário 1.....	127
Figura 8.20 - Modelos de respostas de erros críticos e recursos utilizados.....	128
Figura 8.21 - Especificações das respostas de BV, BF e AC.....	130
Figura 8.22 - Especificações para as respostas de erro de escolha.....	131
Figura 8.23 - Especificações para a resposta de erros críticos.....	132
Figura 8.24 - Especificações para as respostas de recursos utilizados.....	132
Figura 8.25 - Especificações para as respostas de estados da bancada (parada, partindo, operando).	133
Figura 8.26 - Especificações complementares.....	134

Figura 9.1 - Autômato modelado em arquivo Grail (adaptado de DA CUNHA e GARCIA, 2002).	154
Figura 9.2 - Tela de operação do CTCT.....	156
Figura 9.3 - Formato do arquivo LABEL.....	157
Figura 9.4 - Tela de seleção do modelo de CLP.....	158
Figura 9.5 - Tela de entrada dos arquivos dos supervisores.....	159
Figura 9.6 - Tela de configuração dos símbolos.....	159
Figura 9.7 - Tela de configuração das variáveis de representação dos estados.	160
Figura 9.8 - Tela de geração do código e da lista de declaração de variáveis.....	160

LISTA DE TABELAS

Tabela 1 - Métodos para busca de princípios de solução (FERREIRA, 1997 e FORCELLINI, 2002).	16
Tabela 2 - Síntese de modelos utilizados em engenharia (DE NEGRI, 2004).	21
Tabela 3 - Estimativa de estados e eventos do modelo comportamental.	96
Tabela 4 - Eventos que determinam o funcionamento das agências 1 e 2.	99
Tabela 5 - Eventos relativos aos modelos das respostas a u1 e u2.	103
Tabela 6 - Relação de modelos e especificações em arquivos Grail.	106
Tabela 7 - Representação por sistema produto do exemplo de teste.	107
Tabela 8 - Resultado dos supervisores locais e supervisores locais reduzidos.	108
Tabela 9 - Componentes do circuito hidráulico e suas funções.	111
Tabela 10 - Elementos componentes dos conjuntos importantes ao controle.	115
Tabela 11 - Eventos do sistema de atuação (SA).	119
Tabela 12 - Eventos dos modelos de pedido.	125
Tabela 13 - Eventos dos modelos de resposta.	128
Tabela 14 - Lista de arquivos Grail gerados para o cálculo do controlador.	135
Tabela 15 - Representação por sistema produto dos modelos do sistema da UPCH.	136
Tabela 16 - Resultado dos supervisores locais e supervisores locais reduzidos da UPCH.	137
Tabela 17 - Comparação entre métodos de projeto de SED - Álgebra de Boole e TCSML.	141
Tabela 18 - Principais funções GRAIL utilizadas.	155
Tabela 19 - Exemplos de operações com funções Grail.	155

SIMBOLOGIAS E ABREVIATURAS

ASCII	American Standard Code for Information Interchange
CLP	Controlador Lógico Programável
EE	Energia Elétrica
EH	Energia Hidráulica
EM	Energia Mecânica
E/S	Entrada e Saída
IHM	Interface Homem-Máquina
LASHIP	Laboratório de Sistemas Hidráulicos e Pneumáticos.
LED	Light Emitting Diode
LI	Lista de instruções
MS-DOS®	Microsoft Disk Operating System
PC	Personal Computer
RSP	Representação por Sistema Produto
SA	Sistema de Atuação
SAM	Sistema de Atuação-Medição
SED	Sistemas a Eventos Discretos
SFC	Sequential Function Charts
SM	Sistema de Medição
SMMA	Sistema de Manipulação e Montagem Automatizados
SP	Sistema de Pedido
SR	Sistema de resposta
SPR	Sistema de Pedido-Resposta
SO	Seqüência operacional
TCSML	Teoria de Controle Supervisório Modular Local
UFSC	Universidade Federal de Santa Catarina
UPCH	Unidade de Potência Hidráulica
\subseteq	Esta contido ou é igual a
//	Produto síncrono de linguagens
Σ	Alfabeto de uma linguagem
$\alpha, \beta, \gamma, \alpha_1, \beta_1 \dots$	Eventos
ε	Cadeia vazia
D	Modelo de comportamento de um SED através de linguagens
$D\alpha, D\gamma$	Variável de desabilitação dos eventos controláveis α e γ respectivamente
$f: X \times \Sigma \rightarrow X$	Função de transição que leva de um estado de X a outro qualquer de X através de cadeias de Σ .

G	Autômato (também denominado gerador)
L	Linguagem Gerada
L_m	Linguagem Marcada
$P, P_0, P_n, P_{n-1}, P_{n+1}$	Passos de um diagrama funcional seqüencial (SFC)
X	Conjunto finito de estados de um autômato
x_0	Estado Inicial
X_m	Conjunto de estados marcados ou finais, pertencentes a X
x_n, x_{n+1}	Estados quaisquer subseqüentes de um autômato da planta
x_{sn}, x_{sn+1}	Estados quaisquer subseqüentes de um supervisor

RESUMO

Este trabalho apresenta um estudo sobre os métodos, técnicas, ferramentas e modelos de projeto de sistemas automáticos e propõe um aperfeiçoamento de sua modelagem, sistematização de projeto e implementação. Com base no projeto conceitual de sistemas de manipulação e montagem automatizados (SMMA), estuda-se a modelagem estrutural e funcional de sistemas automáticos, em rede Canal/Agência, com a inclusão de novas entidades para comportar o conceito de modelagem do ambiente externo e a conseqüente proposição de novos modelos comportamentais para estas entidades, utilizando a teoria de linguagens e autômatos. Esta abordagem trata da comunicação do sistema com o ambiente externo, especialmente com usuários, de maneira mais sistemática que a abordagem tradicional e introduz os procedimentos de projeto e estrutura de implementação em CLP para o controle projetado através da teoria de controle supervísório modular local.

Muitos estudos sobre técnicas de projeto de diversas classes de sistemas automáticos têm sido realizados, como forma de estabelecer uma metodologia de projeto para estes sistemas, otimizando o projeto e contribuindo assim para o aquecimento das atividades industriais em todos os níveis. Para a classe de sistemas abordada neste trabalho, os sistemas a eventos discretos (SED), alguns avanços já foram realizados principalmente nos casos de controle do fluxo de matéria contável, entretanto as técnicas apresentadas nesta dissertação apresentam-se sob uma forma mais geral sendo baseadas em um sistema de controle do fluxo de matéria e energia de forma contínua. É demonstrado o projeto de um sistema hidráulico simples com teste e validação em bancada de simulação e posteriormente é apresentado o projeto do controle de uma unidade de potência hidráulica que alimenta uma bancada didática de ensino e pesquisa, no laboratório de sistemas hidráulicos e pneumáticos (LASHIP) da UFSC. Os resultados obtidos nos dois casos demonstram que a proposta é viável e que promove a melhor compreensão do problema, auxiliando nas atividades de projeto de sistemas automáticos.

ABSTRACT

A study is here presented on the methods, techniques, tools and models of automatic systems design and the optimization of their modeling, design systematization and implementation is proposed. Through the conceptual design of automatic manipulation and assembly systems (SMMA in portuguese), the structural and functional modeling of automatic systems is studied. The model utilized is Channel/Instance net with the inclusion of new entities to represent the interconnection with the external environment and thereby propose new behavioral models for these entities using the theory of languages and automats. With this approach the communication of the system with the external environment, especially with users, is dealt with in a more systematic way than the traditional methods. The design procedures and implementation structure in PLC for the control are achieved through the theory of local modular supervisory control.

Many studies on design techniques for diverse types of automatic systems have been carried out to establish a design methodology for these systems. In this way, design has been optimized contributing to an improvement in industrial activities at all levels. For the group of systems researched here, the discrete events systems (DES), some advances have previously been made mainly regarding flow control of countable products. However, the techniques presented in this study are more general being based on control of material and energy in a continuous way. The design of a simple hydraulic system with its testing and validation using a simulation facility installed in the laboratory of hydraulic and pneumatic systems (LASHIP) at UFSC were carried out. The results obtained demonstrate that the proposal is viable and they promote a better understanding of the problem, assisting in the activities of automatic systems design.

CAPÍTULO 1

INTRODUÇÃO

1.1 Cenário atual

O processo de projeto em engenharia é motivo de estudo de diversos pesquisadores em várias instituições ao redor do mundo, tendo originado vários modelos de sistematização de atividades, promovendo um embasamento teórico capaz de auxiliar engenheiros e técnicos nas atividades de projeto. Para o projeto de produtos e sistemas técnicos, diversas são as metodologias e as ferramentas propostas, destacando-se, no Brasil, a apresentada em BACK (1983). Tendo como base este trabalho, diversas instituições e profissionais têm se dedicado ao estudo das metodologias de projeto de produto e sistemas técnicos, destacando-se dentre elas o Núcleo de Desenvolvimento Integrado de Produtos (NeDIP) da Universidade Federal de Santa Catarina (UFSC).

Nota-se, que as metodologias de projeto, seja para sistemas técnicos ou software, dividem-se em fases com o objetivo de alcançar resultados específicos, começando pelo levantamento das informações necessárias ao desenvolvimento do projeto, passando pela elaboração de um conceito sobre o objeto alvo e terminando com uma documentação clara do que deverá ser o produto final do esforço de concepção. Os modelos existentes convergem para um modelo dito consensual, compreendido das fases de projeto informacional, projeto conceitual, projeto preliminar e projeto detalhado que, por sua vez, são compostas de etapas de desenvolvimento sistematizadas que conduzem a resultados intermediários tais como especificação, estrutura funcional, leiaute preliminar, documentação detalhada, entre outros.

Atualmente o modelo consensual está fortemente direcionado para o projeto de produtos, sendo abundantemente provido de ferramentas de apoio às suas etapas de desenvolvimento. No entanto, as etapas que compõem este modelo não são plenamente adequadas ao projeto de sistemas automáticos ou mecatrônicos, devido às características peculiares deste tipo de sistema, dificultando a obtenção de resultados ótimos, conforme tem sido investigado no LASHIP - Laboratório de Sistemas Hidráulicos e Pneumáticos (EMC/UFSC) a cerca de 10 anos.

A dificuldade em lançar mão de modelos estruturados do projeto de produtos para o caso de sistemas automáticos ocorre devido ao aspecto multidisciplinar e multitecnológico envolvido no âmbito da engenharia de controle e automação. Este aspecto é comentado por MAGALHÃES e ALMEIDA (1996) lembrando que: “A diversidade de processos passíveis de serem automatizados justifica plenamente o enorme espectro da atual engenharia de auto-

mação. Por seu turno, automatizar processos consideravelmente diferentes exige soluções francamente distintas, tanto em nível de informação como de tecnologia e energia.”

Visto que, “O projeto aplicado ao ramo da engenharia [...] é uma atividade tecnológica, estruturada e gerenciável, que visa a solução de problemas típicos da engenharia, voltada ao futuro e usando a criatividade.” (FONSECA, 2000) e lembrando que os sistemas automáticos estão inseridos no contexto da engenharia, pode-se dizer que recursos e modelos de auxílio ao projeto deste tipo de sistema também devem ser eficazes como no caso dos produtos e sistemas técnicos em geral. O projeto de sistemas automáticos tem sido conduzido tradicionalmente de acordo com as orientações dadas pelos fabricantes de hardware e software aplicados a automação industrial (DE NEGRI, 1996) ou com metodologias específicas a certos tipos de acionamentos, como por exemplo o projeto de sistemas automáticos com acionamentos pneumáticos (BOLLMANN, 1996). De maneira geral os métodos de projeto de sistemas automáticos levam em consideração a pré-existência do sistema físico, para que se projete o sistema de controle do dispositivo, não sendo moldados para o caso de projeto de novos sistemas.

Assim como no caso do conflito industrial entre setores de engenharia e de processos e produção, objeto de atuação da engenharia simultânea, o projeto de sistemas automáticos tem sido tipicamente o projeto da automatização e/ou controle de um sistema preexistente, seja ele mecânico, elétrico, químico etc... Isto quer dizer que o sistema é geralmente concebido *a priori*, normalmente com um esforço de otimizar suas funcionalidades e custos o que não significa que o projeto posterior da automatização e/ou controle será simplificado, sendo possível até que o problema de controle não tenha solução ou os custos de levá-lo a cabo sejam proibitivos.

Uma tentativa de resolver esse problema foi proposta por SANTOS (2003) que, tomando como referência o modelo consensual estabelecido para produtos e sistemas técnicos, formulou um novo modelo de projeto conceitual para o caso de sistemas de manipulação e montagem automatizados (SMMA), procurando abordar a execução simultânea dos projetos da parte física e do controle de forma a obter uma concepção do novo sistema de maneira integrada. Para tanto, SANTOS apresenta uma nova proposta de descrição funcional, estrutural e comportamental utilizando a rede Canal Agência (C/A) e modelos de sistemas a eventos discretos (SEDs) baseado em autômatos e, desta forma, inseriu na atividade de projeto ferramentas formais de modelagem e síntese de controladores (teoria de controle supervísório), estudadas pelo Grupo de Pesquisa em Sistemas a Eventos Discretos (DAS/UFSC). Outra contribuição interessante deste trabalho foi a criação de um arcabouço de especificações operacionais aplicáveis a sistemas de manipulação e montagem automatizados, contribuindo para a agilização da atividade de projeto, visto que o projetista terá a sua disposição uma biblioteca de modelos de especificações de acordo com a configuração do sistema a ser projetado.

As ferramentas de modelagem utilizadas por SANTOS também vêm sendo alvo de estudo para realização de melhorias nos aspectos de modelagem ou na interface com o projetista, entre outras. Os exemplos vão desde as pesquisas realizadas sobre o método consensual e suas ferramentas de auxílio, passando pelos modelos de SEDs que têm evoluído juntamente com as técnicas de síntese de controladores, além dos modelos estruturais e funcionais em rede Canal/Agência.

Pode-se dizer que todo o esforço descrito anteriormente tem por objetivo alcançar a maturidade em métodos, modelos e ferramentas de análise e/ou síntese, amplos o bastante para englobar a maior gama possível de áreas científicas e tecnológicas do contexto da automação industrial, de tal sorte que se torne possível a elaboração de uma ferramenta de projeto, a exemplo das existentes em outras áreas como as de CAD. Uma ferramenta CAD, embora não dispense o conhecimento básico sobre técnicas de desenho técnico, deixa transparente ao projetista todos os métodos, cálculos, algoritmos e conhecimentos complexos envolvidos na formulação e manipulação de desenhos em 3D, deixando-o livre para exercitar seu conhecimento sobre o objeto do desenho e assim estimular a criatividade, entre outras vantagens.

Uma ferramenta de projeto de sistemas automáticos poderá deixar oculto ao engenheiro projetista detalhes sobre a modelagem em autômatos e a síntese de controladores para SEDs, por exemplo, fazendo com que ele preocupe-se mais com as especificações e soluções para resolvê-las do que com modelos matemáticos, algoritmos de cálculo e complexidades e limitações computacionais.

1.2 Objetivos

Balizado pelos esforços já realizados e resultados alcançados, principalmente pelos grupos de pesquisa nos quais este trabalho está inserido, tomar-se-á por ponto de partida especialmente o trabalho realizado por SANTOS (2003) sobre projeto conceitual de SMMA, ou seja, o presente trabalho baseia-se no modelo consensual de projeto em engenharia e nas ferramentas de modelagem estrutural, funcional e comportamental utilizadas naquele trabalho.

Desta forma, tem-se então como objetivo geral contribuir com o esforço coletivo de formulação e aprimoramento de metodologias, técnicas e ferramentas que auxiliem aos projetistas de sistemas automáticos em sua tarefa projetar equipamentos de maneira cada vez mais rápida, com custo menor e qualidade total. Para tanto, estudam-se os resultados obtidos até o momento e discutem-se os aspectos passíveis de melhoria, correção e adequação, sobretudo no que diz respeito à utilização do projeto conceitual de SMMA para sistemas automáticos mais gerais e que contemplem também a comunicação complexa com o ambiente externo ao sistema.

Entre os objetivos específicos está, a realização da contextualização clara desta linha de pesquisa no âmbito do projeto de equipamentos industriais, através de uma discussão sobre o papel dos métodos e técnicas na atividade de projeto de sistemas automáticos, de modo a explicar ao pessoal técnico não ligado diretamente a esta linha de trabalho, a relevância e abrangência de uma metodologia de projeto de sistemas automáticos.

Como principal objetivo específico, tem-se a solução de um problema de controle real, diferente de um SMMA, que foi utilizado como fonte de informação, estudo, levantamento de dúvidas, teste de técnicas existentes, teste de novas propostas de modelagem e novas abordagens conceituais. O estudo deste problema real e o esforço para alcançar uma solução coerente proporcionaram uma visão nova deste tipo de problema, permitindo o surgimento de perspectivas e idéias jovens que serão apresentadas no decorrer deste documento.

Este problema real se trata do projeto de uma plataforma hidráulica proporcional, composta de duas bancadas de trabalho didáticas, sistema de projeto e controle de circuitos hidráulicos em computador, para execução de experiências, e uma unidade de potência hidráulica que fornece recursos de energia às bancadas de ensino. A Figura 1.1 ilustra esquematicamente o sistema.

Plataforma Hidráulica Proporcional

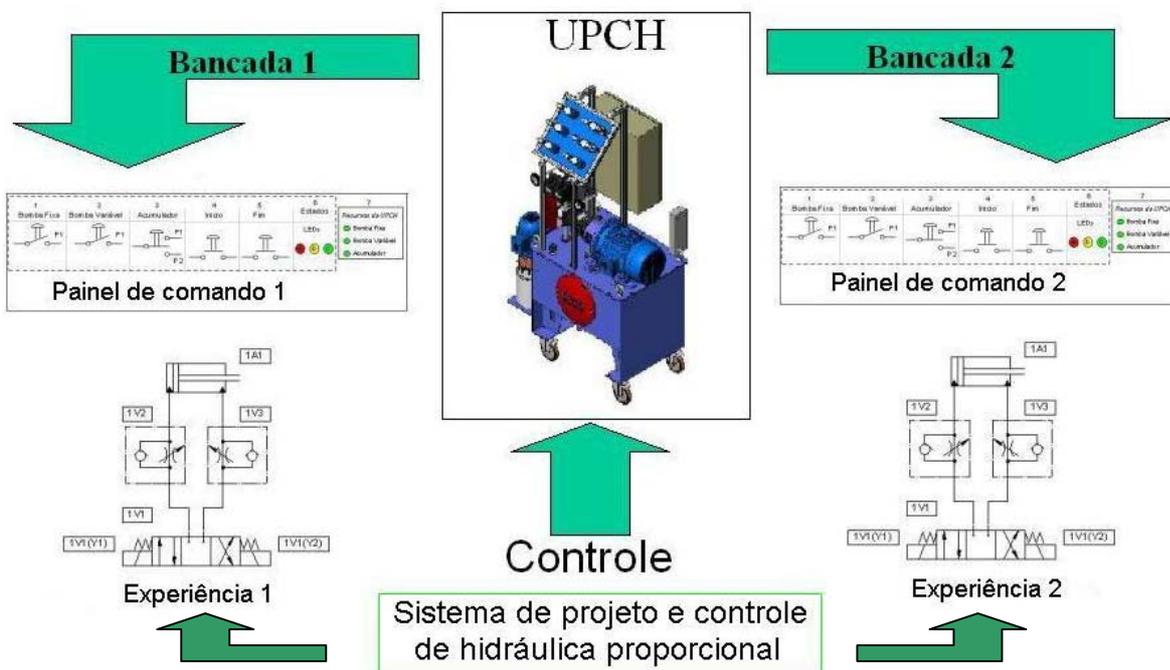


Figura 1.1 - Plataforma hidráulica proporcional.

A solução que será mostrada no decorrer deste trabalho, diz respeito ao controle da unidade de potência hidráulica (UPCH) e sua comunicação com as bancadas de trabalho,

que do ponto de vista da UPCH vazem parte do ambiente externo, através dos painéis de comando. Será visto que se tem então um problema de controle de fluxo de energia, com características discretas, recursos limitados e compartilhados, além é claro de uma comunicação com o ambiente externo, caracterizando este problema como bastante adequado ao estudo para proposta de novas técnicas de projeto de sistemas automáticos.

Desta forma, o que se quer alcançar como mais um objetivo específico é a criação de uma modelagem formal para a comunicação com o ambiente externo, de modo a auxiliar às atividades de projeto de sistemas automáticos quando está presente a necessidade de uma comunicação complexa com outros sistemas ou operadores humanos, ou seja, usuários.

1.3 Estrutura do documento

Este trabalho será apresentado de maneira a ambientar o leitor no contexto da engenharia de projeto, mais particularmente no projeto de sistemas automáticos, a fim de facilitar o entendimento das propostas que virão na seqüência. Para alcançar este objetivo, os capítulos seguintes contemplarão uma revisão dos conceitos, métodos e técnicas abordados, uma discussão sobre a abrangência de uma metodologia de projeto de sistemas automáticos, a proposta de novos artifícios para a solução de problemas de projeto, um exemplo com a nova proposta, a discussão dos resultados e as conclusões e perspectivas.

Realizam-se então, nos capítulos 2 e 3, as revisões das áreas de conhecimento envolvidas nesta pesquisa, abrangendo o método de projeto de produtos industriais que utiliza o modelo consensual, a modelagem e controle de sistemas automáticos principalmente para o caso de SED e o projeto conceitual de SMMA.

No Capítulo 4, faz-se uma discussão sobre os aspectos envolvidos no projeto de sistemas automáticos e as justificativas de se investir na elaboração de técnicas e ferramentas que permitam evoluir o conhecimento existente sobre o projeto de sistemas automáticos a uma metodologia de projeto para esta classe de sistemas. Avalia-se neste capítulo, os impactos sobre a qualidade dos resultados diante da utilização de sistemáticas de projeto adequadas aos objetivos almejados pelos projetistas.

As propostas deste trabalho estão explanadas nos capítulos 5 e 6, em que primeiramente é realizada a proposta de aperfeiçoamento dos modelos que representam a estrutura, função e comportamento, com base nos modelos já existentes para que a comunicação com o ambiente externo seja contemplada. Na seqüência, as estruturas necessárias para a implementação desta proposta em CLP são também apresentadas.

Para demonstrar a viabilidade das propostas, um exemplo é apresentado no capítulo 7 juntamente com uma proposta de sistematização de projeto. Neste exemplo, os passos da sistemática proposta são seguidos e os resultados são gradativamente apresentados, finalizando com um teste e validação em uma bancada de simulação. No capítulo 8, é apresen-

tado o projeto de uma unidade de potência hidráulica utilizando as mesmas técnicas do exemplo do capítulo 7, onde se pode avaliar os resultados desta aplicação diante de um problema real de controle.

Fechando este documento, são apresentadas as conclusões sobre a relevância das propostas e dos resultados, além de citadas algumas perspectivas sobre trabalhos futuros.

CAPÍTULO 2

MODELO CONSENSUAL

A pesquisa realizada mundialmente para atingir a sistematização adequada das atividades de projeto de produtos resultou em alguns tipos de modelos dentre os quais destaca-se , segundo FORCELLINI (2002) e FONSECA (2000), o modelo de fases seguido por vários autores. Avaliando os modelos de fase existentes, FERREIRA (1997) conclui que: “De uma maneira geral, constata-se uma **grande similaridade** entre os modelos do processo de projeto [...]. Não parece haver grandes diferenças nas atividades indicadas nem na seqüência seguida. Pequenas diferenças encontram-se na distribuição das atividades entre as fases.” (grifo nosso). Desta forma FERREIRA cunhou para o modelo de fases o termo “Modelo Consensual”, composto das fases de esclarecimento da tarefa, projeto conceitual, projeto preliminar e projeto detalhado. A fase de esclarecimento da tarefa foi posteriormente organizada e sistematizada dando origem ao projeto informacional (FONSECA, 2000), caracterizando o modelo consensual na forma que este é divulgado hoje (Figura 2.1).

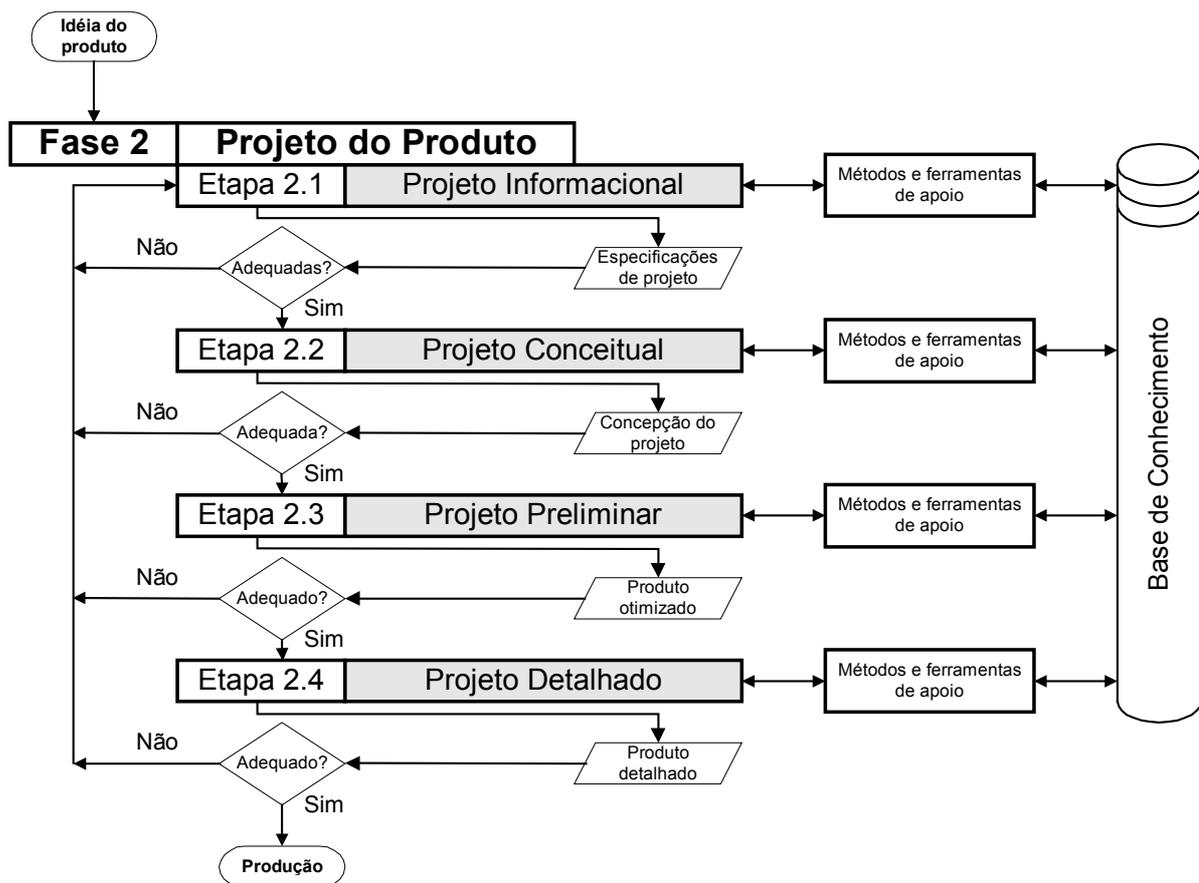


Figura 2.1 - Modelo de fases do projeto de produto (Modelo Consensual)(FORCELLINI, 2002).

As atividades descritas na Figura 2.1 estão arranjadas de modo a gerar um ganho constante de informações, documentação das ações e organização da seqüência de passos

e ferramentas utilizadas, facilitando o entendimento de todo o processo e sua realimentação para modificações, adequações ou correções. As fases 1, 3 e 4, que não estão descritas na figura, estão relacionadas à definição, produção e o lançamento e acompanhamento do produto durante sua permanência no mercado, respectivamente.

Será realizada uma revisão do método de projeto que se utiliza no modelo consensual, com maior ênfase nas fases de projeto informacional e conceitual em virtude dos objetivos deste trabalho. A informações contidas neste capítulo não devem ser tomadas como única fonte de estudo para a realização das etapas de projeto, visto que devido à complexidade e volume de informações, bem como o grande número de ferramentas utilizadas, o tema encontra-se resumido aqui, sendo aconselhável uma consulta às obras referenciadas.

2.1 Projeto Informacional

O problema de projeto de um produto ou sistema técnico tem o objetivo de atender a uma ou mais necessidades através da materialização de um resultado real e tangível de uma solução. Para tanto, primeiramente deve-se possuir o máximo conhecimento possível sobre quais são as necessidades a serem atendidas bem como quais informações são relevantes para o entendimento integral do problema.

As ações necessárias ao levantamento do conhecimento útil às atividades de projeto são sistematizadas na fase de Projeto Informacional, ilustradas na Figura 2.2.

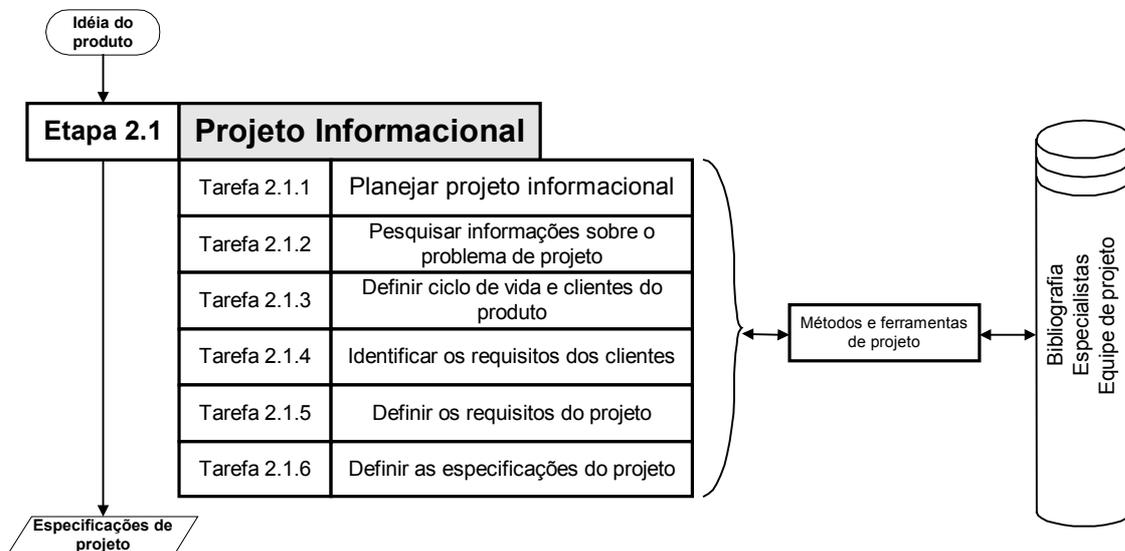


Figura 2.2 - Projeto Informacional (FORCELLINI, 2002).

Pode-se notar que esta fase inicia com uma idéia do produto e evolui de uma necessidade para uma lista de especificações de projeto que é a base para as demais fases do trabalho.

2.1.1 Planejar o projeto informacional

Esta atividade visa organizar a execução das demais atividades do projeto informacional e não está ligada diretamente à obtenção de resultados para o produto em si. A criação da equipe de trabalho, divisão das tarefas, definição de responsabilidades são, entre outras, algumas das atividades que podem ocorrer no planejamento, assim como a identificação preliminar de recursos que podem ou devem ser necessários durante as atividades futuras.

2.1.2 Pesquisar informações sobre o problema de projeto

Para organizar a pesquisa de informações sobre o problema, FONSECA (2000) sugere o desenvolvimento de três tarefas: Analisar o problema de projeto, procurar a informação necessária para o trabalho de projeto e definir os produtos concorrentes. De maneira resumida pode-se descrever estas tarefas como segue

2.1.2.1 Análise do problema de projeto

Clarificar os objetivos, ou seja, ambientar-se com o problema e reunir o maior volume possível de informações sobre o assunto. O guia de informações mínimas para o início do projeto deve contemplar: estudo de marketing prévio, classificação do tipo de produto (Figura 2.3); Classificação do tipo de projeto pelo *modelo de Jansson* (Figura 2.4), planejamento do volume de fabricação, desejos explícitos expostos no problema de projeto e restrições do projeto ou do produto.

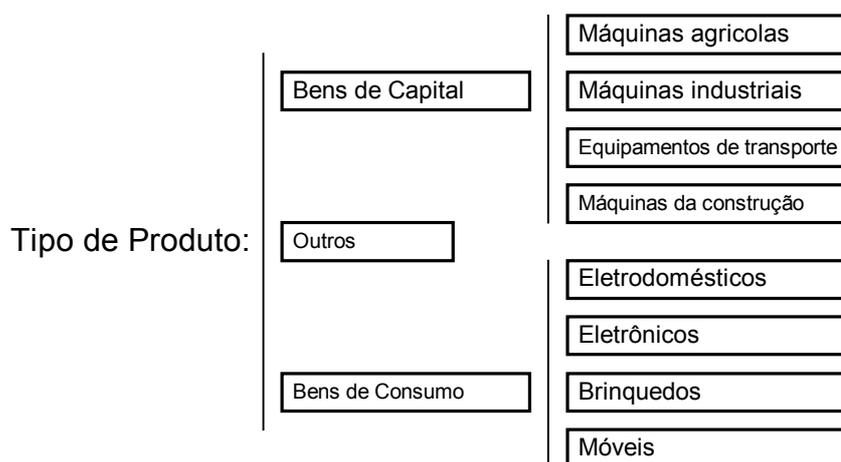


Figura 2.3 - Proposta mínima de classificação de tipos de produto (FONSECA, 2000).

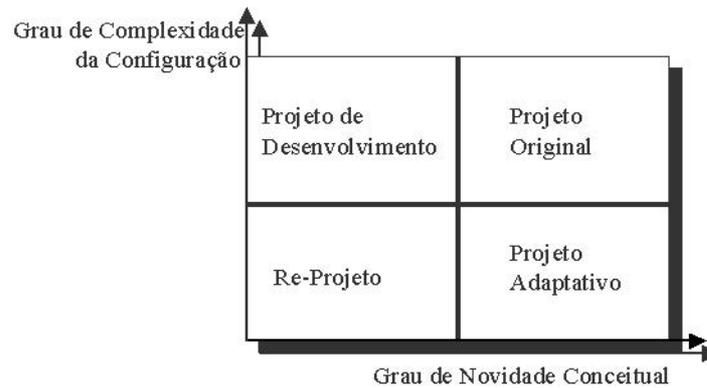


Figura 2.4 - Classificação de tipos de projeto (CONDOOR et al., 1992, apud FONSECA, 2000).

2.1.2.2 Pesquisa de informações necessárias ao trabalho de projeto

Procurar por patentes sobre o produto que vai ser projetado, tecnologias e métodos de fabricação disponíveis e informação sobre produtos similares. Estas atividades de pesquisa podem ser realizadas simultaneamente utilizando-se da Internet, o instituto nacional de patentes (INPI) ou de seu similar no país em que o produto vai ser produzido, além de catálogos, folhetos e informativos de concorrentes para o caso de produtos similares.

2.1.2.3 Pesquisa de produtos concorrentes ou similares

Com base nas informações de produtos similares da tarefa anterior pode-se obter informações de quais devem se configurar como modelos concorrentes e dentre estes quais são líderes de mercado por qualidade ou preço, por exemplo.

Os resultados da tarefa descrita no item 2.1.2 devem ser: **Documento de ordem de projeto**, definição dos **objetivos do projeto**, identificação de **produtos concorrentes e patentes relacionadas e tecnologias viáveis de fabricação**.

2.1.3 Definir ciclo de vida e clientes do produto

Esta tarefa objetiva identificar os atributos e o ciclo de vida que o produto terá, assim como identificar quais serão os clientes que ele deve atingir. FONSECA também divide esta tarefa em três atividades.

- Estabelecer as fases do ciclo de vida do produto. A Figura 2.5 lustra o ciclo de vida baseado na espiral de desenvolvimento;
- Definir os clientes e usuários: Identificar aqueles que serão afetados ou terão alguma relação com o produto que será projetado;
- Definir os atributos do produto. A Figura 2.6 ilustra uma proposta resumida de como classificar os atributos do produto.

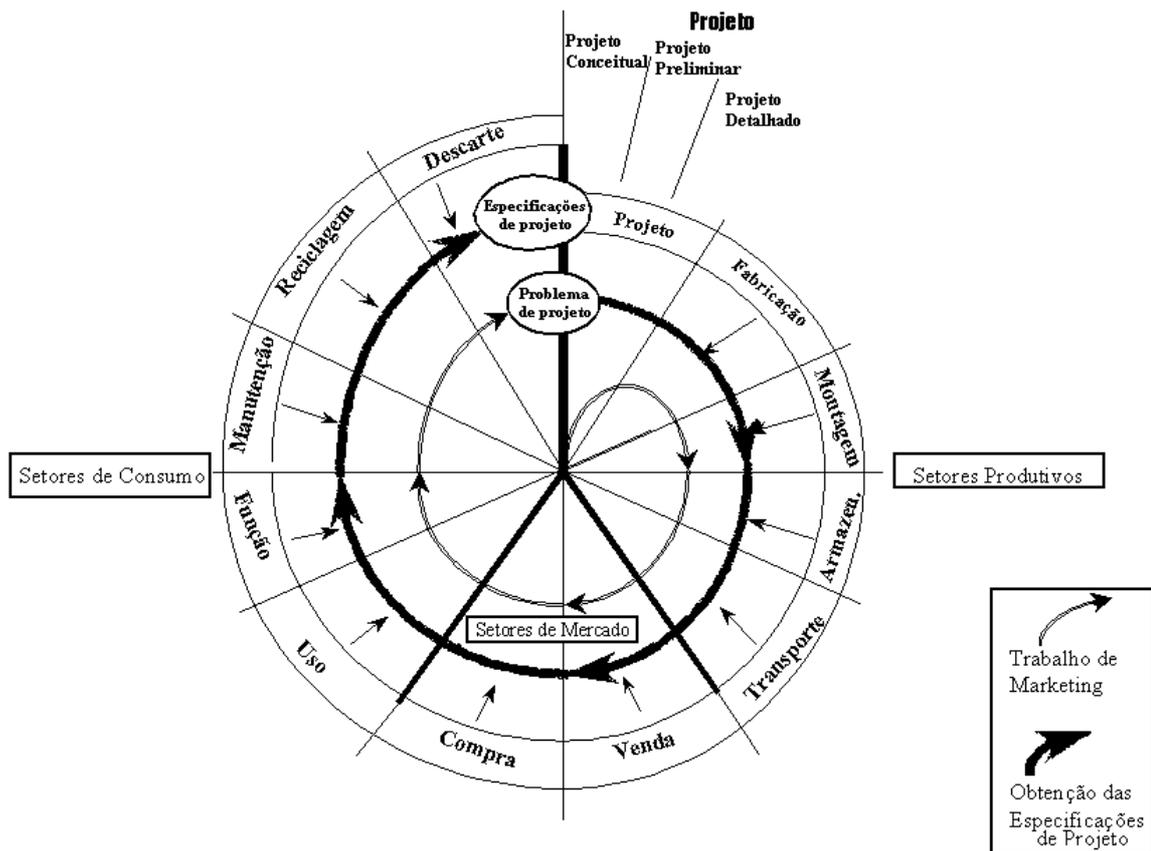


Figura 2.5 - Espiral de desenvolvimento (FONSECA, 2000).

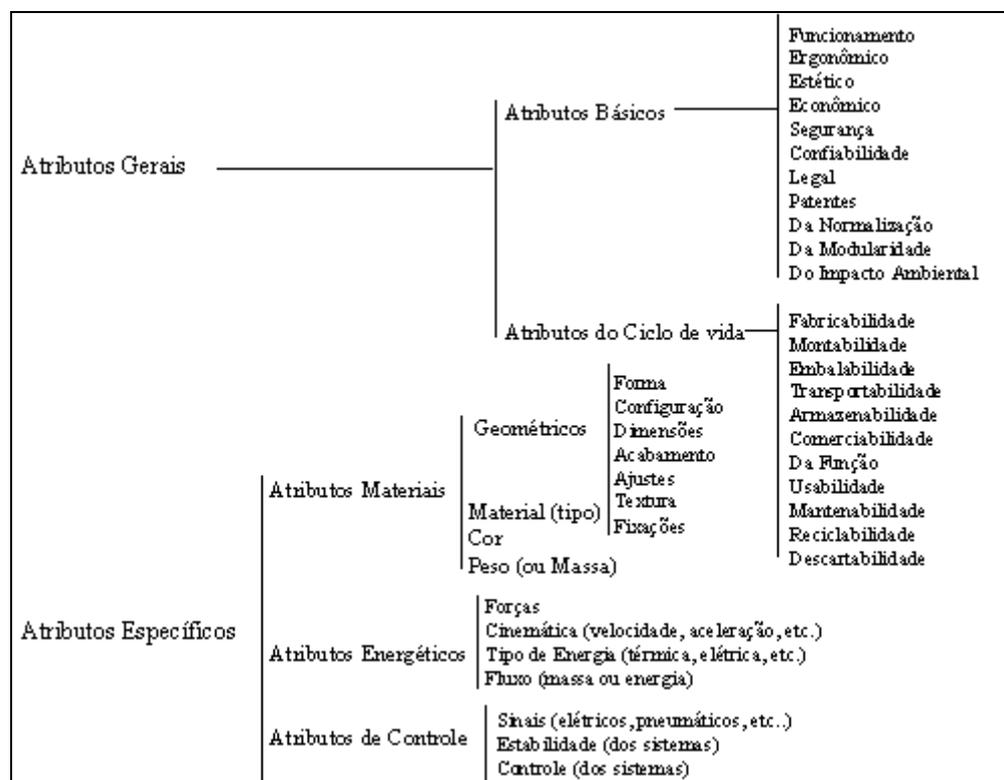


Figura 2.6 - Classificação resumida dos atributos do produto (FONSECA, 2000).

2.1.4 Identificar os requisitos dos clientes do produto

FORCELLINI agrupa em três atividades esta tarefa, baseado na tarefa proposta por FONSECA, a saber:

- **Coletar as necessidades dos clientes de cada fase do ciclo de vida:** O gráfico da Figura 2.7 ilustra o as necessidades de cada cliente durante o ciclo de vida do produto;
- **Agrupar e classificar as necessidades obtidas;**
- **Analisar e definir os requisitos dos clientes:** Converter as necessidades em requisitos de usuários. A Figura 2.8 exhibe resumidamente esta atividade.

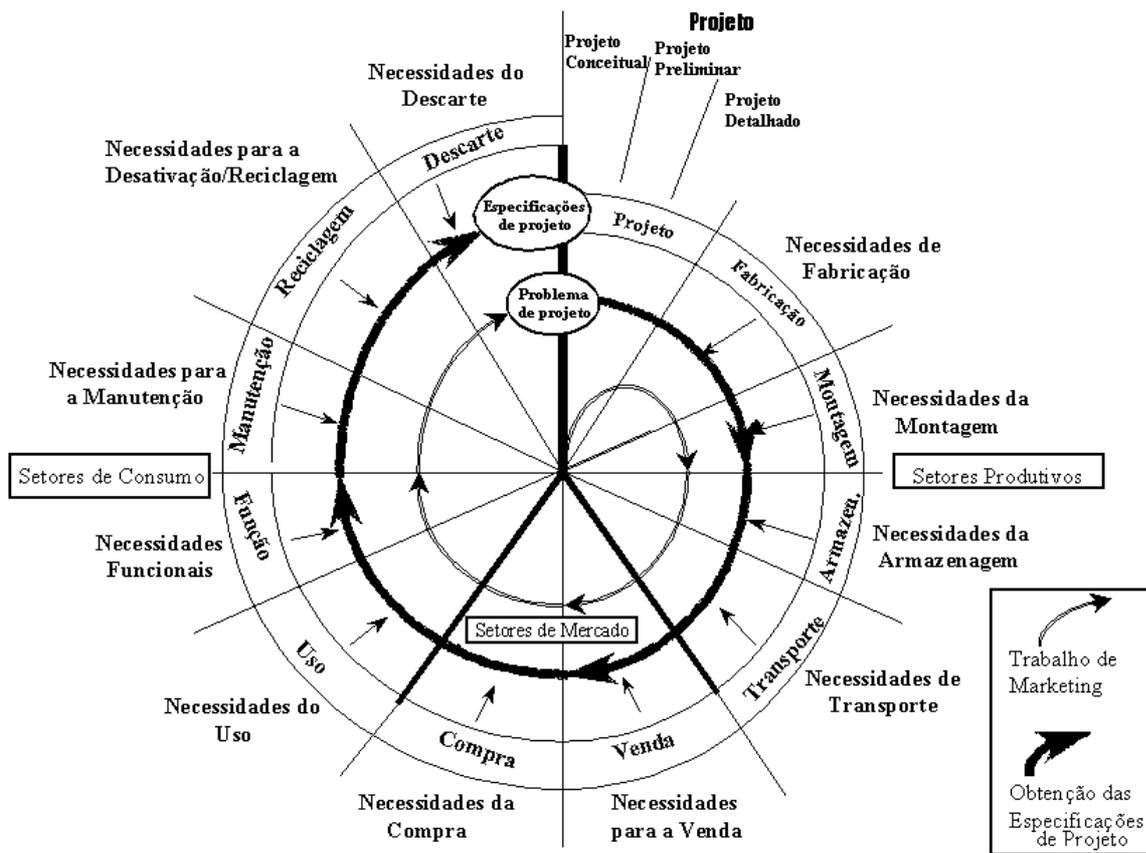


Figura 2.7 - Ciclo de vida e suas necessidades (FONSECA, 2000).

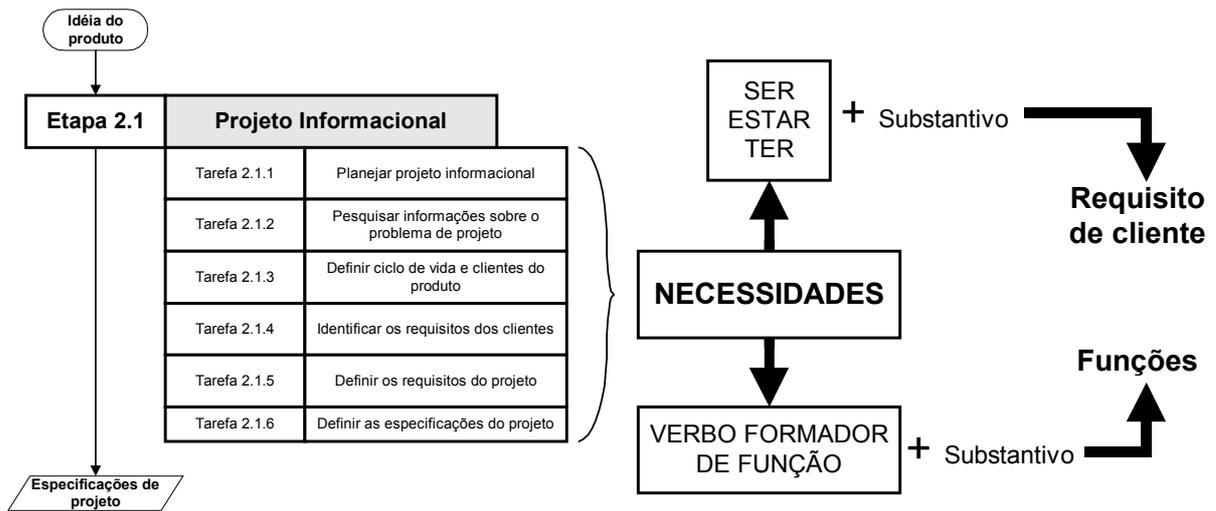


Figura 2.8 - Conversão de necessidade em requisitos de cliente (FORCELLINI, 2002).

2.1.5 Definir os requisitos de projeto do produto

Esta tarefa é dividida em duas atividades com o objetivo de transformar as informações vindas dos requisitos dos clientes, geralmente vagas e subjetivas, em características mensuráveis de acordo com uma linguagem técnica de engenharia, permitindo uma comunicação precisa entre os profissionais envolvidos no processo de projeto.

- **Converter requisitos de usuário em expressões mensuráveis:** Transformar os requisitos em informações técnicas;
- **Definir e classificar os requisitos de projeto:** Classificar os requisitos obtidos utilizando o método QFD, mais conhecido como Casa da Qualidade (HAUSER e CLAUSING, 1988, apud FORCELLINI, 2002).

2.1.6 Definir as especificações de projeto do produto

- **Confrontar os requisitos de projeto com o problema de projeto:** retomar a idéia inicial do problema que deu origem ao projeto como forma de avaliar quais elementos importantes devem ser incluídos ou retirados;
- **Incluir metas, objetivos e restrições:** Incluir nas especificações as diretivas explícitas vindas do problema original e do estudo de marketing e expor abertamente as metas a serem atingidas e as restrições impostas ao projeto e produto;
- **Definir as especificações de projeto:** as especificações de projeto são então constituídas pelo conjunto de requisitos selecionados como especificações, metas a serem atingidas, restrições gerais e da descrição do produto que será projetado.

2.2 Projeto conceitual

Com a fase de projeto conceitual, as especificações de projeto da fase informacional são trabalhadas de modo a se obter uma concepção do produto ou sistema técnico que representa fundamentalmente a solução para o problema de projeto sob a forma de uma função global. Esta fase é tida como a mais importante do processo de projeto em virtude das decisões tomadas nela influenciarem fortemente as atividades das demais fases de projeto.

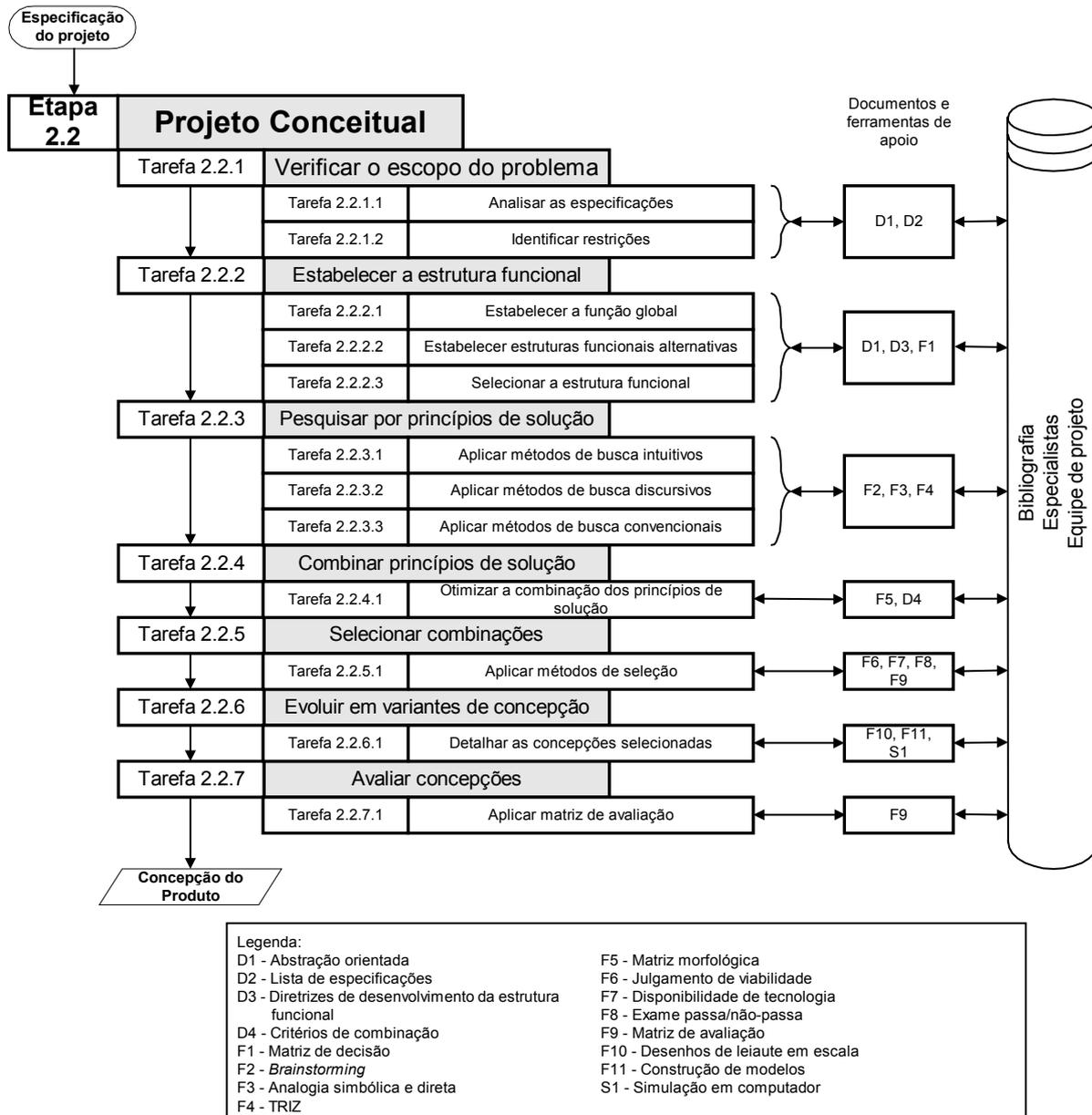


Figura 2.9 - Projeto Conceitual (adaptado de FORCELLINI, 2002).

A Figura 2.9 ilustra as atividades e ferramentas comuns à fase de projeto conceitual. Nota-se que se trata de uma fase criteriosa e complexa, envolvendo diversas ferramentas de auxílio para que um problema apresentado de forma abstrata tenha sua solução concretizada na forma de um conceito. Para otimizar a apresentação do assunto, FORCELLINI agrupa as atividades de projeto conceitual em quatro partes: Verificação do problema, análise

se funcional, pesquisa por princípios de solução e geração, seleção desenvolvimento e avaliação das variantes de concepção.

2.2.1 Verificação do problema

Realiza-se um estudo do problema através de sua abstração, ou seja, através da observação de aspectos gerais ignorando-se o que é particular. As especificações de projeto são então analisadas, ajudando na identificação de restrições fictícias e, as vezes, também na sua eliminação.

2.2.2 Análise funcional

Nesta fase, a abstração do problema é realizada através da identificação de sua estrutura de funções, começando pela função global e decompondo em funções parciais até o ponto em que seja possível identificar operações elementares do sistema, como mostrado na Figura 2.10.

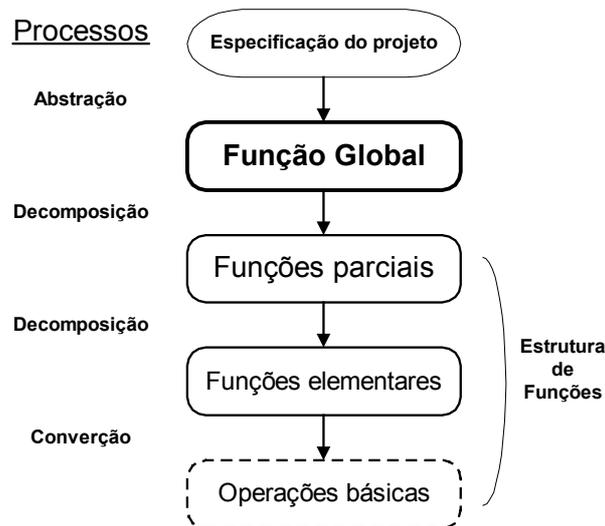


Figura 2.10 - Tarefas e processos envolvidos na análise funcional (adaptado de FORCELLINI, 2002).

Como função entende-se a relação entre a entrada e a saída de um sistema através da realização de uma tarefa, sendo que esta função pode ser composta por outras subfunções a exemplo da ilustração da Figura 2.11.

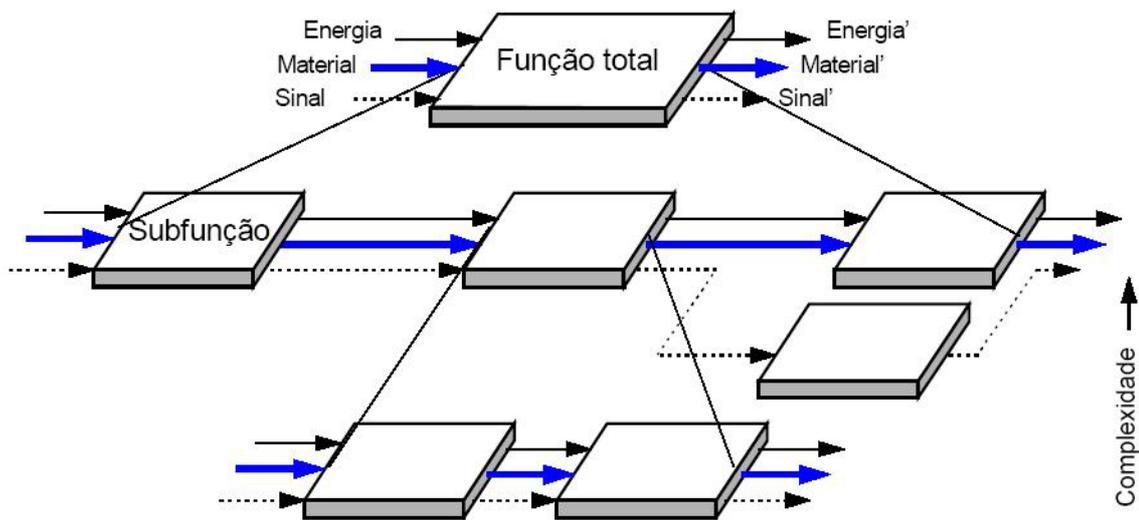


Figura 2.11 - Estabelecimento da estrutura de funções (PAHL e BEITZ, 1984 apud FERREIRA, 1997).

O objetivo da decomposição funcional é facilitar a busca por soluções que antes seria para a função global, passando a ser para as subfunções.

2.2.3 Pesquisa por princípios de solução

A busca por princípios de solução promove a passagem do projeto do campo abstrato para o campo concreto, em que para cada uma das subfunções identificadas é associada um ou mais princípios de solução. Um princípio de solução é a materialização de uma função em um sistema ou subsistema físico, na forma de uma representação esquemática idealizada de maneira qualitativa. A pesquisa por princípios de solução é classificada em três tipos e os métodos relacionados estão descritos na Tabela 1.

Tabela 1 - Métodos para busca de princípios de solução (FERREIRA, 1997 e FORCELLINI, 2002).

Classificação	Métodos
Convencionais	Pesquisa bibliográfica, análise de sistemas naturais, análise de sistemas técnicos, analogias, medições e testes em modelos.
Intuitivos	<i>Brainstorming</i> , Método 635, Método Delphi, Sinergia, analogia direta, analogia simbólica, combinação de métodos
Discursivos	Estudo sistemático de sistemas técnicos, Uso de esquemas de classificação, Uso de catálogos de projeto, TRIZ, matriz morfológica.

2.2.4 Geração, seleção desenvolvimento e avaliação das variantes de concepção

Com base nos princípios de solução encontrados, são realizadas combinações destes com o objetivo de se atender à função global, utilizando ferramentas como a matriz morfológica para auxiliar a combinação adequada, as quais são então selecionadas lançando-se

mão também de métodos de seleção. As combinações mais promissoras devem ser desenvolvidas em variantes de concepção que são posteriormente avaliadas de modo a orientar a tomada de decisão quanto à solução mais coerente que será escolhida como a melhor concepção.

Como mostrado, o projeto conceitual é uma fase em que o problema é trabalhado de maneira abstrata no início, decomposto para melhor entendimento e solução, o que o caracteriza em aspectos mais concretos que quando recompostos emergem na forma de uma concepção global de solução. Esta evolução é mais bem apresentada na ilustração da Figura 2.12.

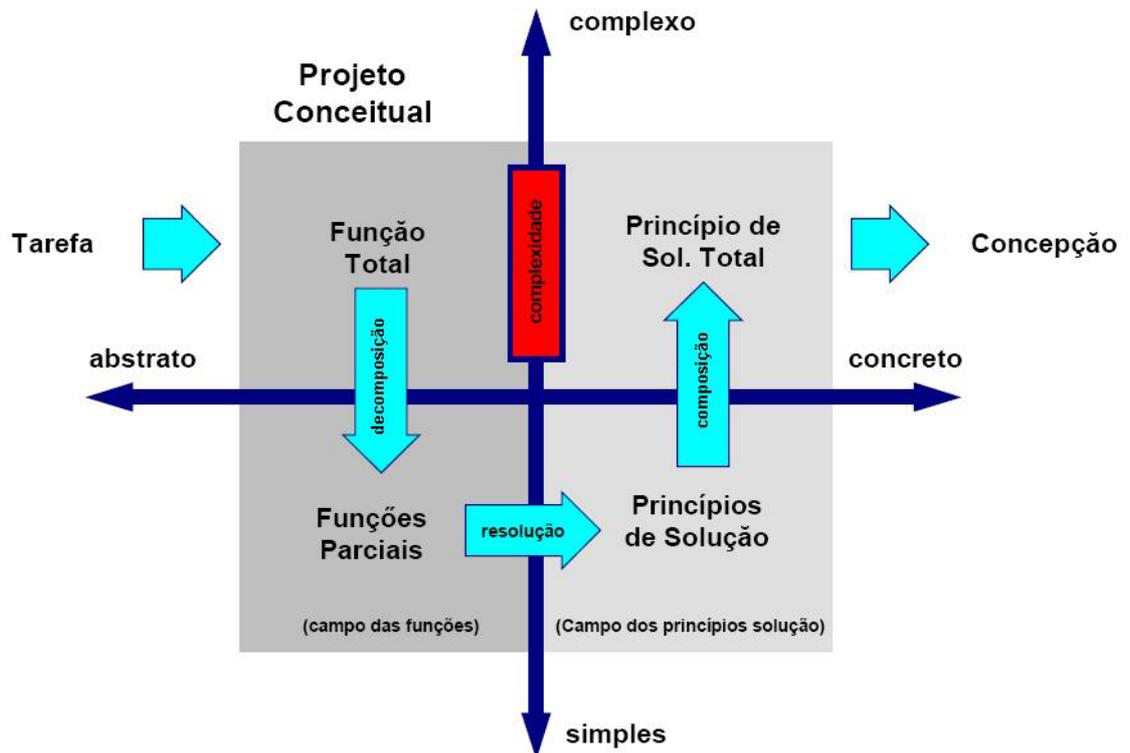


Figura 2.12 - Diagrama complexidade *versus* concreticidade do Projeto conceitual (FERREIRA, 1997).

2.3 Projeto preliminar

Partindo do conceito do sistema, gerado na fase de projeto conceitual, na fase seguinte de projeto preliminar é gerado o leiaute otimizado do sistema. As tarefas tradicionais do projeto preliminar são descritas por FORCELLINI (2002):

- Elaborar leiautes e formas para as funções principais;
- Elaborar leiautes e formas para as funções auxiliares;
- Otimização final do leiaute.

Como nas fases anteriores, o projeto preliminar também é composto de diversas tarefas e ferramentas de auxílio, cada qual adequada a certas áreas de atuação. Por exemplo, as atividades do projeto preliminar poderão ser adaptadas, para o caso do projeto de siste-

mas automáticos em pneumática, de maneira semelhante ao descrito por FURST e DE NEGRI (2002), como mostrado na Figura 2.13.

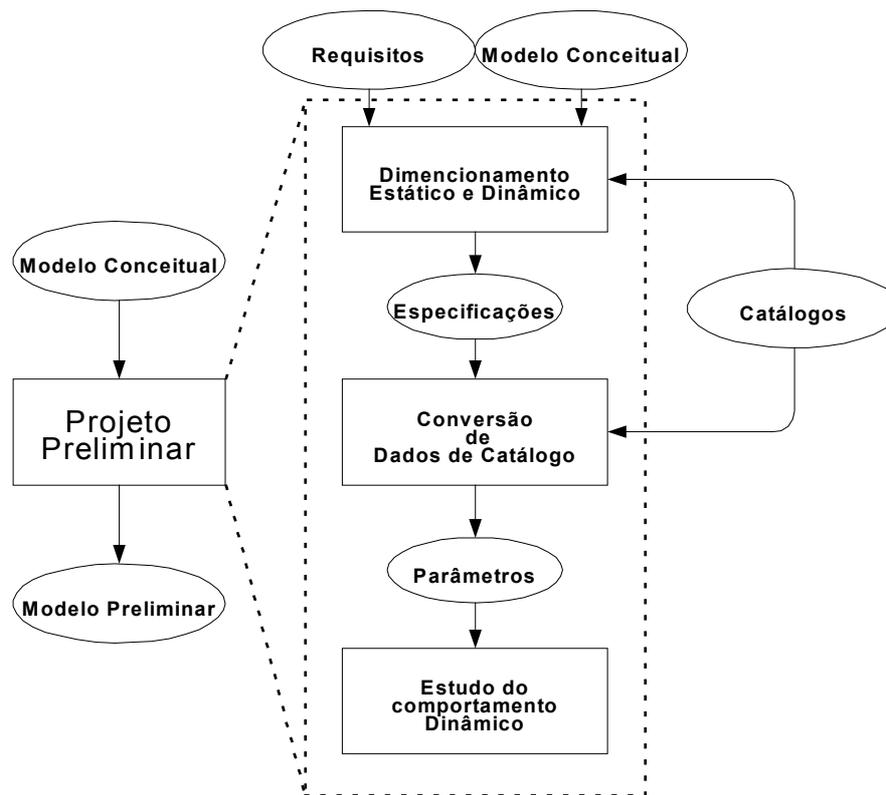


Figura 2.13 - Etapas do projeto preliminar de sistemas hidráulicos de controle de posição (FURST e DE NEGRI, 2002).

2.4 Projeto detalhado

No projeto detalhado, o layout definitivo do produto ou sistema técnico é definido. Seu desenvolvimento deve chegar ao ponto onde aspectos como função, durabilidade, produção, montagem, operação e custos, possam ser testados. Os procedimentos para a produção são definidos e as atividades relativas à finalização do projeto são executadas, como a elaboração de manuais, por exemplo.

Nesta fase a equipe de projeto também faz um registro de todas as lições aprendidas durante o processo, que servirá de base de conhecimento para projetos futuros, auxiliando nas soluções ou evitando a ocorrência de erros conhecidos.

2.5 Comentários

A metodologia de projeto consensual é bastante prática para as atividades de projeto de sistemas técnicos e produtos, sendo que o conhecimento envolvido no presente trabalho é fortemente baseado nesta metodologia, como será mostrado mais adiante. Entretanto, os estudos desenvolvidos no decorrer desta pesquisa concentram-se principalmente na fase de projeto conceitual, pressupondo a fase de projeto informacional já realizada.

Toda via, as atividades de implementação de controle em CLP que serão vistas confundem-se entre as fases de projeto preliminar e detalhado, mesmo que de certo modo não tenham sido o foco principal do trabalho, embora tenham sido realizadas.

CAPÍTULO 3

MODELAGEM E CONTROLE DE SISTEMAS AUTOMÁTICOS

A modelagem de sistemas, em especial os sistemas mecatrônicos e software orientados a objetos, é realizada basicamente sobre três perspectivas denominadas de estrutural, funcional e comportamental. Para compreender estas perspectivas DE NEGRI (2004) as define como:

- **Perspectiva funcional** – estabelece de forma inequívoca a função de cada componente no sistema e qual é a inter-relação entre estas funções. Para esta definição, entende-se como função de um sistema o efeito que ele provoca sobre o ambiente externo (CHANDRASEKARAN e JOSEPHSON, 2000);
- **Perspectiva estrutural** – representa o conjunto de elementos em um sistema e as relações que os conectam uns aos outros, que podem indicar conexões físicas, de comunicação ou relações hierárquicas. A estrutura mostra que o sistema é formado por uma rede de elementos com certo arranjo interno, ordem, organização, constituição, construção etc;
- **Perspectiva comportamental** – informa quando e como as funções de um sistema serão executadas. As características de modelos comportamentais dependem do tipo de sinal de entrada e saída envolvido na execução da função do sistema, sendo divididos em: modelos a estado contínuo e a estado discreto.

Os modelos podem ser classificados ainda quanto à representação em icônicos (semelhante ao sistema real), analógico (analogias), matemáticos, diagramáticos e em linguagem natural. As diferentes formas de modelar são adequadas muitas vezes a áreas de atuação restritas e outras vezes a áreas mais gerais no âmbito da engenharia. Alguns exemplos destes modelos são apresentados na Tabela 2.

Tabela 2 - Síntese de modelos utilizados em engenharia (DE NEGRI, 2004).

Modelo	Perspectiva	Representação	Área técnica
Estrutura de funções	Funcional	Diagramática	Projeto de produto (intertecnológico, mas com ênfase à área mecânica)
Diagrama de blocos	Comportamental	Diagramática + Matemática	Intertecnológico: Elétrica, mecânica, hidráulica, pneumática.
Diagrama de estados	Comportamental	Diagramática	Software, Microeletrônica
Rede de Petri marcada	Comportamental	Diagramática	Sistemas de manufatura
Diagrama de funcionamento - SFC	Comportamental	Diagramática	Automação, pneumática
Grafos de ligação (bond graphs)	Funcional + Comportamental	Diagramática	Multitecnológico: Elétrica, mecânica, hidráulica, pneumática

Rede Canal/Agência	Estrutural + Funcional	Diagramática	Multitecnológico: Elétrica, mecânica, hidráulica, pneumática, software
DFD (Diagrama de fluxo de dados)	Funcional	Diagramática	Software
Diagramas de circuitos elétricos	Funcional	Diagramática	Elétrica
Diagramas E/R	Estrutural	Diagramática	Software
Diagramas de Classe	Estrutural	Diagramática	Software
Desenho mecânico	Estrutural	Ícônica	Mecânica
Função de transferência	Comportamental	Matemática	Intertecnológico
Equações dinâmicas (variáveis de estado)	Comportamental	Matemática	Intertecnológico
Diagrama de circuitos hidráulicos e pneumáticos	Funcional	Diagramática	Hidráulica e pneumática
Maquetes	Estrutural	Ícônica	Mecânica

Para o caso da modelagem de sistemas automáticos, tem-se exemplos específicos de modelos de comportamento para os sistemas a estados contínuos e a estados discretos, além de modelos estruturais e funcionais. Como modelos comportamentais pode-se citar como exemplo a Rede de Petri (CARDOSO e VALETTE, 1997), os autômatos (CURY, 2001, 2003), o MFG (Mark Flow Graph) (MIYAGI, 1996), e funcional e estrutural a rede Canal/Agência (DE NEGRI, 1996) (HEUSER, 1990), PFS (Production Flow Schema) (MIYAGI, 1996). Em virtude do escopo do presente trabalho, serão então apresentados os modelos adequados aos sistemas guiados a eventos discretos, através da teoria de linguagens e autômatos, e o voltado ao projeto de sistemas automáticos, através da Rede Canal/Agência, conforme se mencionou anteriormente.

3.1 Rede Canal/Agência e o modelo estrutural e funcional de sistemas automáticos

A rede Canal/Agência (C/A), introduzida e apresentada por vários autores¹, é um modelo estrutural e funcional cuja representação diagramática apresenta-se através de dois elementos gráficos básicos: um retângulo que representa as entidades ativas do sistema modelado, e um círculo representando as entidades passivas. Os canais são interligados às agências através de conexões que representam os fluxos de energia, matéria e informação, ou uma combinação de energia e matéria, sendo que estas conexões têm uma conotação de portas físicas de entrada e saída das agências.

As agências são então os elementos que realizam atividades sobre os canais ligados às suas portas de entrada e saída, sendo que suas atividades podem ou não depender do

¹ REISIG (1985), REISIG (1992) apud FREY e LITZ (1998, 2000), HOUSER (1990), DE NEGRI (1996) e SANTOS (2003).

conteúdo dos canais de entrada, bem como os canais ligados às portas de saída podem ou não ser afetados pela agência (HOUSER, 1990). Na Figura 3.1, é apresentada a estrutura gráfica da rede C/A.

Elementos básicos			
Símbolo	Nome genérico	Perspectiva funcional	Perspectiva estrutural
	Unidade ativa	Atividade(função)	Agência
	Unidade passiva	Recurso	Canal

Interconexão dos elementos	
Símbolo	Tipo de recurso
	Fluxo de Informação
	Fluxo de Energia
	Fluxo de Matéria
	Fluxo de Energia e Matéria

Figura 3.1 - Rede Canal/Agência (Rede C/A) (DE NEGRI, 1996).

A rede C/A, seguindo algumas regras, pode ser refinada ou condensada de modo a se poder identificar sub funções e/ou sub estruturas, facilitando o entendimento do sistema modelado, a identificação de componentes, estruturas e locais por onde os recursos fluem (Figura 3.2).

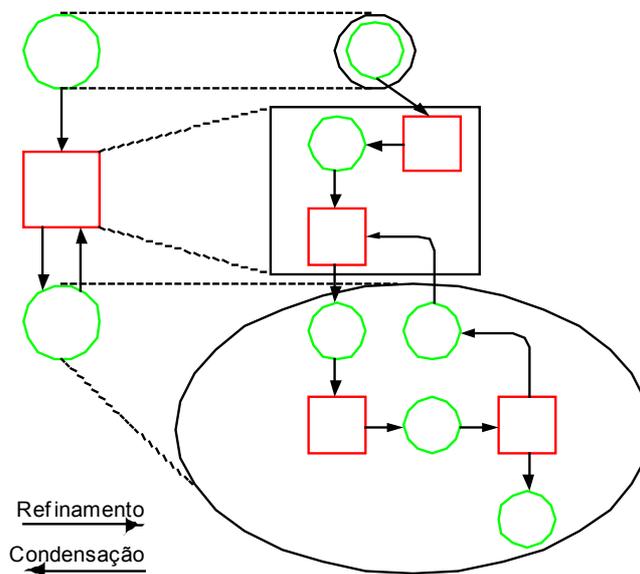


Figura 3.2 - Mecanismo de refinamento e condensação de redes C/A (DE NEGRI, 1996).

Um exemplo de modelagem em rede C/A pode ser visto na Figura 3.3 que ilustra o modelo de um grupo turbina-gerador hidroelétrico geral de maneira condensada, ou seja, sem muitos detalhes.

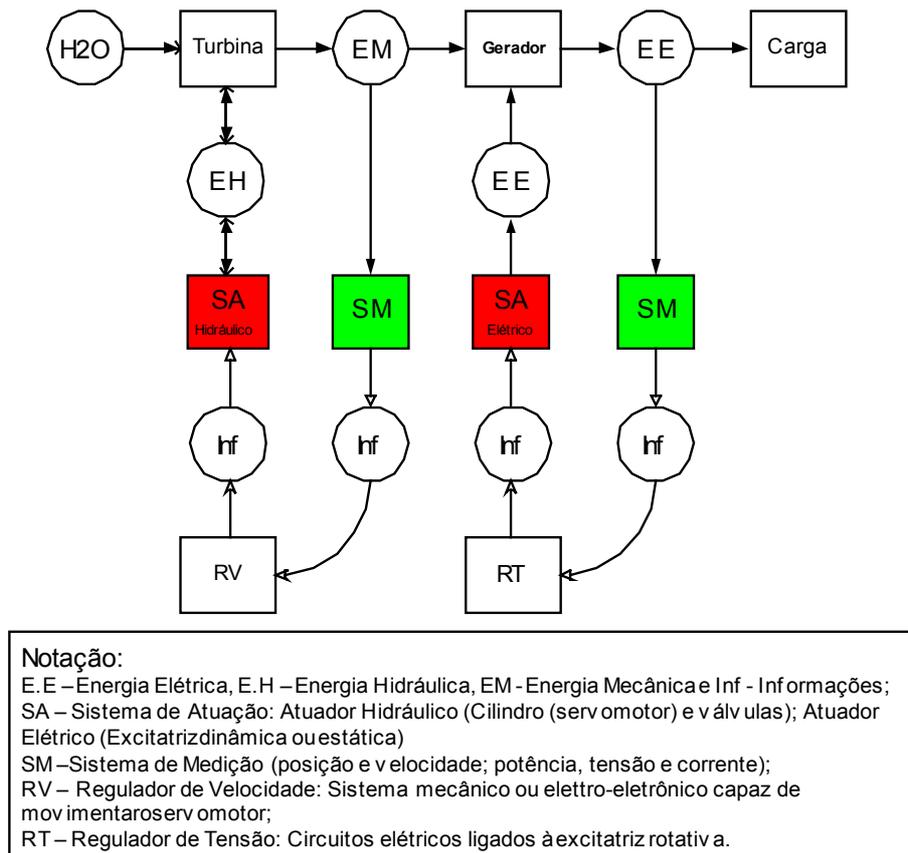


Figura 3.3 - Rede C/A de um grupo Turbina-Gerador genérico (adaptado de PAES e DE NEGRI, 2002).

A representação em rede C/A pode ser utilizada para a atividade de estabelecimento de estrutura funcional da fase de projeto conceitual, como a estrutura de funções proposta por PAHL e BEITZ (1984) apud FERREIRA (1997), mostrada na Figura 2.11. Entretanto, deve-se notar que na representação de PAHL e BEITZ as entradas e saídas têm uma conotação de causa e efeito, o que no caso da rede C/A têm significado de portas físicas, quer dizer, locais por onde energia, matéria e informação fluem.

Utilizando esta ferramenta de modelagem, DE NEGRI (1996) propõe um modelo de representação para os sistemas automáticos em geral, identificando a existência de uma entidade que trata do processamento de matéria e energia e outra que trata do processamento de informações. A comunicação destas entidades, denominadas sistema energético/material e sistema de informação, respectivamente, é promovida por canais de fluxo de informações que são em verdade compostos de sistemas de atuação (SA) e sistemas de medição (SM). Este modelo é apresentado na Figura 3.4, onde se pode observar uma forma condensada e outra um pouco mais refinada.

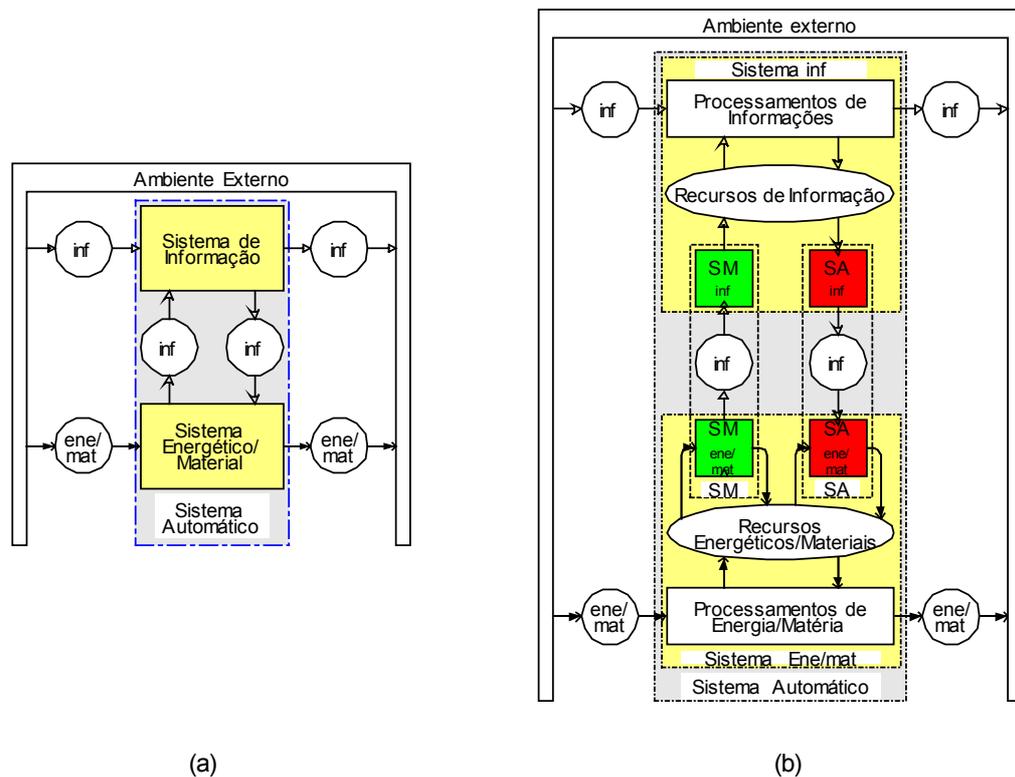


Figura 3.4 - Modelo funcional e estrutural, em rede C/A , de um sistema automático - (a) Condensado, (b) Refinamento em primeiro nível (DE NEGRI, 1996).

Este modelo é então fracionado durante as atividades de projeto, permitindo que o problema seja trabalhado em partes menores de sub estruturas e sub funções, promovendo resultados mais aprimorados, sem, no entanto, perder-se a visão do sistema como um todo. DE NEGRI ressalta ainda que os sistemas de atuação e medição podem se fundir em alguns casos e nesta ocorrência o autor denomina-os de sistema de atuação-medição (SAM). Para os aspectos comportamentais são utilizados modelos adequados ao tipo de sistema como funções de transferência ou diagramas de estado, sendo estes modelos sempre vinculados às agências da representação estrutural e funcional.

As informações trocadas com o ambiente externo pelo sistema de processamento de informações são geralmente embutidas no modelo comportamental do sistema energético/material de maneira direta, exemplo do caso de representações em SFC (Sequential Function Charts) (IEC, 1988), ou indireta, como no caso de modelagem por diagramas de estado em que a comunicação é adicionada no detalhamento operacional do sistema, como será mostrado mais adiante. A entrada e saída de energia e matéria por ser caracterizada pelo conhecimento dos sistemas anterior e posterior ou suposta para efeito de simplificação de projeto, como por exemplo, pode-se supor que uma peça está sempre disponível na entrada e que sempre existe espaço para descarregar uma peça na saída.

3.2 Teoria de linguagem e autômatos e o modelo comportamental de sistemas automáticos

Os sistemas automáticos têm seu comportamento dinâmico representado através de modelos de acordo com a classe de seu comportamento, que pode ser guiada ou não pelo tempo e com sinais de entrada e saída discretos ou contínuos. Para os casos contínuos, geralmente guiados pelo tempo, o comportamento é representado através de equações diferenciais ou a diferenças, equações de estado etc, mas para sistemas guiados por eventos discretos (SED) estes paradigmas não são adequados.

Os SEDs, classe de sistemas abordada neste trabalho, são sistemas onde a simples passagem do tempo não é suficiente para garantir que haja evolução de seu estado, sendo necessária a ocorrência de eventos, internos ou externos ao sistema, para que ocorra uma mudança de seu estado de forma instantânea. A definição formal de SED é então:

Definição 1 *Sistema a eventos discretos é um sistema dinâmico que evolui de acordo com a ocorrência abrupta de eventos físicos, em intervalos de tempo em geral irregulares e desconhecidos.*

Os eventos são estímulos referentes às ocorrências no mundo externo e interno, sendo por definição considerados instantâneos. Como eventos pode-se considerar o início ou término de uma tarefa, a chegada de uma pessoa ou objeto a uma fila, o clique de um mouse de computador, a chegada de um e-mail, etc.

Embora existam vários modelos para SEDs, tais como as redes de Petri, Cadeias de Markov, Teoria de filas, Álgebra Max-Plus, Teoria de linguagens e Autômatos e etc, nenhuma foi estabelecida como paradigma para a representação de um SED. Dentre os modelos citados destaca-se a teoria de linguagens e autômatos por ser dotada de um procedimento automático de síntese de controladores.

Os temas a seguir estão apresentados de maneira simples e resumida, evitando-se ao máximo os formalismos matemáticos próprios a teoria de linguagens e autômatos e a teoria de controle supervisorio. Para uma melhor compreensão do assunto e o estudo através dos equacionamentos adequados, sugere-se a leitura das obras referenciadas¹ sobre o assunto ou como leitura complementar o artigo de QUEIROZ e CURY (2000) - **Controle Supervisorio Modular de Sistemas de Manufatura Discretos** – disponível no ANEXO 1.

3.2.1 Linguagens

Da mesma maneira que nos idiomas, em que símbolos (letras) são agrupados em cadeias (formando palavras) próprias a cada idioma, as linguagens dos modelos de SEDs

¹ CASSANDRAS e LAFORTUNE (1999), CURY (2001, 2003), CURY e BITTENCOURT (2003), MIYAGI (1996), QUEIROZ (2004), QUEIROZ e CURY (2000, 2002), QUEIROZ et al (2001)

são conjuntos de cadeias de determinados símbolos relativos à seqüência de eventos reconhecida por um sistema.

Definição 2 *Uma linguagem L , definida sobre um alfabeto Σ , é um conjunto de cadeias formadas por símbolos pertencentes a Σ .*

Assim, o comportamento de um SED pode ser descrito por um par de linguagens, onde o alfabeto Σ é o conjunto de todos os eventos que afetam o sistema, através de uma dupla $D=(L, L_m)$. Neste modelo D , a linguagem L descreve o comportamento gerado pelo sistema, ou seja, é o conjunto de todas as cadeias de eventos possíveis de ocorrerem no sistema e a linguagem L_m descreve o comportamento marcado que é o conjunto de cadeias de L correspondentes às tarefas completas que o sistema pode realizar. Exemplo:

$$\Sigma = \{\alpha, \beta, \gamma\} \quad L = \{\varepsilon, \alpha, \alpha\beta, \alpha\gamma, \beta\gamma, \alpha\beta\gamma, \dots\} \quad L_m = \{\varepsilon, \alpha\beta\gamma, \alpha\beta\gamma\alpha\beta\gamma, \dots\}$$

onde ε é a cadeia vazia.

3.2.2 Autômatos

Embora existam autômatos de estados infinitos e/ou não determinísticos, apresentar-se-á a seguir a definição de autômatos determinísticos de estados finitos que são os modelos que representam os sistemas estudados neste trabalho. Um autômato determinístico de estados finitos é uma quintupla $G = (X, \Sigma, f, x_0, X_m)$, onde:

- X é o conjunto finito de estados;
- Σ é o conjunto de eventos (alfabeto);
- $f : X \times \Sigma \rightarrow X$ é a função de transição que leva de um estado a outro qualquer pertencente a X ;
- x_0 é o estado inicial do autômato;
- X_m é o conjunto de estados marcados ou finais, $X_m \subseteq X$.

Um autômato pode ser representado através de um diagrama de transição de estados que são grafos dirigidos onde os nós representam os estados e os ramos representam as transições entre os estados, ou seja, os eventos. Neste grafo, o estado inicial é identificado através de uma seta e os estados finais com círculos duplos concêntricos.

A Figura 3.5 ilustra o grafo para um autômato G de uma máquina de dois estados (parado, operando), onde α é um evento controlável sendo seu ramo interceptado para representar este fato. Os eventos controláveis são aqueles em que um controlador pode impedir sua ocorrência, o que não acontece no caso de eventos não controláveis como o evento β no exemplo. O autômato G poderia representar o comportamento de uma máquina ferramenta, por exemplo, em que o evento α seria o comando para o início de operação e o evento β seria o sinal de que a usinagem terminou.

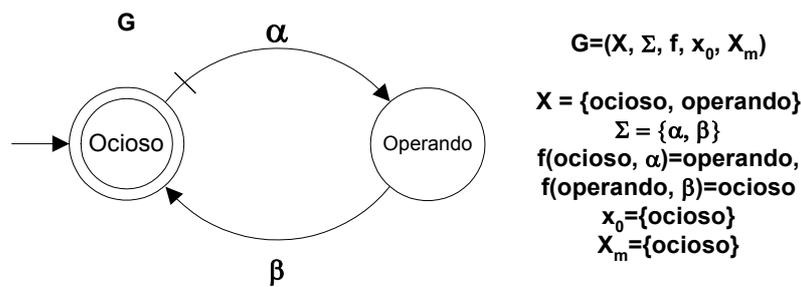


Figura 3.5 - Exemplo de autômato para uma máquina qualquer.

Pode-se notar que o autômato G é uma abstração da máquina, não representando o detalhamento da operação de usinagem que envolve vários comandos e sinais de sensores, temporizadores etc, que podem ser representados através de outros modelos, como será apresentado mais adiante. Em geral, um SED pode ser dividido em vários sub sistemas que por sua vez podem ser representados por autômatos e, assim, para que se obtenha a representação global do sistema, estes autômatos têm que ser compostos de modo a se obter um modelo geral da planta do sistema.

A composição de autômatos é denominada produto síncrono, representada por duas barras paralelas (//), sendo que no autômato resultante um evento comum aos modelos originais só pode ocorrer de maneira síncrona, ou seja, o evento somente ocorrerá se for possível que ele ocorra nos dois modelos originais.

3.3 Teoria de controle supervisorio modular local

Com a modelagem de SED através de da teoria de linguagens e autômatos, o número de estados dos modelos cresce exponencialmente à medida que se agregam novos sub-sistemas ou especificações e este fato impõe fortes limitações computacionais para a modelagem, cálculo de controladores e implementação de sistemas a eventos discretos.

Como forma de reduzir o esforço computacional de síntese e implementação do controle, como o caso de memória e velocidade de processamento de CLPs, foi proposta por Queiroz e Cury (2000) a abordagem de controle modular local que será apresentada de forma resumida. Para explicar o conceito de controle modular local são necessárias as definições dos conceitos de controle supervisorio monolítico e modular, sistema produto e planta local.

3.3.1 Controle supervisorio monolítico e modular

No controle supervisorio monolítico, um supervisor único representado por um autômato S tem a função de desabilitar eventos controláveis da planta supervisionada em função do estado em que a planta se encontra e em concordância com a lei de controle estabelecida (Figura 3.6).

Estas leis de controle são obtidas através de especificações modeladas também em autômatos, que restringem o comportamento da planta livre com a determinação de quais cadeias de eventos são permitidas no funcionamento controlado do sistema.

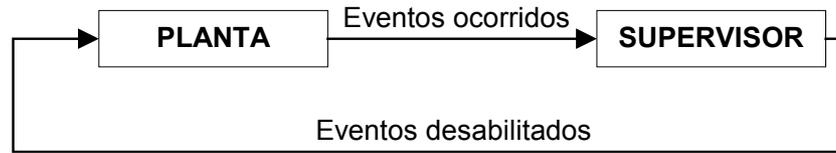


Figura 3.6 - Esquema de controle monolítico (QUEIROZ e CURY, 2000).

Para os casos em que a especificação de controle é uma composição de especificações menores pode-se adotar o controle modular que consiste de vários supervisores modulares, cada um atendendo a uma especificação, que trabalhando em conjunto representam o mesmo controle realizado por um supervisor monolítico. Esta abordagem pode ser vista na Figura 3.7.



Figura 3.7 - Esquema de controle modular (QUEIROZ e CURY, 2000).

Como os supervisores são menores, sua obtenção e futuras modificações são em geral mais fáceis de serem realizadas. No entanto pode existir um problema de conflito entre os supervisores modulares, já que cada um deles tem uma visão parcial do sistema, conflito este que pode provocar o bloqueio mortal do sistema ou impedi-lo de completar uma tarefa, por exemplo.

3.3.2 Representação por sistema produto (RSP)

Na fase de modelagem de um sistema pode-se dividi-lo em subsistemas de maneira que eles não contenham eventos em comum e estes subsistemas são denominados subsistemas assíncronos. Este tipo de modelagem denomina-se representação por sistema produto (RSP).

Para obter-se um sistema produto deve-se combinar (produto síncrono) todos os subsistemas que contenham eventos em comum de modo a obter subsistemas completamente independentes em relação aos eventos que os compõem.

A Figura 3.8 mostra a relação de intercessão entre os alfabetos Σ_1 a Σ_6 dos sub sistemas M1 a M6, em um exemplo para um sistema composto de seis sub sistemas. Neste exemplo, vemos o procedimento para a obtenção do sistema produto em que subsistemas com eventos em comum (M1, M2) e (M3,M4, M5) são combinados (Md1=M1//M2, Md2=

M3//M4//M5, Md3=M6) para obter subsistemas assíncronos, ou seja, sem eventos em comum, denominados módulos do sistema produto. Neste exemplo, os alfabetos do módulos são respectivamente $\Sigma_7 = \Sigma_1 \cup \Sigma_2$, $\Sigma_8 = \Sigma_3 \cup \Sigma_4 \cup \Sigma_5$ e $\Sigma_9 = \Sigma_6$.

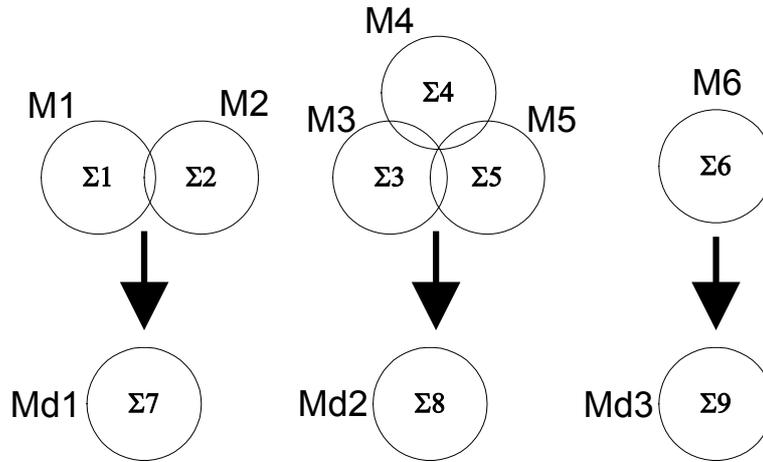


Figura 3.8 - Relação dos conjuntos dos alfabetos (eventos) de um sistema composto de sub sistemas M1 á M6 e os módulos do sistema produto Md1, Md2 e Md3.

3.3.3 Planta local

Após a representação do sistema global por sistema produto é necessário ainda combinar os módulos resultantes de acordo com as especificações existentes para o controle modular da planta, de maneira que cada supervisor modular esteja relacionado a uma sub-planta que é a composição de todos os subsistemas assíncronos a qual a especificação se refere.

Os resultados destas composições são denominados plantas locais como pode ser visto no exemplo da Figura 3.9. As plantas locais são obtidas através do produto síncrono dos módulos do sistema produto que estão relacionados através de uma especificação, ou seja, se uma especificação E_b esta relacionada a uma sub planta M2, pertencente a um módulo do sistema produto Md1, e a outra sub planta M3, pertencente a outro módulo do sistema produto Md2, a planta local M1b, relacionada a esta especificação, é obtida através do produto síncrono Md1//Md2.

Assim tem-se, a partir da especificação E_a relacionada a M1 a planta local M1a, de E_b resulta a planta local M1b e de E_c a planta local M1c, como pode ser observado na Figura 3.9, sendo os alfabetos das plantas locais respectivamente Σ_7 , $\Sigma_{78} = \Sigma_7 \cup \Sigma_8$ e Σ_9 .

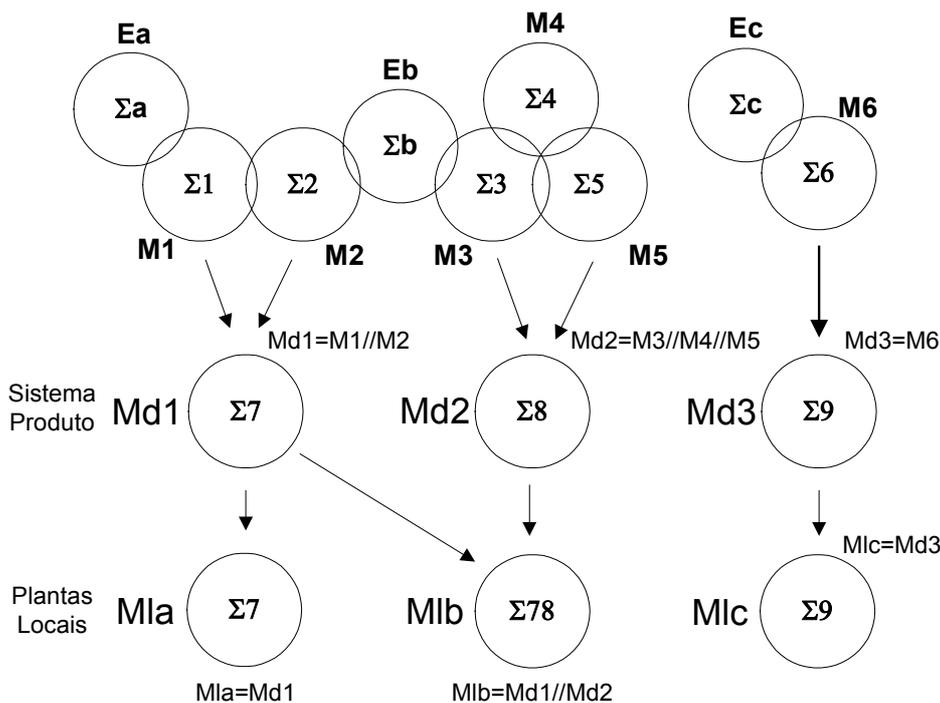


Figura 3.9 – Geração das plantas locais, composição dos módulos do sistema produto de acordo com as especificações (Ea , Eb e Ec).

3.3.4 Supervisor modular local

O controle modular local é implementado através dos supervisores modulares locais sobre as respectivas plantas locais como ilustrado na Figura 3.10.

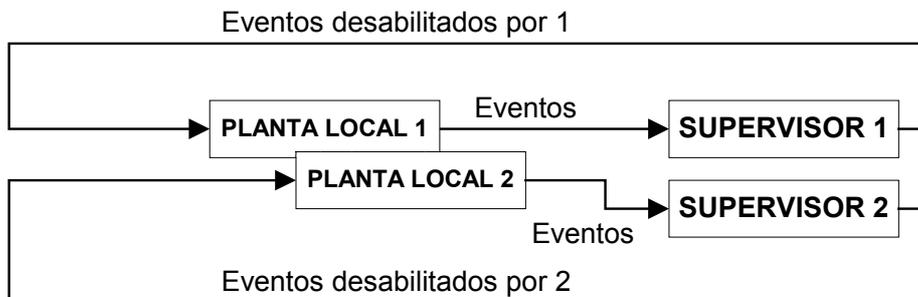


Figura 3.10 - Esquema de controle modular local (QUEIROZ e CURY, 2000).

A síntese do supervisor modular local será então realizada como apresenta a teoria de controle supervisorório, encontrando supervisores não-bloqueantes e minimamente restritivos, utilizando-se para isto cada especificação com sua planta local correspondente.

Conforme foi visto no item 3.3.1, pode existir conflito entre supervisores modulares, restringindo menos do que o necessário o sistema ou até provocando o seu bloqueio. A esta interferência que uma especificação pode ou não exercer na outra se dá o nome de modularidade, ou seja, se as especificações de dois supervisores modulares não fizerem com que estes supervisores provoquem o bloqueio do sistema ou não impeçam que o sistema complete suas tarefas, pode-se dizer que elas são modulares.

A modularidade pode ser verificada através de um teste que consiste em realizar o produto síncrono de todos os supervisores modulares locais, verificando se o resultado final é aparado (denominado TRIM), ou seja, se o autômato resultante do produto síncrono dos supervisores modulares locais é não-bloqueante.

Caso a modularidade não se verifique este aspecto deve ser resolvido de maneira que não haja supervisores conflitantes que impeçam a implementação do controle de maneira modular. Segundo QUEIROZ e CURY (2000), existem várias técnicas elaboradas para a resolução de conflitos, como por exemplo, a aplicação de controle hierárquico, mas uma técnica que talvez seja a mais evidente é a aplicação do controle monolítico nas especificações conflitantes, ou seja, realizar o produto síncrono do mínimo possível de especificações conflitantes até que não exista mais conflito.

Com objetivo de implementar o controle obtido em um dispositivo físico como um CLP, a redução dos supervisores obtidos deve ser levada em consideração, já que a memória dos CLPs é limitada e sua utilização está diretamente relacionada ao número de estados que o supervisor possui. Neste sentido, QUEIROZ e CURY também citam o algoritmo de minimização de VAZ e WONHAM (1986) e SU e WONHAM (2004), que resolve este problema.

A já citada explosão exponencial de estados do controle monolítico, problemática para a limitação de memória dos dispositivos físicos, tem seu efeito bastante reduzido com a utilização do controle modular local, tornando possível a implementação de sistemas mais complexos.

3.4 Estrutura de controle e implementação

Embora existam outras formas de estruturar o controle para a sua implementação, será apresentada a proposta feita por QUEIROZ et al (2001), para a implementação em CLP utilizando a linguagem em diagrama escada (*Ladder Diagram*), que serviu de base para algumas propostas deste trabalho. A estrutura de controle divide-se então basicamente em três partes: os supervisores modulares, o sistema produto e as seqüências operacionais, cujo conceito será apresentado mais adiante.

Esta estrutura é mostrada na Figura 3.11 sendo que, os estados dos supervisores e módulos do sistema produto são representados através de variáveis do CLP e as mudanças ocorrem através de relações lógicas entre os estados e os eventos ocorridos. Para tanto, os supervisores desabilitam os eventos controláveis no sistema produto, em concordância com as leis de controle, observando os eventos ocorridos nos módulos do sistema produto.

Desta forma, uma estrutura que também deve ser implementada no CLP é a de desabilitação de eventos controláveis, sinalizando que o evento que ocorrerá no sistema produto é o que não está desabilitado pelos supervisores locais e é possível de ocorrer no estado atual. Pela lei de controle este evento deve ser único, ou seja, para o caso dos eventos

controláveis a ocorrência é sempre determinística em malha fechada, o que não acontece no caso de eventos não controláveis.

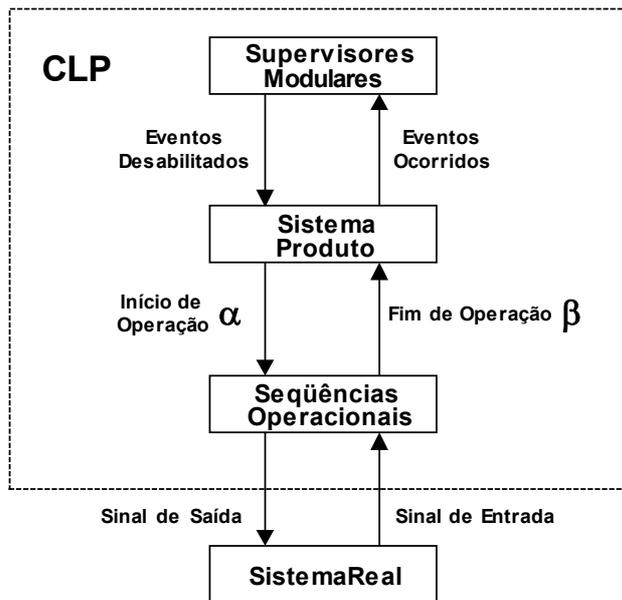


Figura 3.11 - Estrutura básica de um sistema de controle (QUEIROZ et al, 2001).

A estrutura de desabilitação é necessária porque a teoria de controle supervisorio assume que a evolução dos sistemas reais é espontânea, o que não é observado na prática. Por exemplo, se uma máquina está desligada e pode ser ligada, ela não irá ligar sozinha, deve haver um comando externo à máquina que mude seu estado de desligada para ligada como uma chave manual ou o comando de um CLP.

Os módulos do sistema produto são abstrações dos sub sistemas que compõem a planta e, desta forma, não contemplam os detalhamentos do funcionamento destes sub-sistemas. Um exemplo clássico desta forma de modelagem é o de uma furadeira automática modelada por um autômato de dois estados {parada, operando}, que na verdade tem a sua operação composta por uma seqüência de ações tais como prender e soltar a peça, ligar e desligar a broca, baixar e levantar a broca (furação), além da detecção da peça, do furo, de broca quebrada, etc. Para fazer a interface entre a abstração do sistema nos modelos do sistema produto e as minúcias de operação nos sub sistemas reais são implementadas as estruturas de seqüências operacionais (SO).

As SO são geralmente projetadas utilizando-se a modelagem por SFC (Sequential Function Charts), também conhecido como Grafcet, padronizado através da norma IEC 848 (1988). As SO recebem um comando de início de operação do sistema produto (geralmente representado por eventos com prefixo α), executam a seqüências de operações, acionando atuadores e recebendo respostas de sensores na planta, e ao fim das operações a SO informa ao sistema produto que a seqüência terminou (geralmente representado por eventos com prefixo β).

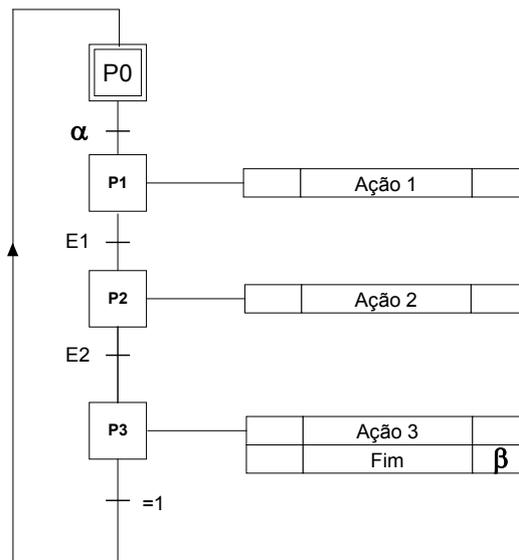


Figura 3.12 - Sequência operacional modelada através de um SFC.

No exemplo da Figura 3.12, uma SO de 4 passos realiza três ações cujas condições são o comando de início α , e sinais E1 e E2, que podem ser sensores, terminando com o sinal β de fim de operação.

A interpretação de modelos de SEDs em programas de CLP como os diagramas escada ou diagramas de blocos é apresentada por QUEIROZ et al (2001) e VIEIRA (2003), para o caso de diagrama escada e modelos realizados em autômatos, e por BOLLMANN (1997), para os casos de diagrama escada e de diagrama de blocos para modelos realizados em SFC (Grafcet). Pode-se perceber que existem poucas diferenças entre os dois casos e talvez a maior delas seja a possibilidade de paralelismo no caso do SFC, o que é proibido para o caso dos autômatos, isto porque os dois diagramas representam os estados de maneira diferente.

Basicamente a lógica implementada em um programa em CLP é: “Se o sistema está em um estado x_n (uma variável x_n interna do CLP esta em nível lógico alto =1) e uma certa condição Y está satisfeita, então sair do estado x_n e passar ao estado x_{n+1} (fazer x_{n+1} =1 e $x_n=0$)”.

A Figura 3.13 mostra como interpretar um autômato em diagrama escada para o caso da ligação de estados 1 a 1, o caso de dois ou mais estados levarem a um estado pode ser visto na Figura 3.14.

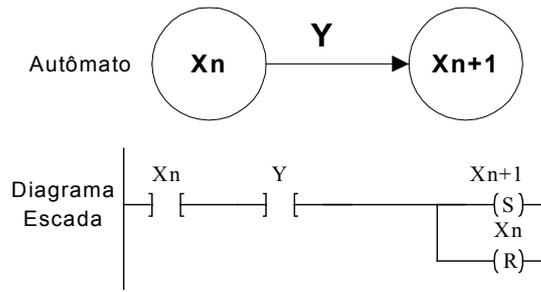


Figura 3.13 - Estrutura de interpretação de um Autômato em Diagrama Escada.

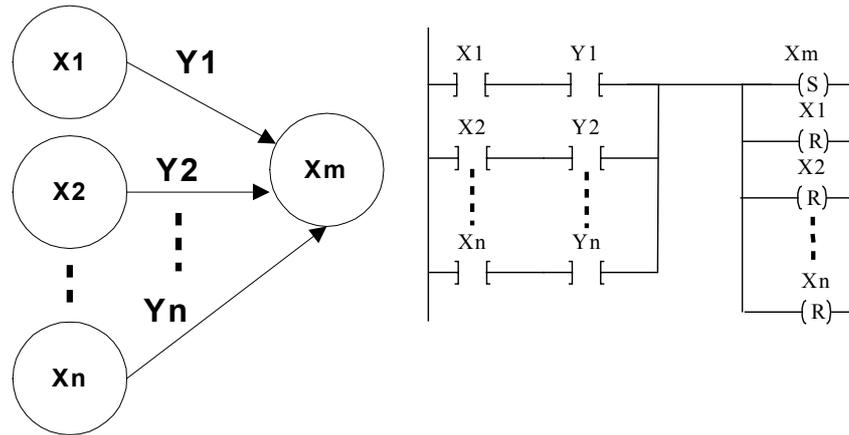


Figura 3.14 - Interpretação de dois ou mais estados que levam a um estado.

Como pode ser visto é razoavelmente simples a implementação dos modelos em diagrama escada, mas dois cuidados devem ser tomados para garantir o perfeito funcionamento do controle. O primeiro, já mencionado, é a implementação de uma estrutura de desabilitação de eventos controláveis relacionada com o modelo. O modelo na verdade será implementado em função de uma condição satisfeita para os eventos não controláveis e de uma condição não negada para os eventos controláveis. A Figura 3.15 mostra a estrutura em diagrama escada de como é realizado o sistema produto.

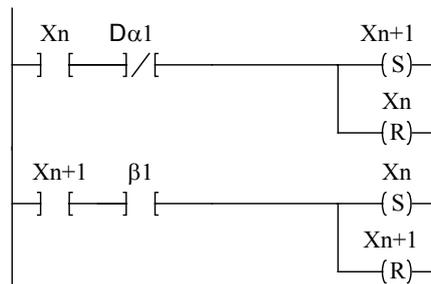


Figura 3.15 - Representação do sistema produto onde $\alpha 1$ é controlável e $\beta 1$ é não controlável.

A estrutura de desabilitação é uma lista de eventos desabilitados $D\alpha i$ ($i=1,2,3...$), onde αi são os eventos controláveis, criados pelos supervisores de maneira que um estado x_n ativo do sistema produto sempre evolui para o estado x_{n+1} se a não estiver desabilitado

($D\alpha=0$), como ilustrado na Figura 3.15. Para o caso dos eventos não controláveis ocorre a evolução da planta sempre que ocorrer o evento.

A implementação da lista de desabilitação é realizada juntamente com a implementação do supervisor, sendo que, como o supervisor é também representado por um autômato, sua implementação deve ficar como ilustrado na Figura 3.13 e na Figura 3.14. A Figura 3.16 mostra os fragmentos do diagrama escada de um supervisor, com a mudança do estado x_{sn} para o estado x_{sn+1} através do evento $\alpha 1$, e da estrutura de desabilitação onde, para este supervisor, estando nos estados x_{sn+1} e x_{sn-2} o evento $\alpha 2$ deve ser desabilitado ($D\alpha 2=1$).

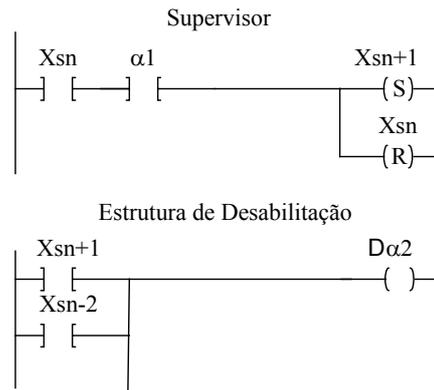


Figura 3.16 - Diagrama escada para o supervisor e estrutura de desabilitação.

O segundo cuidado que deve ser tomado é a garantia de que uma nova evolução do sistema produto não vai ocorrer enquanto o estado do supervisor não for atualizado, já que a teoria de controle supervisório não prevê a ocorrência de dois eventos simultaneamente. As soluções para este problema são descritas por QUEIROZ e VIEIRA, mostrando que um desvio do programa para uma posição adequada, ou a sinalização por uma variável booleana de que o sistema não pode evoluir enquanto não houver condição (atualização do supervisor), resolvem de maneira satisfatória o que seria uma anomalia de controle.

Em verdade, os diagramas escada mostrados acima estão simplificados para a melhor compreensão do assunto, sendo necessária a diferenciação entre os sinais de início e fim de operação e os eventos relacionados a eles e que são tratados pelos supervisores. Assim, quando um evento controlável não está desabilitado e pode ocorrer no estado atual, a linha de código gera o comando de início e marca a variável relativa indicando ao supervisor que o evento ocorreu e acontece de maneira similar o sinal de fim de operação.

A última estrutura a ser implementada é a das seqüências operacionais que, como já foi mencionado, contém as entradas e saídas do CLP. As seqüências operacionais são geralmente representadas através de SFC (Grafcets) por ser a forma mais aceita no projeto de sistemas seqüenciais e por quase todos os grandes fabricantes de CLPs disponibilizarem ambientes de programação destes equipamentos diretamente neste modelo.

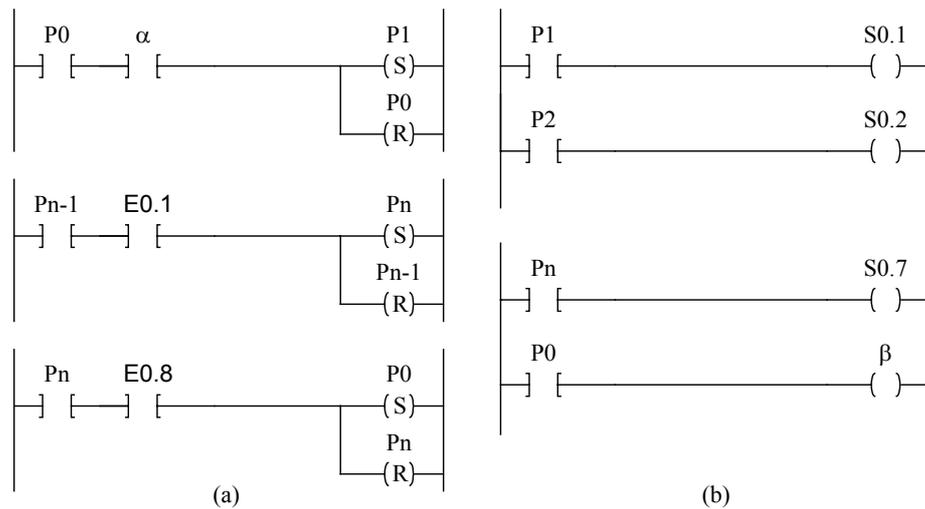


Figura 3.17 - Exemplo de diagrama escada de uma seqüência operacional onde, (a) – Bloco de passos e (b) - Bloco de saídas.

Supondo um SFC qualquer como o visto na Figura 3.12, a estrutura de passos representada é implementada em diagrama escada como visto na Figura 3.17. Desta maneira pode-se ver que os passos de um SFC são realizados em um bloco do diagrama escada e as saídas destes passos são realizadas em um outro bloco. Esta implementação é mais bem descrita por BOLLMANN (1997) que faz uma análise de sistemas seqüenciais e mostra como implementar SFCs em diagrama escada, resolvendo desvios, paralelismo, exclusão mútua, entre outras.

Uma recomendação pertinente é a de que se deve considerar os eventos não-controláveis como sendo prioritários em relação aos controláveis, isto implica alguns cuidados na implementação em CLP devido ao modo de trabalho destes equipamentos, por ciclos de operação. Assim, deve-se implementar as linhas do diagrama escada que contenham os eventos não-controláveis antes das linhas com eventos controláveis, pois o CLP muda as variáveis internas assim que existe condição para isto, não esperando que a execução chegue ao final do código.

3.5 Exemplo de controle supervisorío

Como forma de esclarecimento será apresentado um exemplo simples de representação, síntese de supervisor e implementação em CLP. Tomar-se-á como exemplo um sistema fictício de usinagem composto de duas furadeiras e um armazém intermediário (*Buffer*) de capacidade unitária.

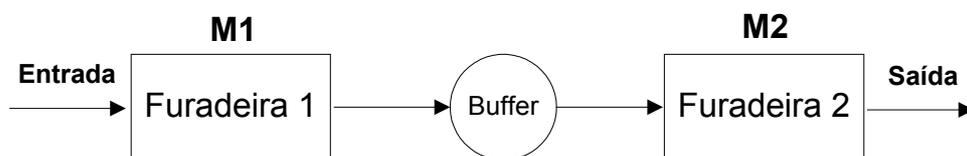


Figura 3.18 - Sistema exemplo.

Como especificação (E) deve-se evitar que a furadeira 1 armazene mais de uma peça no *buffer*. O sistema produto referente a cada furadeira e a especificação que descreve a restrição comportamental ao sistema estão mostrados na Figura 3.19.

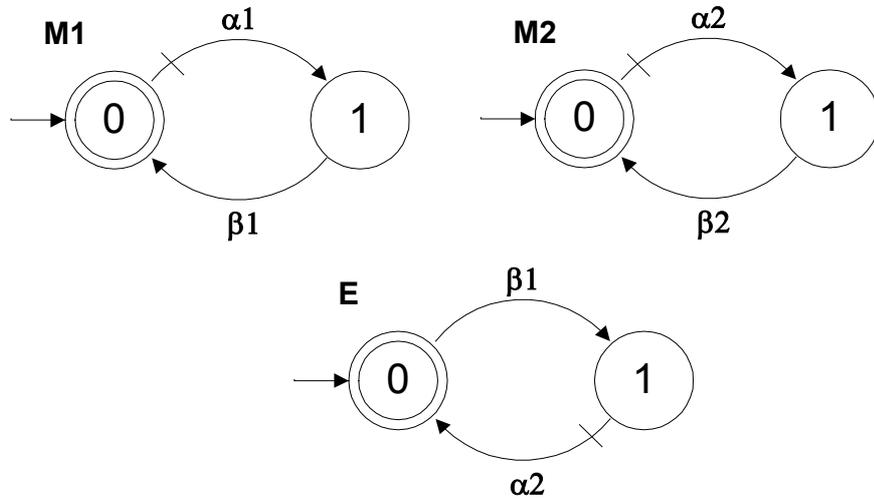


Figura 3.19 - Modelo das furadeiras por sistema produto (M1 e M2) e da especificação (E).

Os eventos controláveis α_1 e α_2 representam o início de operação de M1 e M2, respectivamente, e da mesma maneira os eventos não controláveis β_1 e β_2 representam o final de operação. Como o exemplo é pequeno, com uma única especificação, a modelagem global e síntese monolítica é idêntica a modelagem por sistema produto e síntese modular local. Sendo assim o planta local fica como mostrado na Figura 3.20.

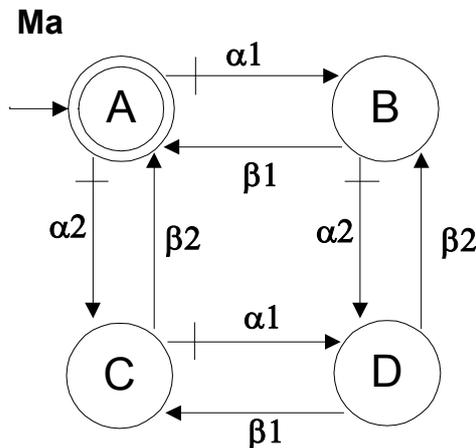


Figura 3.20 - Planta local.

Utilizando os algoritmos para o cálculo do supervisor minimamente restritivo tem-se o supervisor modular local mostrado na Figura 3.21. O diagrama escada do supervisor terá então 6 linhas e, como exemplo para o caso do estado A1, e a estrutura de desabilitação relacionada a ele ficará segundo mostrado na Figura 3.22.

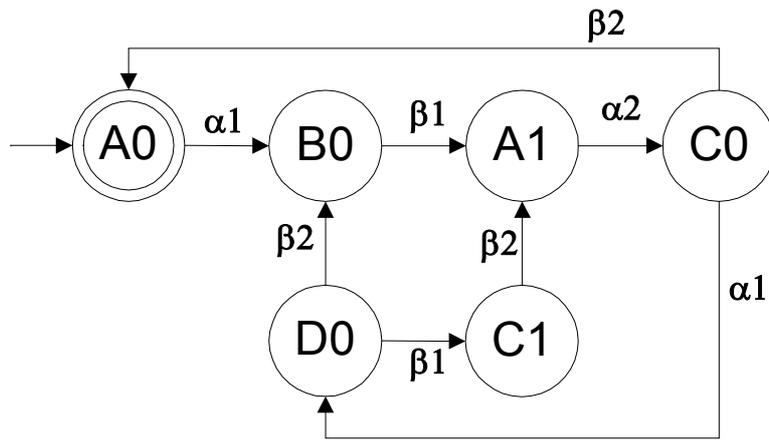


Figura 3.21 - Supervisor modular local.

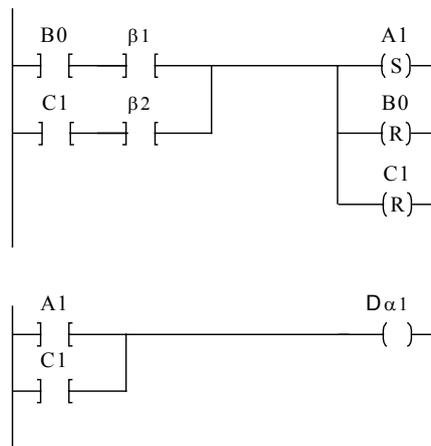


Figura 3.22 - Diagrama escada para o estado A_1 e a desabilitação de α_1 .

O supervisor modular mostrado pode ser reduzido a um supervisor equivalente de 2 estados, para promover sua melhor compreensão e implementação. A título de exemplificação, a forma reduzida do supervisor mostrado na Figura 3.21 pode ser observada na Figura 3.23.

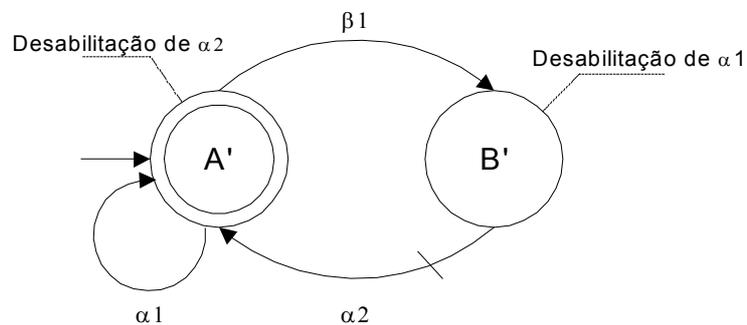


Figura 3.23 - Supervisor reduzido para o exemplo das duas furadeiras.

O sistema produto é implementado conforme a Figura 3.24. Por sua vez, as seqüências operacionais dependem das entradas e saídas do CLP e das operações de cada sub-sistema, ficando similar ao mostrado na Figura 3.25.

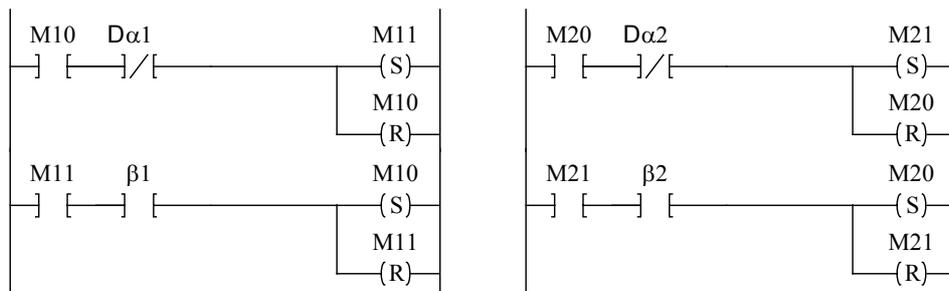


Figura 3.24 - Diagrama escada do sistema produto das furadeiras M1 e M2.

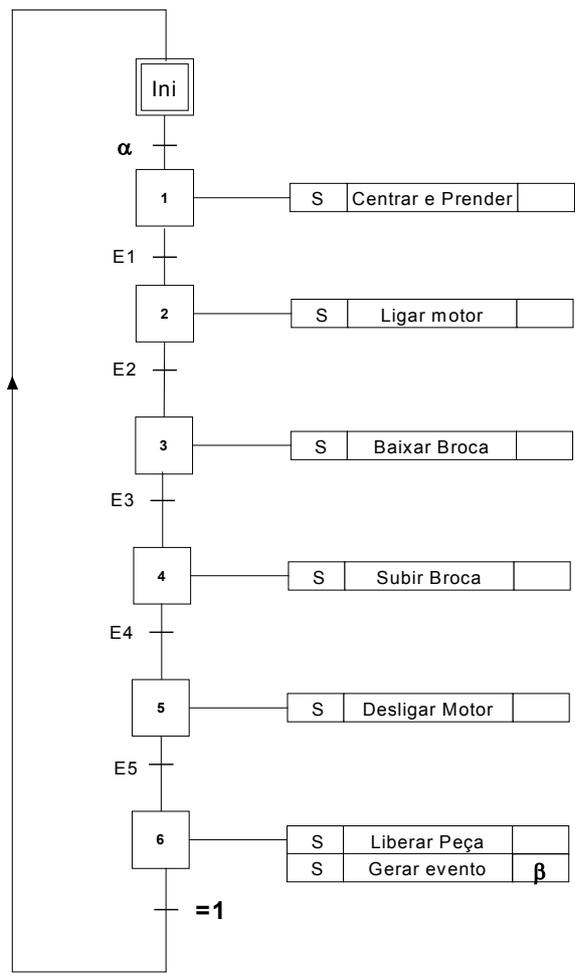


Figura 3.25 - Exemplo de SO para uma furadeira simples, em SFC.

3.6 Projeto conceitual de SMMA

Conforme mencionado, um dos mais recentes resultados rumo à elaboração de uma metodologia de projeto de sistemas automáticos foi a proposta de um modelo procedural para o projeto conceitual de sistemas automáticos de manipulação e montagem, realizada por SANTOS (2003), em sua Tese de doutorado. O projeto conceitual é fundamentalmente composto de atividades de descrição funcional, de tal modo que se pode evoluir dos requisitos de projeto, vindos do projeto informacional, para uma concepção inicial do sistema.

Utilizando os modelos de descrição estrutural, funcional e comportamental descritos no Capítulo 3, SANTOS propõe um procedimento de projeto conceitual para sistemas de manipulação e montagem automatizados (SMMA) que contempla de maneira integrada o projeto físico e o projeto do controle. Este procedimento está descrito na Figura 3.26.

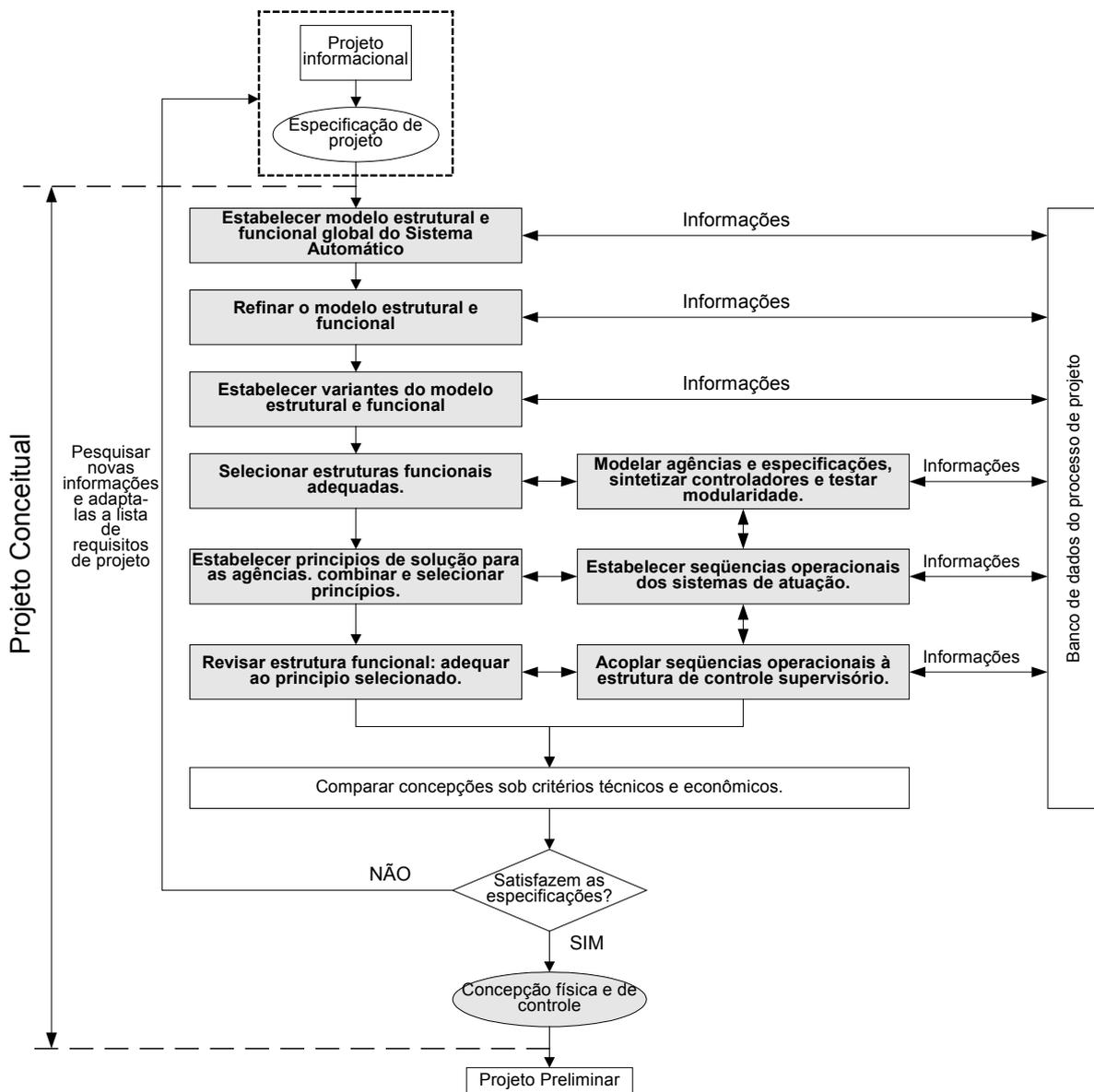


Figura 3.26 - Projeto conceitual para SMMS (adaptado de SANTOS, 2003).

A proposta de SANTOS também inclui um banco de dados relativo ao processo de projeto, contendo um conjunto de soluções na forma de modelos em autômatos do comportamento elementar das agências e um arcabouço de especificações operacionais associadas a SMMA. Este estudo abrangeu os transportadores síncronos e assíncronos, bem como as configurações mistas destes, além da identificação de agências equivalentes em partes distintas da estrutura do sistema modelado.

O modelo em autômato utilizado neste procedimento é o clássico de dois estados, como o exemplo mostrado na Figura 3.27, tornando simples a modelagem comportamental das agências. O arcabouço de especificações abrange uma boa gama de soluções para o

funcionamento adequado do sistema controlado, mesmo em casos de intermitência no fluxo de peças ou para o caso de canais exclusivamente de passagem, ou seja, sem operações sobre o objeto manipulado. Este último caso, de canais sem atividades, é especialmente problemático pois o fluxo de peças através do sistema é monitorado pela cadeia de eventos iniciada por sua entrada no processo, já que geralmente não se utilizam sensores de presença em todos os canais por onde os recursos transitam, e este fato complica bastante o aspecto das especificações relativas às agências adjacentes a este canal.

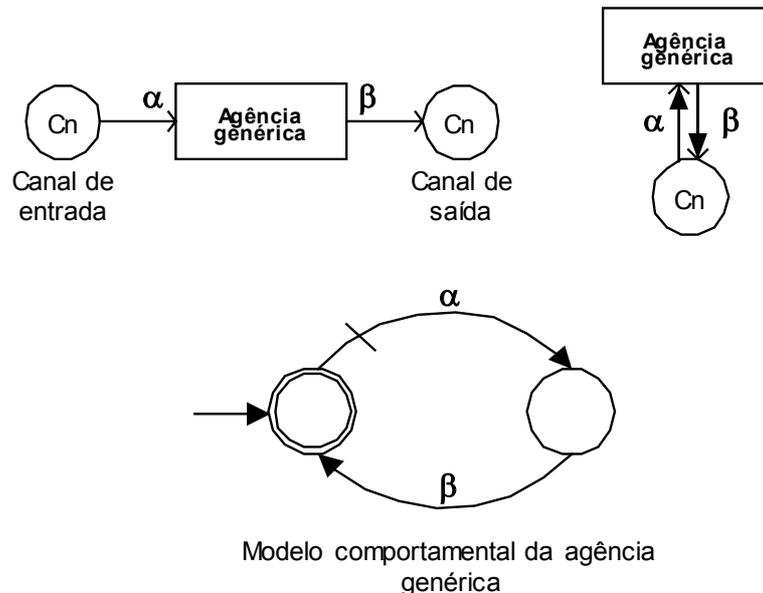


Figura 3.27 - Exemplo de modelo comportamental de uma agência (SANTOS, 2003).

Como forma de minimizar esta dificuldade SANTOS propõe ainda a utilização de um sensor de presença nos canais sem atividade, cujo modelo em autômato contém um único estado em que um evento em autolaço (*Self Loop*) sinaliza a presença da peça. A utilização de sensores na modelagem de SED também é estudada por BOUZON (2004) que, como SANTOS, mostra que esta abordagem simplifica as especificações permitindo a redução da complexidade computacional associada ao processo de síntese de controladores através da teoria de controle supervisorio modular local (TCSML).

Os resultados obtidos por SANTOS mostram que a proposta é bastante adequada para as classes de sistemas de manipulação e montagem, facilitando o trabalho de projeto através da sistematização dos procedimentos e das informações sobre soluções preestabelecidas para diversos casos. A expectativa a partir deste ponto seria, na visão de SANTOS, a ampliação da proposta para outros domínios de aplicação que não os SMMA e para os casos deste tipo de sistema, ampliar o arcabouço de especificações a ponto de expandir as informações de projeto para todas as configurações construtivas destes.

De maneira resumida, pode-se descrever as tarefas ilustradas na Figura 3.26 como segue:

- Estabelecer o modelo funcional e estrutural global – utilizando a rede C/A (Figura 3.1) e o modelo de sistemas automáticos representado por ela (Figura 3.4), estabelece-se o modelo macro do sistema especificando todas as entradas e saídas importantes;
- Refinar o modelo estrutural e funcional – desmembrar o modelo macro em sub-funções e estruturas a exemplo da Figura 3.2;
- Estabelecer variantes para o modelo estrutural e funcional – estabelecer diferentes sub funções e/ou estruturas;
- Selecionar estruturas funcionais adequadas – selecionar dentre as diversas variantes a que julgar mais adequada;
- Modelar agências e especificações, sintetizar controladores e testar modularidade – utilizar a teoria de linguagens e autômatos e controle supervisorio modular local, para modelar o comportamento do sistema, as leis de controle e gerar os controladores;
- Estabelecer princípios de solução para as agências, combinar e selecionar princípios – utilizar ferramentas de auxílio adequadas como a matriz morfológica e estabelecer princípios de solução para as funções das agências, elétricos, mecânicos, hidráulicos, etc;
- Estabelecer seqüências operacionais dos sistemas de atuação – estabelecer os detalhamentos operacionais de cada módulo do sistema produto utilizando modelos em SFC, por exemplo;
- Revisar a estrutura funcional adequando-a ao principio selecionado – dependendo do princípio selecionado algumas funções e estruturas podem ser refinadas ou condensadas, ou ainda aparecerem como estruturas similares;
- Acoplar as seqüências operacionais à estrutura de controle supervisorio – estabelecer as variáveis de acoplamento e a estrutura de implementação a ser utilizada;
- Comparar os resultados com os critérios técnicos e econômicos de projeto – verificar se a concepção é viável técnica e economicamente;
- Verificar se o resultado atende às especificações de projeto estabelecidas no início do processo.

Após estas atividades tem-se uma concepção mecânica e de controle, cujo projeto foi realizado de forma paralela não beneficiando isoladamente nenhuma das duas partes. Este modelo de projeto conceitual favorece as atividades de projeto, não somente pela sistematização mas também pelo paralelismo entre projeto mecânico e de controle, além da recursividade existente no processo que permite alterações nos modelos anteriores para facilitar a criação dos posteriores e dos supervisores modulares.

3.7 Comentários

Como foi visto, a modelagem de sistemas automáticos, etapa crucial no projeto destes sistemas, pode ser realizada segundo basicamente três perspectivas: funcional, estrutural e comportamental. Seguindo a linha de pesquisa a qual este trabalho está inserido, adotou-se a Rede Canal/Agência como ferramenta de modelagem, por ela proporcionar uma visão segundo as perspectivas funcional e estrutural concomitantemente, e a teoria de controle supervisorio modular local, pois ela permite a síntese automática de controladores modulares facilitando a implementação.

Os passos adotados no projeto conceitual de SANTOS serão tomados como referência e ponto de partida para a modelagem e controle da comunicação com o ambiente externo em sistemas automáticos, proposta central deste trabalho.

CAPÍTULO 4

CARACTERÍSTICAS DO PROJETO DE SISTEMAS AUTOMÁTICOS

4.1 Introdução

A atividade de projeto de sistemas automáticos pode não ser considerada algo novo já que os equipamentos automáticos já fazem parte de nosso cotidiano, no entanto estes equipamentos são relativamente novos se comparados às máquinas desenvolvidas pela ciência humana desde os seus primórdios. Mas mais recentes ainda são os esforços de pesquisa para a criação de métodos de projeto de engenharia que tornem a atividade de projeto mais eficiente e gerem resultados finais com mais qualidade, lembrando que ainda não se tem um método completo e definitivo para o projeto de sistemas automáticos. Mas o que compreende a atividade de projeto de sistemas automático? Quais são as diferenças destas atividades em relação a uma outra classe de projeto de engenharia? Que mudanças estas possíveis diferenças entre as atividades provocam em uma metodologia de projeto? Estas diferenças tornam os métodos de projeto já estabelecidos inadequados ao projeto de sistemas automáticos?

Para responder a estas e outras questões será necessário primeiro avaliar o papel de uma metodologia no processo de projeto de sistemas técnicos, bem como estabelecer quais as diferenças entre outros sistemas técnicos e os sistemas automáticos. Assim, será possível compreender a problemática de projeto envolvida no caso dos sistemas automáticos e estabelecer qualitativa e quantitativamente as técnicas e ferramentas necessárias à sua solução. Este entendimento deve auxiliar a visão global do problema que se aborda neste trabalho.

Desta forma, o que se busca neste capítulo é avaliar o grau de adequação de metodologias estabelecidas no projeto de sistemas técnicos, para o caso de sistemas automáticos, justificando o investimento de esforço em desenvolver novas metodologias e técnicas.

4.2 A influência da metodologia de projeto na qualidade dos resultados

O processo de projeto talvez seja a atividade mais nobre da engenharia e também a mais complexa, pois ela envolve conhecimento, intuição, subjetividade, decisão entre outros fatores que podem ser interpretados como complicadores para as atividades de projeto. No ambiente industrial não é muito incomum que o projeto de sistemas técnicos, nos quais podemos incluir desde dispositivos de processo, software, subsistemas de produtos finais e até os próprios produtos finais, seja feito de maneira muito desorganizada e alicerçada nos conhecimentos e experiências dos projetistas, quando existe mais de um projetista.

Conforme citado por FORCELLINI (2002), este cenário se agrava com as novas exigências do mercado globalizado por produtos cada vez mais modernos, eficientes, baratos e de qualidade, o que torna o seu ciclo de vida cada vez mais curto e exigindo cada vez mais do processo de projeto. Lembrando que o cenário de desenvolvimento em questão é tal que o conhecimento e a experiência pessoal são seus sustentáculos, não é difícil imaginar que acelerar o processo de projeto para atingir às expectativas do mercado acaba gerando uma inconsistência com relação à eficiência e qualidade.

O conhecimento e experiência pessoais são fatores determinantes na solução de problemas de projeto, mas não determinam o sucesso do projeto, principalmente diante das exigências da globalização. No entanto, conforme descrito por FUTAMI (2002), a organização do conhecimento cria vantagens competitivas para as empresas a medida que evita a reincidência de erros, e a sistematização do processo de projeto, através de metodologias e ferramentas de auxílio, favorece a utilização deste conhecimento.

Assim, para estabelecer a influência da utilização de uma metodologia de projeto na qualidade dos resultados é interessante compreender os conceitos envolvidos tais como: processo de projeto, metodologia, qualidade e conhecimento. Para tornar este estudo mais compreensível, estes conceitos serão abordados nos subtópicos seguintes, dando ênfase, sempre que possível, ao caso de projeto de produtos ao invés de generalizar-se para todos os casos de projetos de sistemas técnicos.

4.2.1 O processo de projeto

O processo de projeto pode ser visto como: "...um conjunto de atividades que busca, a partir da identificação das necessidades, alcançar as especificações detalhadas para a construção ou implementação de um sistema técnico." (DE NEGRI, 1996)

Desta maneira DE NEGRI alerta que se deve identificar necessidades e evoluí-las em especificações que servirão de guia para a síntese de um sistema técnico. Deve-se entender que sistema técnico, no âmbito da engenharia, vai desde o software até os produtos manufaturados e que para qualquer destes a definição de processo de projeto se aplica.

Para o caso específico de produtos industriais, o processo de projeto torna-se processo de desenvolvimento de produtos, como descrito por JURAM (1992) apud FUTAMI (2002), e pode ser definido como um processo experimental de escolha e definição de características que correspondam à satisfação das necessidades dos clientes, e que para tal consiste da transformação de informações com um tempo limitado.

Mais ainda: "o projeto de um componente ou um sistema apresenta, em cada caso, características e peculiaridades próprias. Mas à medida que um projeto é iniciado e desenvolvido desdobra-se uma seqüência de eventos, numa ordem cronológica, formando um modelo, o qual quase sempre é comum a todos os projetos". (BACK, 1983)

Pode-se concluir que o processo de projeto de sistemas, e em particular de produtos, é uma atividade complexa em que as fases de identificação da informação e do tratamento que ela recebe são quase sempre comuns a todos os processos de projeto (ver Capítulo 2) e que atividades como identificar, definir, escolher, avaliar, entre outras, são componentes intuitivos do processo dependentes do conhecimento e experiência dos projetistas.

4.2.2 O conhecimento no projeto

Como já foi dito anteriormente, o conhecimento é essencial para o projeto de sistemas, mas somente ele não garante o cumprimento das metas estabelecidas para o resultado final. Este ponto de vista é confirmado por AMARAL e ROZENFELD (2001) citando que as atividades do desenvolvimento de produtos são essencialmente de caráter criativo, ou seja, dependem do conhecimento e experiências dos projetistas. Entretanto, este caráter criativo pode ser aprimorado pelo uso intensivo de técnicas e métodos de auxílio.

Abstendo-se momentaneamente desta última informação, em que métodos e técnicas podem aprimorar o poder criativo, vê-se que o conhecimento totalmente pessoal parece ser o ponto de partida e a base de sustentação de toda a atividade de desenvolvimento de sistemas. Segundo FUTAMI (2002) e AMARAL et al (2001), este conhecimento pessoal, denominado Tácito, está contido na base e ponto de partida de projetos, porém não é o único, uma vez que ele compõe o conjunto do conhecimento total juntamente com duas outras formas de conhecimento, o explícito e o organizacional. Neste contexto, em relação às atividades de projeto tem-se que:

“O conhecimento é criado através da interação e do compartilhamento que ocorre entre as pessoas na execução dessas atividades, e o fluxo de informações ocorre de forma caótica durante esse processo. O conhecimento tácito que emerge desse processo interativo é a base do processo de criação do conhecimento organizacional.” (FUTAMI, 2002)

Em outras palavras, o conhecimento organizacional é o fruto da experiência corporativa disseminada de maneira explícita na corporação, podendo inclusive estar contida no conhecimento explícito que nada mais é que toda forma de informação capaz de ser verbalizada, transportada, distribuída e compartilhada, ou seja, normas, registros bibliográficos, livros, procedimentos de trabalho, entre outros.

Pode-se perceber que todas estas formas de conhecimento disseminado geram, durante o processo de projeto, um fluxo de informações de forma caótica como citado por FUTAMI e este caos certamente não facilita a realização da atividade fim, o projeto de sistemas ou produtos. Pode-se, então ratificar a informação dada por AMARAL e ROZENFELD em que a utilização de técnicas e métodos aprimora o processo criativo, à medida que ela sistematiza a utilização dos conhecimentos existentes e a geração de novos, modificando o fluxo de informação de sua forma caótica original.

4.2.3 Qualidade do projeto e de seus resultados

Nota-se, pelo exposto nos subitens anteriores, que existe uma forte ligação entre conhecimento, informação e atividades de projeto e que técnicas e métodos de ação otimizam os resultados no sentido de atingir os objetivos traçados para sua realização. Relembrando as definições de processo de projeto pode-se ainda completar com a definição de projeto em engenharia como sendo, “[...] uma atividade orientada para o atendimento das necessidades humanas [...]” (BACK, 1983). Por necessidades humanas, pode-se entender como sendo as exigências do mercado globalizado por custos menores, eficiência, modernidade e qualidade.

A qualidade, que aparece como uma das necessidades, tem um conceito bastante amplo na consciência geral, mas no âmbito industrial o dicionário HOUAISS (2001) define qualidade como: “O cumprimento estrito das normas preestabelecidas de produção.”.

Mais ainda, com referência a produtos, SLACK et al. (1997), cita as definições de qualidade em cinco abordagens diferentes, a saber:

- Abordagem Transcendental – sinônimo de excelência inata, por exemplo, o relógio de qualidade nesta abordagem poderia ser o Rolex;
- Abordagem baseada em manufatura – abordagem em que a manufatura se preocupa em fazer produtos livres de erros, ou seja, que correspondem estritamente às especificações de projeto;
- Abordagem baseada no usuário – assegura que o produto seja adequado ao propósito e ao usuário que se destina;
- Abordagem baseada no produto – vê a qualidade como um conjunto mensurável e preciso de características que são requeridas para satisfazer o consumidor;
- Abordagem baseada no valor – define a qualidade e função dos custos e do preço.

Assim, adotando-se uma generalização do conceito de qualidade em relação a sistemas e mais especificamente em relação a produtos, pode-se definir que a exigência do mercado globalizado, sem perda de generalidades, é exclusivamente por qualidade. Por consequência, não seria incoerente dizer que o processo de projeto, que visa alcançar especificações que atendam às necessidades dos usuários ou clientes, pode ser visto como um processo com o objetivo de atingir resultados com qualidade.

Abusando um pouco deste conceito de qualidade, também é correto dizer que um projeto feito com qualidade é aquele que consegue atingir as expectativas mínimas com relação aos resultados e que esta qualidade será tanto maior quanto maior for a proximidade entre o resultado obtido e o idealizado pelo processo.

4.2.4 Metodologia

Como já foi discutido, propõe-se que os métodos e técnicas sejam ferramentas que auxiliam na organização do conhecimento e na sistematização das atividades do processo de projeto e de acordo com HOUAISS tem-se:

“Técnica – Conjunto de procedimentos ligados a uma arte ou ciência.”

“Método – procedimento, técnica ou meio de se fazer alguma coisa especialmente de acordo com um plano - processo organizado, lógico e sistemático de pesquisa, instrução, investigação, apresentação e etc - ordem lógica ou sistema que regula uma determinada atividade.”

Completando o raciocínio tem-se também segundo HOUAISS que:

“Metodologia é o corpo de regras e diligências estabelecidas para realizar uma pesquisa.”

Como pode ser visto, o intuito de uma metodologia é realmente o de regular, sistematizar e organizar uma atividade de maneira lógica como para o caso das atividades de projeto. Lembrando o que BACK (1983) cita, os modelos das atividades de projeto são quase sempre comuns a qualquer projeto, mas seria precipitado concluir que então pode existir uma única metodologia que solucione os problemas de projeto para qualquer caso, em outras palavras, não seria correto afirmar que qualquer projeto pudesse ser organizado, sistematizado ou otimizado por uma mesma metodologia.

Neste sentido YOSHIKAWA (1989) apud DUFOUR (1996), comenta que não existe uma teoria de projeto que possa ser adequada à solução de todos os tipos de problemas e que as metodologias de projeto existentes diferem no nível de abordagens e profundidade de detalhamento das atividades que as compõem. Cada metodologia existente é então mais adequada a certos tipos de projeto, isto é, apresenta resultados melhores para a solução de alguns tipos de problemas.

A inadequação de algumas metodologias para alguns modelos de projeto não implica necessariamente a impossibilidade de sua aplicação, o que pode ocorrer é uma qualidade inferior de resultados com esta escolha em detrimento da escolha de uma metodologia mais adequada. Esta diferenciação pode ser explicada pelas diferentes abordagens feitas nas diversas metodologias, como explicado por DUFOUR dizendo que existem várias escolas ou linhas de pensamento que abordam os problemas sob diferentes pontos de vista. Por exemplo, a escola Semântica preocupa-se com o fluxo de energia dos sistemas e as escolas Psicológicas e Filosóficas preocupam-se com fatores como a criatividade e os aspectos do pensamento humano, respectivamente. Com base nestas linhas de pensamento, surgiram as diversas metodologias descritas na literatura para tentar auxiliar o desenvolvimento de projetos cada uma a seu modo.

4.2.5 Comentários

Para avaliar a influência da metodologia de projeto na qualidade dos resultados, conforme proposto inicialmente, foi necessário conceituar processo de projeto, conhecimento e qualidade, além de suas correlações. Viu-se que o projeto é uma transformação de informações segundo um modelo quase sempre comum a todos os tipos de problemas, visando a satisfação de certas necessidades, e que para isto o conhecimento em todas as suas formas é essencial. Também foi visto que qualidade pode ser definida como o atendimento das expectativas em relação ao projeto e aos resultados destes.

Desta maneira relacionou-se o processo de projeto com a qualidade dos seus resultados através do atendimento das especificações definidas, podendo ser concluído que melhorias no processo de projeto certamente provocam melhorias na qualidade dos resultados.

Finalmente pôde ser definido o conceito de metodologia que, como foi visto, tem o intuito de otimizar atividades como, por exemplo, as atividades de projeto. As metodologias de projeto, em particular, auxiliam na organização dos conhecimentos existentes e na formação de novos e também sistematizam as atividades do processo, contribuindo assim para a memória de projeto que acaba incrementando o conhecimento explícito e organizacional das instituições.

Pode-se seguramente dizer que as metodologias de projeto melhoram o processo de projeto, guardadas as proporções da adequação de cada método com certos tipos de problemas, e que esta melhoria é tanto maior quanto mais adequada a metodologia for ao problema. Deve-se levar em consideração que os métodos são ferramentas de auxílio e que sem um nível de conhecimento adequado e acesso às informações pertinentes eles não podem melhorar resultados ruins a ponto de transformá-los em bons.

Finalizando, pode-se dizer que existe uma relação entre o processo de projeto e a qualidade dos resultados e que o primeiro realmente tem influência no segundo. Além disto pode-se concluir que a magnitude desta influência não é uma constante e depende proporcionalmente dos fatores já citados, de adequação da metodologia, nível de conhecimento e quantidade e qualidade das informações disponíveis.

4.3 Particularidades entre sistemas técnicos e automáticos

Prosseguindo no caminho para atingir às respostas das questões que introduziram este capítulo e tento esclarecido, nos itens anteriores, algumas definições e correlações pertinentes ao tema, realiza-se na seqüência a discussão sobre como é, ou deveria ser, composta uma metodologia de projeto para sistemas automáticos e quais as principais diferenças entre esta metodologia/sistema e outros existentes.

Novamente, para enriquecer a discussão, algumas definições serão introduzidas de maneira resumida, visto que o tema é admiravelmente abrangente.

4.3.1 Sistemas técnicos

O termo 'sistema' pode ser definido de maneira bastante genérica aplicando-se a entidades relacionadas a diversas áreas do conhecimento tais como a física, química, biologia, economia, política, sociologia, entre outras (DE NEGRI, 1996). No contexto da engenharia de projeto, DE NEGRI define os sistemas técnicos como sendo: "...dispositivos capazes de atender às suas necessidades [do homem], tais como máquinas, produtos, construções, equipamentos dentre muitas outras designações correntes."; e ainda completa com: "Um sistema que age sobre energia, matéria e informação."

Os sistemas técnicos em geral estão presentes em nosso cotidiano, na forma de nossos eletrodomésticos, automóveis, residências, ferramentas de trabalho, entre muitos outros exemplos, ou seja, quase tudo que é projetado e construído para o nosso bem estar, conforto e auxílio. O projeto de alguns tipos de sistemas técnicos já possui metodologia e até um conjunto de metodologias de projeto cujas ferramentas, que evoluem constantemente, auxiliam no desenvolvimento e inovação destes sistemas com grande velocidade, como é o caso de automóveis, aparelhos de telefone celular, eletrodomésticos etc.

No entanto, algumas classes de sistemas técnicos ainda não possuem um método de projeto estabelecido, o que pode provocar um número bastante grande de dificuldades quanto a sua criação, inovação e atualização, o que não significa que estes tipos de sistema não sejam projetados e construídos.

4.3.2 Sistemas automáticos

Os sistemas automáticos estão inseridos no contexto de termos amplamente conhecidos como "automação", "mecatrônica" e "controle e automação" que não definem exatamente as mesmas áreas de conhecimento, mas são bastante correlatos no sentido que eles se completam para se definir um sistema automático.

Neste sentido, DE NEGRI (2004) e DE NEGRI e VIEIRA (1997) citam o conceito de mecatrônica como sendo uma tecnologia que reúne várias áreas de conhecimento e tecnologias como forma de estabelecer um embasamento teórico e formular métodos práticos buscando promover uma melhor comunicação entre especialistas e a adoção de soluções inovadoras para muitos problemas de projeto. Mais ainda, ressaltam que o desenvolvimento de um sistema mecatrônico deve buscar realizar funções não disponíveis, escolher tecnologias adequadas às funções, aumentar a flexibilidade durante o desenvolvimento e utilização do sistema, integrar fisicamente componentes eletrônicos e eletromecânicos e melhorar a organização estrutural e manutenibilidade deste sistema.

DE NEGRI (2004) define então os termos sistema mecatrônico, sistema de automação e sistema de controle como segue:

- **Sistema mecatrônico:** sistema construído para automatizar ou controlar processos ou tarefas que incluem diversas tecnologias tais como mecânica, hidráulica, pneumática, elétrica, eletrônica e informação. Distinguindo-se em:
 - Dispositivos mecatrônicos: sistemas com grande união física entre componentes, em relação aos sistemas em que os componentes são interconectados, sem modificação significativa das características mecânicas ou elétricas de cada um. Ex: Aparelhos de CD, máquinas fotográficas etc.;
 - Equipamentos mecatrônicos: sistemas onde é possível identificar sensores e atuadores como dispositivos completos conectados com outros através de portas físicas como conectores elétricos, eixos, tubos etc...Ex: máquinas ferramentas, robôs industriais, controladores de turbinas etc.;
- **Sistemas de automação:** conjunto de componentes interconectados com a função principal de realizar uma ou mais ações segundo uma lógica pré-determinada e em resposta ao estado em que se encontra o equipamento e à ocorrência de eventos;
- **Sistemas de controle:** conjunto de componentes interconectados com a função principal de realizar uma ou mais ações que são observadas ao longo do tempo e cuja modificação decorre da aplicação de sinais de entrada.

No caso de sistemas de automação, as ações podem ser o avanço ou recuo de um cilindro, acionamento ou parada de um motor elétrico, por exemplo, e os eventos correspondem à sinalização do término de uma tarefa ou a mudança do estado de um dispositivo como, por exemplo, o acionamento de um botão, o sinal de um sensor de presença de peça ou a chave de fim de curso de um cilindro. Já o caso de sistemas de controle as ações podem ser a regulagem de grandezas físicas como força, velocidade ou posição de um cilindro, bem como pressão ou corrente em circuitos hidráulicos e elétricos, respectivamente, onde os sinais estão relacionados às variáveis observadas no tempo.

Assim, o termo **automação** é visto por DE NEGRI como mais abrangente que o termo **controle** visto que pela definição apresentada o segundo pode claramente fazer parte da lógica do primeiro. Por sua vez, este mesmo autor escreve que: “Tem-se adotado o termo **sistema automático** para designar uma aplicação que envolva automação e/ou controle, ou seja, pode-se observar o problema segundo uma visão lógica ou então de maneira mais aprofundada, avaliando, ao longo do tempo, a resposta da posição, força, velocidade, vazão ou qualquer outra variável.” (DE NEGRI, 2004; grifo do autor.).

Observa-se que os sistemas automáticos são também sistemas técnicos, cujas características peculiares fazem necessária a reunião de amplo conhecimento multidisciplinar para que se possa projetá-los de maneira eficiente. Pode-se encarar os sistemas automáticos como sendo um tipo (ou vários tipos) de produtos industriais em que as características (ou atributos) são qualitativa e quantitativamente complexas.

4.4 Método de projeto de sistemas automáticos

Como foi comentado anteriormente, não existe um método totalmente estabelecido para o projeto de sistemas automáticos mas muitos avanços já foram realizados nesta direção. Diversas ferramentas, técnicas, métodos e modelos vêm sendo desenvolvidos para a solução de problemas de projeto principalmente em aplicações particulares dos sistemas automáticos como os domínios da hidráulica, pneumática e acionamentos elétricos.

O estudo sobre **modelagem** de sistemas, descrito no Capítulo 3, é um exemplo do esforço em aprimorar a base de conhecimento técnico necessária às atividades de projeto. Atualmente, existem inúmeros modelos de representação dos mais diversos tipos de sistemas baseados em **perspectivas** adequadas ao entendimento, ensaio, teste, visualização, demonstração, cálculo, simulação etc.

Muitas abordagens sobre **descrição funcional**, aspecto fundamental para a concepção de sistemas técnicos, fazem parte da gama de ferramentas disponíveis ao projeto. Um estudo sobre as características de algumas destas descrições, talvez as mais representativas, foi realizado por SANTOS (2003), que avaliou descrições funcionais como as da Escola Alemã de Projeto e as denominadas de Representação Funcional, Engenharia de Requisitos, além das voltadas ao Projeto de Sistemas mecatrônicos, Projeto de Sistemas Automáticos e Projeto de Sistemas Automatizados de Manufatura. Dentre as descrições funcionais estudadas, SANTOS julgou que, para o caso de projeto de SMMA, a **voltada ao Projeto de Sistemas Automáticos**¹, que utiliza um modelo baseado em Rede Canal/Agência², é a mais adequada por contemplar uma perspectiva funcional e estrutural concomitantemente.

Outro aspecto bastante desenvolvido é a **descrição comportamental** dos sistemas, que caracteriza a reação que este terá devido às suas entradas, baseado em seu estado atual e anteriores ou não e em suas saídas. Os modelos comportamentais são caracterizados pelo tipo de sinal processado pelo sistema (CASSANDRAS e LAFORTUNE, 1999, apud, DE NEGRI, 2004) e se dividem em **Modelos a estado contínuo** e **Modelos a estados discretos**, cujas variáveis de entrada e saída são, respectivamente de amplitudes contínuas geralmente dependentes do tempo e de amplitude discreta com comportamento do sistema guiado pelo tempo ou por eventos.

Associados às classes de modelos comportamentais está um grande conjunto de **ferramentas matemáticas** que formalizam o comportamento tais como, equações diferenciais, transformadas de *Laplace*, funções de transferência, diagrama de blocos (OGATA, 1993) para casos de estado contínuo e álgebra Booleana, tabela verdade, diagramas lógicos, diagrama de contato, Grafcet (ou SFC), redes de Petri marcadas, teoria de linguagens

¹ (DE NEGRI, 1996) descrita no item 3.1 .

² (HEUSER, 1990) (DE NEGRI, 1996) (SANTOS, 2003).

e autômatos, entre outras (BOLLMANN, 1996) (CARDOSO, 1997) (CURY, 2001 e 2003), para o caso de estados discretos.

Da mesma forma, existem técnicas e ferramentas de **análise e projeto de controladores** também para ambos os casos de sistemas a estados contínuos e discretos, como por exemplo o Lugar das Raízes e a Análise da Resposta em Frequência (OGATA, 1993), para o primeiro caso, e o Mapa de Karnaugh-Veitch e Controle Supervisório para o segundo caso (BOLMANN, 1996) (CURY, 2001 e 2003).

Com este montante de conhecimento e aliado aos métodos de projeto de produto, tecnologias atuais de informática industrial, sensores e atuadores, além de outros conhecimentos e tecnologias em áreas diversas como simuladores e inteligência artificial, dentre muitos outros, é que novos esforços em gerar conhecimento que facilitassem o **projeto e implementação sistemas automáticos** tiveram início.

Pode-se dar como exemplo de contribuição para o projeto e implementação de sistemas automáticos, de maneira mais pontual, a proposta de estrutura de controle para implementação em CLP de QUEROZ et al. (2001) e a sistematização de projeto de sistemas automáticos em Hidráulica e pneumática de ATTÍE (1998), e de maneira mais geral, o projeto conceitual de SMMA proposto por SANTOS (2003). Este último representa um grande passo no sentido de se estabelecer uma metodologia completa de projeto em automação em função de contemplar em um só modelo de projeto a **concepção das partes físicas e de controle de maneira simultânea**.

A esta altura pode-se notar claramente que o campo de atuação da automação como um todo é extraordinariamente amplo e que muito conhecimento já foi agregado à área, mesmo que de maneira relativamente pouco geral. Este último fato ocorre pela diversidade de processos, tecnologias e áreas da ciência que estão envolvidos nos processos de automação, tornando difícil generalizar modelos, ferramentas, técnicas e métodos para todos os casos, o que não quer dizer que não se possa criar uma linha de atuação, um plano sistemático, para projetar sistemas automáticos em geral, que utilize o conhecimento já estabelecido.

4.5 Comentários

Através da ambientação proporcionada no decorrer deste capítulo pode-se agora responder às perguntas do item 4.1 que motivaram esta discussão. Primeiramente deve-se ressaltar que como ainda não se dispõe de uma metodologia totalmente estabelecida e sim de conjuntos de métodos específicos de cada área de atuação da automação, o comentário sobre uma possível metodologia geral para projetos de sistemas automáticos é ainda uma conjectura.

Como se viu anteriormente, a atividade de projeto de sistemas automáticos é compreendida das atividades de descrição funcional, descrição comportamental, a modelagem

adequada a cada uma delas e o projeto de controladores, não deixando de lado a pesquisa sobre as informações relevantes ao projeto, sua simulação, implementação e testes.

Pode-se notar que cada atividade do projeto de sistemas automáticos encontra uma atividade similar, quando não idêntica, no caso de outras classes de projeto de engenharia. No entanto, aquelas atividades geralmente não estão integradas no sentido de se contemplar o projeto físico e lógico do sistema e suas ferramentas de projeto são de tal forma diversificadas, devido à gama de tecnologias envolvidas, que promover esta integração e simultaneidade não é um exercício trivial. Assim, conclui-se que a grande diferença entre as diversas classes de projeto em engenharia e o projeto de sistemas automáticos é que a execução deste último envolve, via de regra, grande diversidade de tecnologias, nem sempre compatíveis, tornando-a sensivelmente mais complexa.

Esta diferença relatada entre atividades de projeto não deve forçosamente inviabilizar a utilização de alguma metodologia estabelecida de projeto de engenharia para o caso de sistemas automáticos. Em princípio, modelos como o Consensual (Capítulo 2), concebidos de maneira a atender ao conjunto dos produtos industriais e sistemas técnicos em geral, podem ser utilizados para o projeto em automação incluindo muitas de suas ferramentas geralmente associadas, já que as atividades são correlatas.

Entretanto, viu-se também que as metodologias existentes não são adequadas a todos os tipos de projeto e que os resultados destes são de qualidade tanto maior quanto mais adequada for a metodologia utilizada. Isto quer dizer que, em se possuindo uma metodologia totalmente adequada aos sistemas automáticos em geral, ter-se-á da mesma maneira condições de realizar projetos de forma otimizada, ou seja, mesmo sendo complexo o resultado poderá ser de qualidade (mínima ou nenhuma diferença entre o resultado obtido e o idealizado).

Esta metodologia certamente deve contemplar a concepção simultânea da parte física e lógica a exemplo do modelo conceitual para SMMA, além de sistematizar as atividades centrando nas características comuns a todos os tipos de sistemas automáticos ou, na impossibilidade disto, pelo menos agrupar os de características afins diminuindo a árvore de possibilidades de ação no projeto. A escolha de ferramentas adequadas a cada atividade, criação de base de padrões de modelos funcionais e comportamentais para referência, bem como de suas especificações de controle, entre outras coisas, simples ou muito complexas, talvez estejam dentre os próximos objetivos a serem atingidos para que se possa estabelecer, por exemplo, uma ferramenta computacional que oculte toda a complexidade envolvida no vasto domínio da automação e deixe para o projetista o livre exercício de sua própria criatividade.

CAPÍTULO 5

MODELAGEM DA COMUNICAÇÃO COM O AMBIENTE EXTERNO DE UM SISTEMA AUTOMÁTICO.

Conforme mencionado anteriormente, um problema real de projeto foi utilizado no decorrer deste trabalho para ponderar sobre a necessidade de incremento ou modificação do conhecimento existente quanto ao projeto de sistemas automáticos. O problema em questão é o controle de uma unidade de potência hidráulica que será usada como fonte de recursos a duas bancadas didáticas de estudo e pesquisa a serem implementadas no laboratório de sistemas hidráulicos e pneumáticos (LASHIP) da UFSC. O sistema prevê a existência de painéis de comando em que usuários irão compartilhar recursos disponíveis e que são severamente limitados na unidade de potência, além do controle dos componentes hidráulicos.

Observa-se inicialmente que se trata de um sistema diferente de um SMMA, no qual se tem um fluxo de entidades contáveis, tendo como função básica o controle do direcionamento de energia proveniente de fontes contínuas de vazão. A avaliação deste sistema mostrou que alguns pressupostos considerados por SANTOS (2003) não são exatamente adequados para resolver o problema, pedindo uma reformulação de modelos e especificações padronizadas em seu trabalho. A necessidade de controlar (limitar) a interação com os usuários e compartilhar os recursos entre eles também estimularam novas interpretações do problema e, conseqüentemente, o surgimento de novas propostas de modelagem.

Surgem então, novos aspectos para a implementação do controle supervisorio em CLP e, com base nas propostas de implementação feitas por QUEIROZ et al. (2001), algumas alterações são introduzidas para levar a cabo o funcionamento do sistema estudado. Para tanto, algumas hipóteses serão apresentadas quando convenientes, mas em aspectos gerais pode-se considerar que:

- O trabalho em questão está relacionado a sistemas a eventos discretos (SED) e desta forma são considerados o sistema estudado (unidade de potência hidráulica) e todos os exemplos contidos neste texto;
- Considera-se também que os sistemas que compõem a unidade de potência hidráulica são limitados quanto aos seus números de estados e possuem comportamento bem conhecido; assim, os modelos em autômatos serão determinísticos e de estados finitos;
- Todos os eventos relevantes aos sistemas estudados são observáveis.

5.1 Motivação

Tendo em mente a Figura 3.4, sabe-se que na modelagem tradicional de sistemas automáticos o ponto de partida é geralmente a modelagem funcional e estrutural do sistema energético-material, que cumpre as ações físicas e/ou químicas a que se destina o dispositivo automático. O sistema de informação é então modelado sob o ponto de vista funcional e comportamental, de forma a inserir o controle a eventos discretos do sistema energético-material bem como o controle contínuo ou as seqüências de operação necessários ao funcionamento de cada dispositivo que compõe o sistema.

Um sistema automático também recebe informações do ambiente externo como forma de comandar ou configurar a operação do sistema de acordo com as necessidades de um sistema anterior, posterior ou mesmo de um usuário (operador) que queira escolher os modos de operação e receber informações sobre o sistema através de um painel, por exemplo (Figura 5.1).

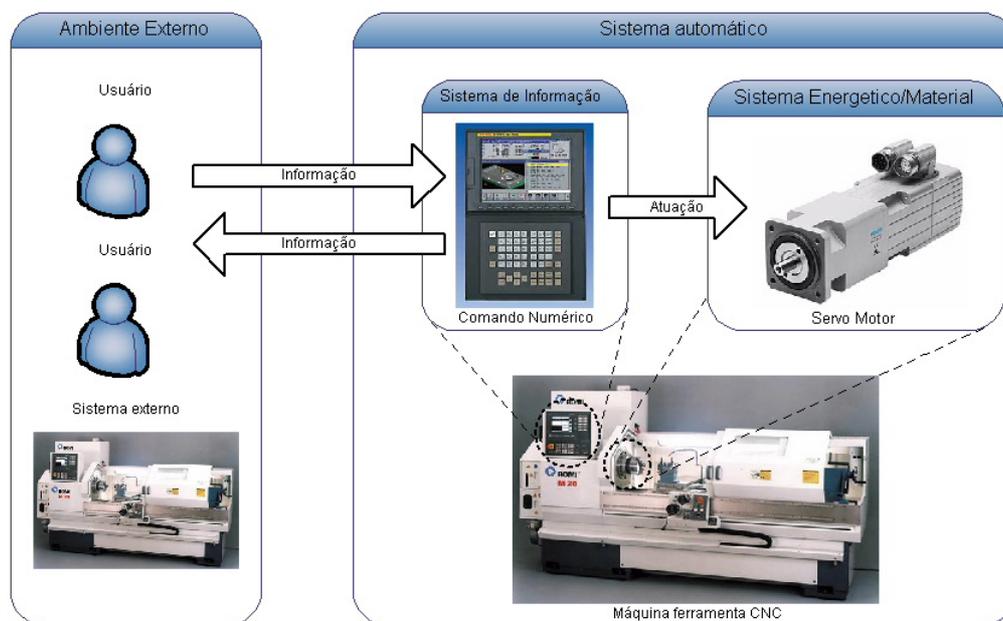


Figura 5.1 - Comunicação entre usuário e um sistema controlado por computador.

A comunicação entre sistemas e usuários pode ser desde um simples comando para iniciar o processo até um conjunto complexo de entradas e saídas de informação que podem ter que ser interpretadas e/ou controladas, total ou parcialmente, no decorrer das atividades do sistema automático. Quando a comunicação é identificada em um processo de projeto, o projetista faz um levantamento das informações relevantes que devem ser trocadas no processo de comunicação, mas ainda não especifica quais os meios físicos pelos quais estas informações irão fluir. Esta etapa de projeto é bastante similar ao projeto do sistema energético-material em que o projetista modela o sistema identificando suas funções sem especificar os princípios de solução que serão adotados para suas execuções.

A modelagem estrutural e funcional, bem como a modelagem comportamental e a estrutura de controle de um sistema energético-material já está bem conjugada para o processo de projeto de sistemas automáticos, o que não acontece para o caso da modelagem do sistema de informação, ao menos no que diz respeito ao processo de comunicação entre o sistema automático e seu ambiente externo.

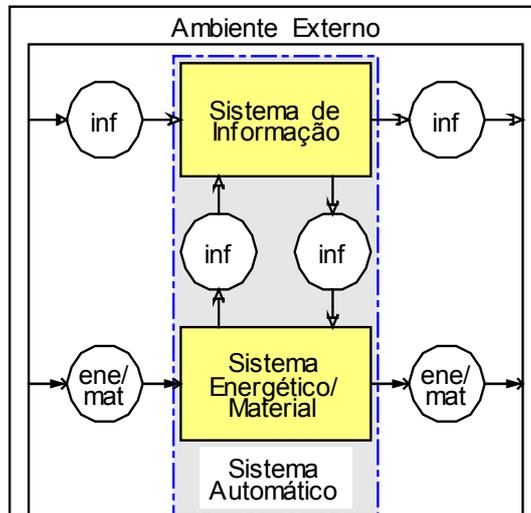


Figura 5.2 - Modelo funcional/estrutural condensado de um sistema automático.

5.2 Comunicação simples entre um sistema automático e seu ambiente externo

As informações trocadas entre o sistema automático e seu ambiente externo podem ser bastante complexas a ponto de tornar necessário seu controle e interpretação através de um modelo específico. Porém, no caso de projeto de sistemas simples, geralmente sub-sistemas de um sistema maior ou mesmo equipamentos automáticos independentes, as informações trocadas com o ambiente externo são muitas vezes bastante elementares. Para este último caso pode-se exemplificar como entradas alguns comandos de início de operação, reinicialização de sistema ou botões de emergência, e como saídas os sinalizadores simples de indicação de máquina em operação ou em emergência, alarmes de falta de peças ou quebra de componentes, etc. Estas informações podem ser de pouca ou nenhuma complexidade, não justificando a geração de modelos específicos, o que somente aumentaria a complexidade do sistema como um todo.

Os resultados finais do sistema dependem, no entanto, de que ele tenha conhecimento dos sinais externos, ou seja, eles devem ser levados em conta em alguma parte da estrutura de controle implementada no sistema de informação da máquina automática. No projeto de máquinas simples, em que o projeto de uma seqüência de operações utilizando métodos sistemáticos, como o projeto passo-a-passo de comandos seqüenciais (BOLLMANN, 1998), é suficiente para atender às especificações, os botões de emergência, início, reinicialização e as respectivas luzes de sinalização são tratados como condições de funcionamento intrínsecas ao projeto ou como condições adicionais. Desta maneira, estas

informações vindas do ambiente externo são inseridas no projeto da seqüência de operações como condições de início dos passos, desvio para um ramo específico (ramo com uma seqüência de reinicialização, por exemplo) ou no caso de botões de emergência, são adicionados em todos os passos para que a seqüência seja desviada para um passo coerente com o tipo de emergência sinalizada. Além disto, uma seqüência de operações também pode conter em algumas de suas ações o acionamento de sinalizadores luminosos que indiquem a condição (status) do sistema.

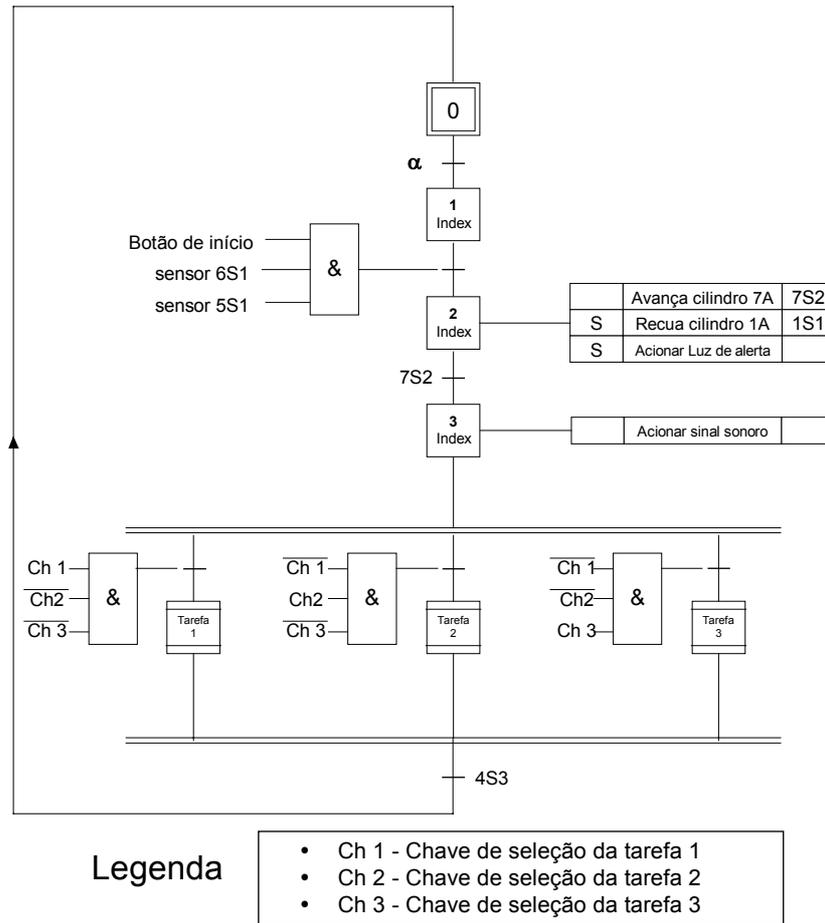


Figura 5.3 – Exemplo de seqüência operacional implementada em diagrama de funcionamento (SFC) – Entradas e saídas ao ambiente externo.

Visto que a técnica relatada aqui tem se mostrado eficiente, pode-se pressupor que ela pode ser aplicada parcial ou totalmente em sistemas mais complexos, como os que envolvem diversas seqüências operacionais e modelos de comportamento específicos para cada sub-sistema. Deve-se levar em consideração que a interpretação de quão simples é, ou não é, a informação e como ela deve ser tratada é bastante intuitiva e que em certos casos vai depender do ponto de vista e experiência do projetista.

Na modelagem do comportamento não-detalhado de um sistema (Sistema Produto) (modelos em autômatos dos subsistemas (agências) desconsiderando as seqüências operacionais), não considerar uma informação é bastante delicado e deve ser feito com bastante cuidado, uma vez que o supervisor não vai ter conhecimento de uma informação que está

influenciando o funcionamento do sistema através de suas seqüências operacionais. Um exemplo desta influência perigosa pode ser observado na implementação de um botão de emergência, como é feito tradicionalmente nas seqüências operacionais, sem o cuidado de adicionar esta condição à implementação dos modelos da planta (ou sistema produto) e do supervisor. Na modelagem e projeto do supervisor, o sinal de emergência geralmente não é considerado e se não for adicionado à implementação pode provocar uma inconsistência de controle já que as seqüências operacionais seriam reinicializadas e a planta e o supervisor não.

Assim, deve-se avaliar com cuidado quais sinais (informações) serão considerados no modelo de comportamento, gerando talvez até um modelo específico para o tratamento da informação. Também se deve analisar quais sinais influenciam somente certas seqüências de operação e que não são importantes para o controle supervísório como um todo. Para os casos em que esta complexidade é verificada e a solução tradicional em seqüências operacionais mostra-se inadequada, serão propostos, nos itens que seguem, modelos estruturais, funcionais e comportamentais destinados a auxiliar o projeto e controle da comunicação entre sistema automático e ambiente externo.

5.3 Modelo funcional e estrutural para a comunicação com o ambiente externo

Avaliando-se o modelo estrutural e funcional de um sistema automático, descrito em rede Canal/Agência, e desconsiderando-se a relação com o ambiente externo, vê-se um sistema físico comunicando-se com um sistema de processamento de informação através de uma interface que são os sistemas de atuação e medição, como mostrado na Figura 5.4.

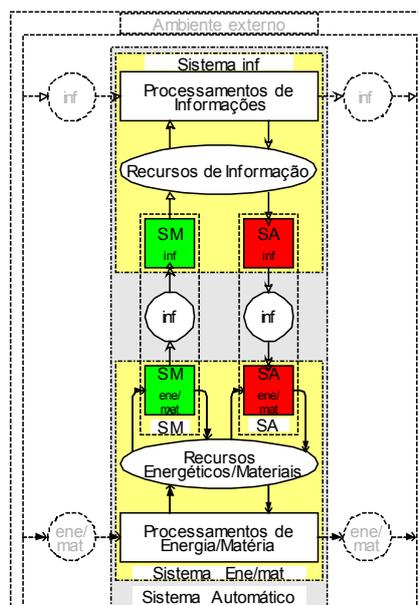


Figura 5.4 - Modelo em C/A de um sistema automático sem ambiente externo.

Observando-se com um pouco mais de detalhamento, vê-se que o sistema de processamento de informação pode ser implementado através de uma estrutura de controle

composta basicamente pelo modelo do sistema físico (que é a abstração de seu comportamento sem controle, denominado planta livre) e por uma lei de controle imposta ao sistema livre para que ele se comporte como desejado pelas especificações de projeto (Figura 3.11).

Observando-se da mesma maneira a comunicação entre o ambiente externo ao sistema automático e seu sistema de processamento de informação, verifica-se certo grau de semelhança com o modelo estrutural e funcional entre este mesmo sistema de informação e o sistema energético-material, no sentido que ambos são compostos por uma parte real com a qual deve ser estabelecida a conexão, através de sinais, com o sistema de processamento de informação onde está implementado o controlador.

Desta maneira pode-se presumir que pode ser realizada, para a comunicação com o ambiente externo, uma estrutura de controle semelhante à utilizada para o controle do sistema energético-material, em que um modelo de comportamento do sistema real (sistema físico) é implementado juntamente com uma lei de controle. Assim, uma estrutura para o monitoramento e controle das informações trocadas com o ambiente externo deve conter um modelo deste ambiente e uma lei de controle para a entrada, saída e interpretação das informações. Além disto, as informações trocadas com outros sistemas ou operadores influenciam e são influenciadas pelo sistema energético-material em questão e, por esta razão, é coerente dizer que deve existir também uma lei de controle entre ele e o modelo do ambiente externo.

Os sistemas de atuação são compostos por partes energético-materiais, que são os próprios atuadores e sensores, e partes de informação, que são os controladores específicos de cada sistema de atuação, denominados de seqüências operacionais, geralmente implementadas em SFC. Desta maneira, fazendo a ponte entre a abstração comportamental (Sistema Produto) de sistemas de atuação e seus próprios atuadores e sensores está um conjunto de seqüências operacionais que controlam o funcionamento de cada dispositivo que compõe o sistema, completando assim a estrutura de controle necessária para a perfeita operação do sistema automático. Mais uma vez, pode-se realizar uma estrutura similar que faça a ponte entre o modelo de comportamento do ambiente externo e o próprio ambiente externo, ou seja, uma interface de comunicação.

Esta interface pode existir em várias facetas dependendo do tipo de tecnologia envolvida na comunicação. Tanto a entrada como a saída de informações pode ser simples, não envolvendo nem hardware e nem software complicados para a sua implementação, como pode aparecer de maneira intrincada necessitando um tratamento via software e/ou hardware que permita o trabalho posterior do controlador. O hardware pode aparecer na forma de condicionadores de sinal, periféricos de entrada e saída de dados como IHM, PCs industriais, hardware de rede industrial etc, e o software na forma de protocolos de rede, algoritmos de interpretação de imagem e tradução padrões diferentes, seqüências de transmissão de mensagens, etc.

Desta maneira, a estrutura de controle para a comunicação com o ambiente externo pode ser realizada como descrito na Figura 5.5.

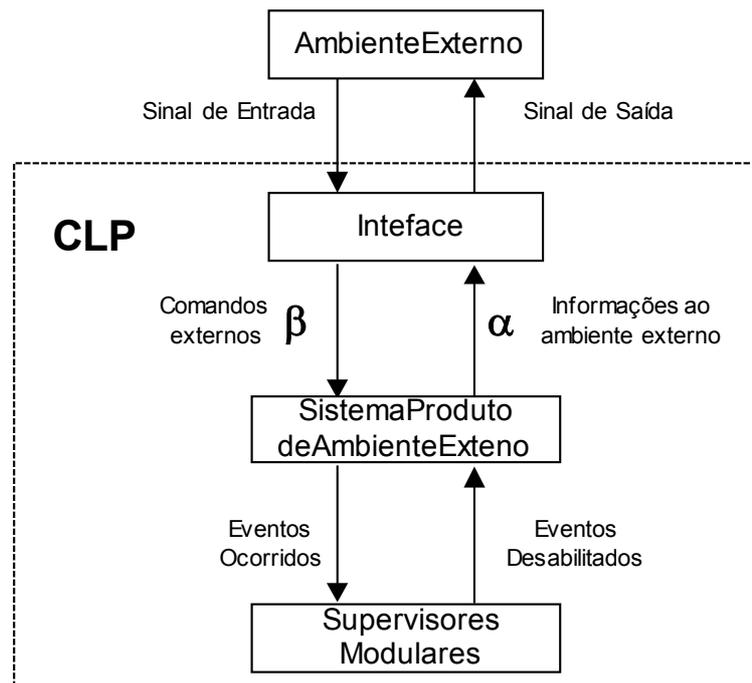


Figura 5.5 - Diagrama de blocos da estrutura de controle para a comunicação do sistema de informação e o ambiente externo.

A partir desta estrutura de controle para o ambiente externo pode-se realizar uma projeção do modelo estrutural e funcional que a contemple e que novamente deve ser semelhante ao modelo estrutural e funcional do sistema energético/material mostrado na Figura 5.4. A diferença básica está no sistema de atuação (SA) e sistema de medição (SM), para o caso do sistema energético/material, que não fazem sentido no caso de comunicação com o ambiente externo.

Conceitualmente, a comunicação entre um usuário e o sistema automático materializa-se na forma de comandos ou ordens de funcionamento, seguindo parâmetros também escolhidos pelo usuário, que podem ou não ser realizados em consequência da condição do sistema em poder ou não atender a estes comandos. Como o sistema só atende á ordens as quais está apto, e aqui se pode acrescentar também que o sistema só deve atender a comandos lícitos, em verdade trata-se então de um pedido, e não uma ordem, ao qual pode ou não estar associada uma resposta de possibilidade ou não de acolhimento.

Assim, pode-se caracterizar um novo modelo funcional e estrutural em que os sistemas de atuação (SA) e medição (SM) são substituídos por um sistema de pedido (SP), que pode ser implementado através de chaves seletoras, teclado, placa de rede, etc, e um sistema de resposta (SR), que pode ser desde uma simples lâmpada para sinalização como um LED em um painel, indo até um condicionador de sinal ou uma rede industrial de dados, por exemplo.

Na Figura 5.6 pode-se observar um exemplo para um painel de comando e um usuário como ambiente externo.

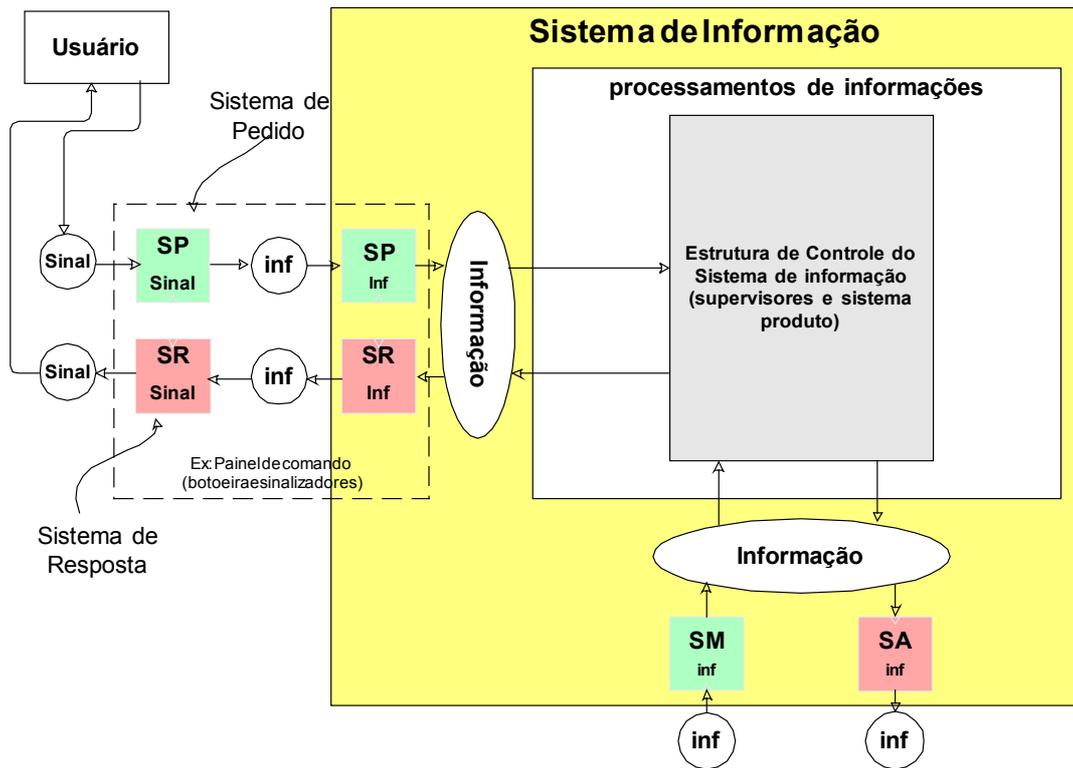


Figura 5.6 – Modelo estrutural e funcional do sistema de informação para o ambiente externo.

A Figura 5.6 mostra também uma pequena parte do modelo para o sistema energético/material (parte de processamento de informação dos sistemas de medição e atuação), lembrando que o sistema de informação em questão é o mesmo para este e o ambiente externo. De fato, o que se quer alcançar é um controle simultâneo que somente será efetivo através de uma única estrutura de controle e, deste modo, o modelo do sistema automático pode ser estendido como mostrado na Figura 5.7. Além disto, estudo da problemática envolvendo a comunicação permitiu um melhor entendimento do funcionamento de um sistema automático de modo que, o modelo original pôde ser incrementado de um sistema de atuação e medição (SAM), tornando explícito o fato de que existem sistemas exclusivamente de medição e exclusivamente de atuação, mas que na maioria dos casos o que surge é um sistema composto com a função de atuar mas que pode responder com os efeitos desta atuação.

Assim sendo, pode-se vislumbrar que, por uma questão de similaridade, a mesma estrutura deve ser observada quanto à comunicação e desta forma podem existir sistemas exclusivamente de pedido, sistemas exclusivamente de resposta e sistemas mistos de pedido e resposta (SPR). Deve-se lembrar também que o modelo exemplificado para o caso de um usuário pode ser utilizado para vários usuários ou mesmo modificado para o caso de comunicação com outros sistemas automáticos ou não automáticos, ou seja, qualquer enti-

dade pertencente ao ambiente externo que necessite trocar informações com o sistema automático em questão.

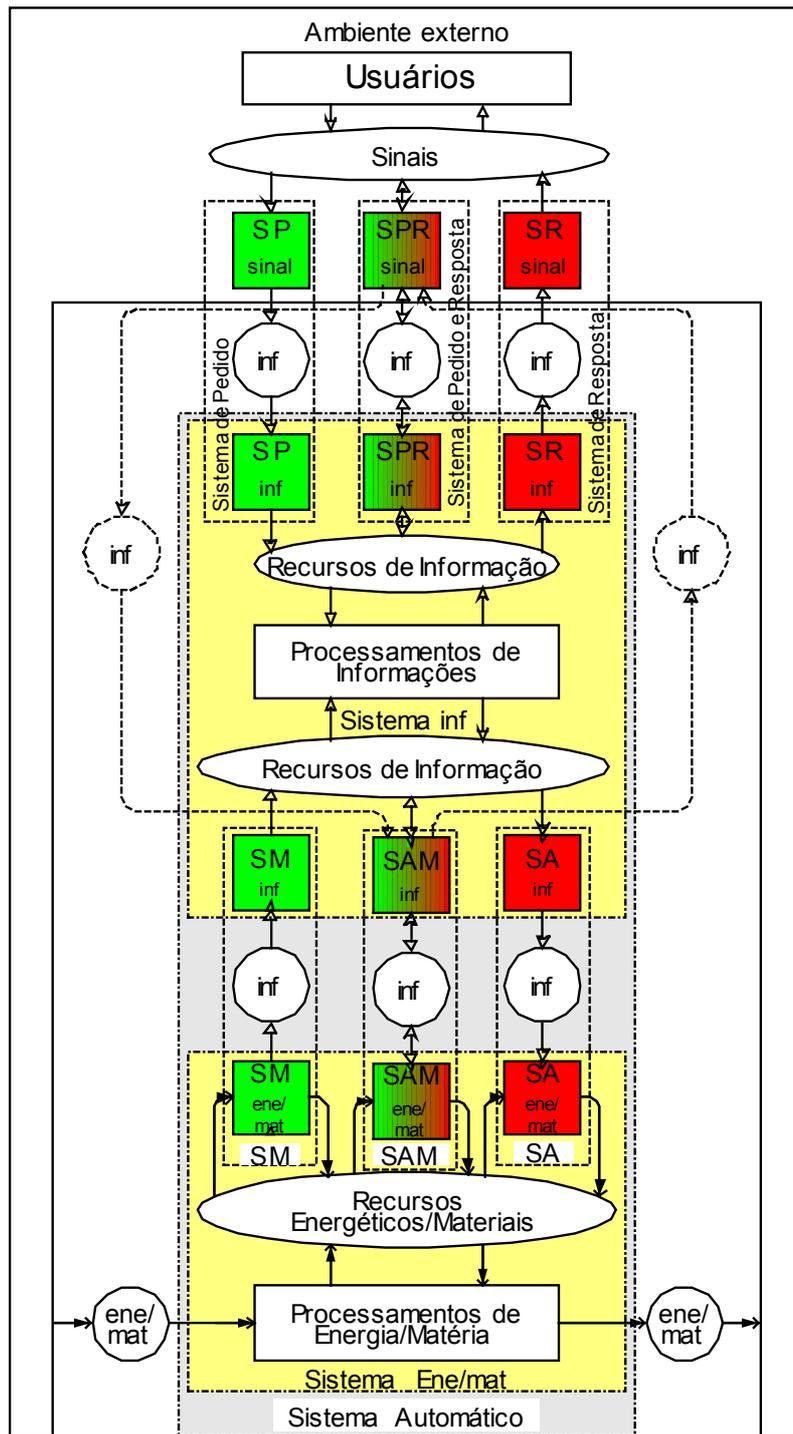


Figura 5.7 – Modelo estrutural e funcional de um sistema automático estendido – caso de um usuário.

Definido um modelo estrutural e funcional mais geral, estendido ao ambiente externo, pode-se esperar que esta mesma generalização aconteça com a estrutura de controle, que é o componente principal do sistema de informação. Observando-se as estruturas de controle mostradas anteriormente, Figura 3.11 e Figura 5.5, nota-se que ambas podem ser acopladas através do conjunto de supervisores modulares que, como já foi comentado, realizará

o controle das informações trocadas com o ambiente externo, do comportamento do sistema energético-material e das influências que ambos terão entre si. Isto quer dizer, por exemplo, que caso um usuário envie uma seqüência de comandos correta, o sistema energético-material deve realizá-la e, caso este último não tenha condições no momento, esta informação deve ser passada ao usuário.

A estrutura de controle estendida ao ambiente externo pode ser então realizada como mostra a Figura 5.8.

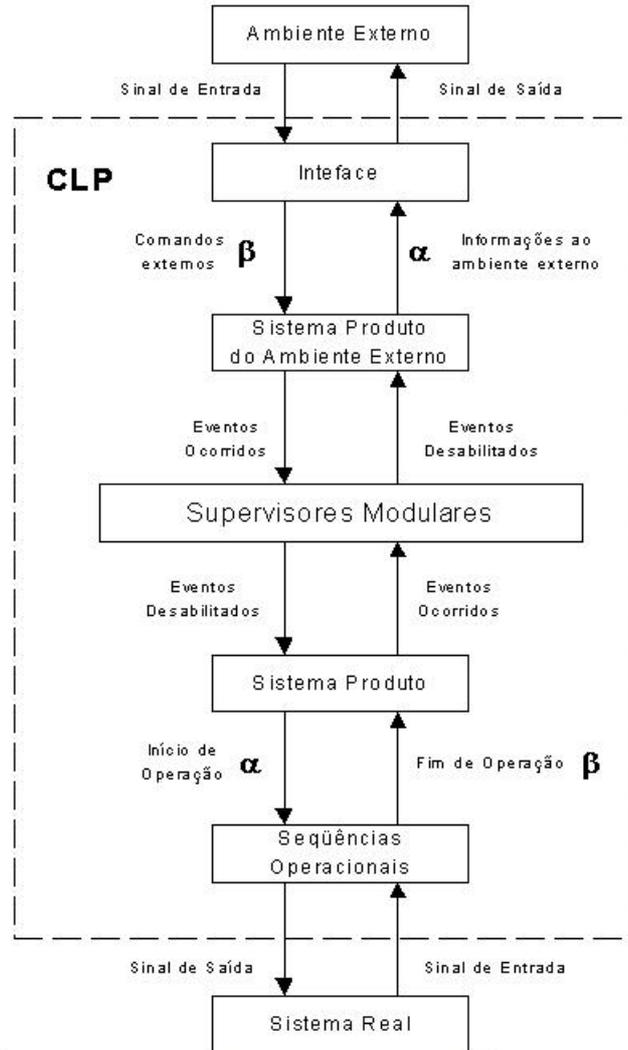


Figura 5.8 – Diagrama de blocos da estrutura de controle estendida ao ambiente externo.

Utilizando a notação em rede Canal/Agência para representar também a estrutura de controle, pode-se estender o modelo para o caso de dois usuários, onde o modelo do ambiente externo agora será o modelo do usuário e os sistemas de pedido e resposta fazem parte de painéis de comando. Esta modelagem é representada na Figura 5.9.

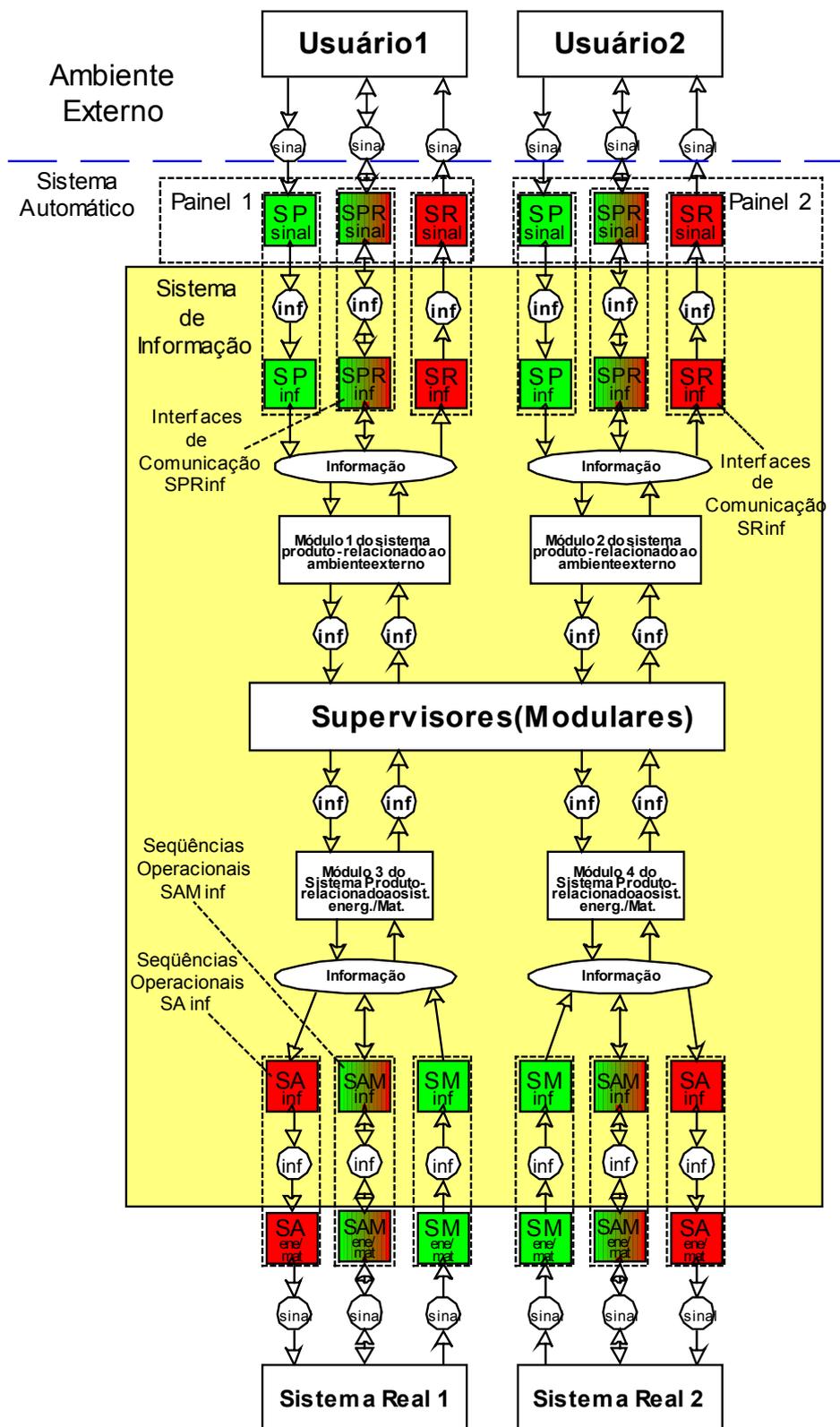


Figura 5.9 – Modelo estrutural e funcional do sistema de informação – Processamento de informações.

Modelar a interação com o ambiente externo pode não ser trivial e realmente não o é para o caso de um usuário, diferentemente de um sistema mecânico ou elétrico em que muitas vezes pode-se associar um modelo matemático, contínuo ou discreto. No entanto, deve-se lembrar que o controle que se quer realizar é em relação às informações trocadas com o

sistema automático e não em relação ao próprio usuário, desta maneira pode-se selecionar quais informações de entrada e saída são relevantes e quais devem ser realmente controladas, para assim começar a formular um modelo que tenha relação com estas informações.

As informações que o usuário troca com o sistema podem ser separadas em dois grandes grupos:

- Os pedidos feitos ao sistema, que informam a este como operar e quando deve iniciar ou terminar a operação – este grupo é definitivamente composto de eventos que não são controláveis já que não se pode limitar o desejo do usuário de fazer escolhas;
- Os sinais que respondem com os resultados das escolhas feitas pelo usuário e/ou os estados do sistema – este grupo é composto de eventos controláveis, e algumas vezes também por eventos não-controláveis, visto que são informações que o sistema monitora e escolhe passar ou não ao usuário.

O modelo de interação com o usuário relacionado ao grupo não-controlável pode ser descrito como sendo o modelo das intenções do usuário que, dependendo do sistema, pode ser decomposto em vários pequenos modelos de intenção de ligar, desligar, modificar, selecionar, interromper, alternar etc. As intenções do usuário podem ou não ser lícitas e, para tanto, ele pode cometer erros, fazer escolhas proibidas, tentar ligar ou desligar quando não pode, utilizar recursos indisponíveis e assim por diante, justificando a necessidade de controle de suas ações e a formulação de um segundo modelo que se relaciona com o grupo de informações controláveis, que é o modelo dos efeitos provocados pelo usuário.

Os dois modelos macro de interação com o usuário devem ter uma relação de causa e efeito e, da mesma maneira com que o modelo não-controlável pode ser decomposto, o controlável também o pode, sendo possível ter-se o modelo do usuário errando, alocando ou devolvendo um recurso, conseguindo realmente ligar ou desligar o sistema etc. De modo a manter a coerência da proposta, que identificou os sistemas de pedido e resposta, os modelos das intenções e efeitos podem então ser denominados de Modelo dos pedidos e Modelo das respostas ao ambiente externo.

Na Figura 5.10 pode-se observar como deve ser realizada a comunicação entre o supervisor, interface e o modelo do usuário, ou seja, com base nas mudanças de estado do modelo não-controlável e na informação do estado atual do modelo controlável o supervisor pode, em acordo às leis de controle estabelecidas, desabilitar os eventos adequados dos modelos controláveis.

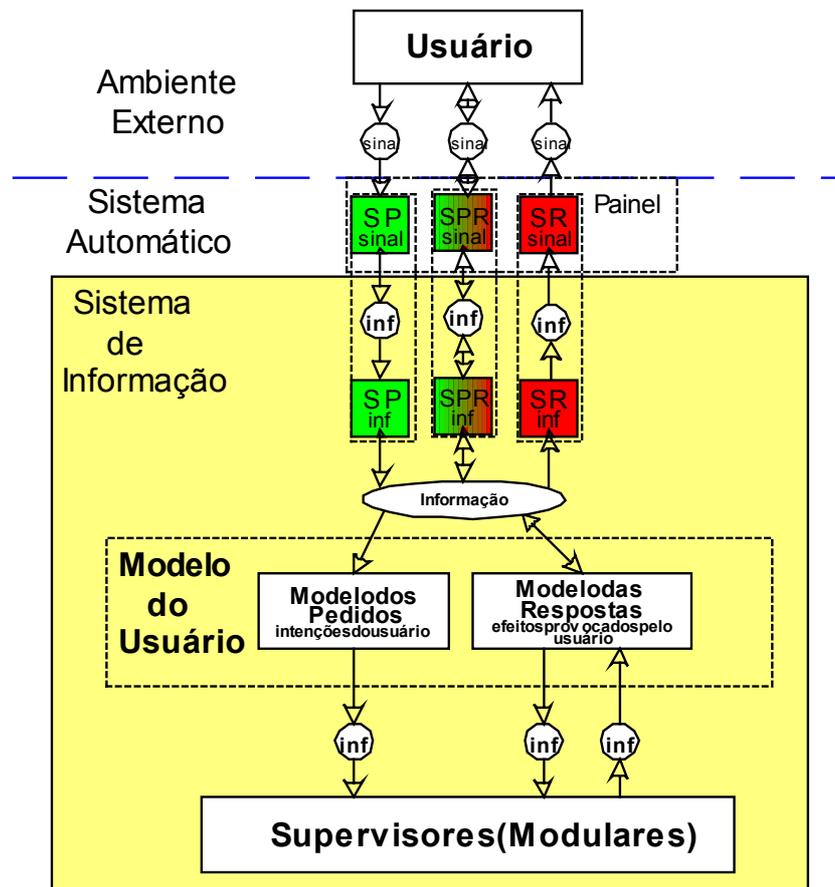


Figura 5.10 – Modelo macro do usuário.

5.3.1 Observações finais

Formulados o modelo estrutural e funcional e a estrutura de controle para a comunicação com o ambiente externo, o próximo passo é estabelecer o modelo comportamental dentro de uma seqüência sistematizada de projeto.

Em primeiro lugar, pode-se observar que modelar o comportamento do ambiente externo ao sistema automático quanto à comunicação entre eles introduz no projeto um conjunto de modelos e lógicas necessárias à interface, além de se fazer necessária a geração de especificações e, conseqüentemente, de supervisores modulares adicionais. Este fato acrescenta uma nova complexidade ao projeto e aos cálculos computacionais, que são uma limitação ao controle supervísório e, sendo assim, deve mostrar uma vantagem explícita para justificar este esforço adicional.

Como mencionado no item 5.2, a complexidade das informações trocadas deve ser avaliada de modo a destacar as de caráter crítico, quer dizer, complexas e muito importantes ao funcionamento do sistema. Em sendo esta complexidade tal que torne a abordagem tradicional difícil, pode ser justificável a utilização dos modelos que estão sendo propostos, no entanto parece não haver impedimento para que a abordagem seja de maneira combinada utilizando um modelo formal para a parte complexa e a abordagem habitual para a co-

municação mais simples. No modelo da Figura 5.7 esta possibilidade está representada pelos canais de informação tracejados entre o SAM de informação e o SPR de sinal.

Ressalta-se também que, para efeito de controle, a estrutura ilustrada na Figura 5.8 em nada difere da estrutura proposta por QUEIROZ et al. (2001) (Figura 3.11) no sentido que são compostas por interface, sistema produto e supervisores modulares e que, para efeito de cálculo dos supervisores modulares e implementação em CLP, são absolutamente idênticas. A representação da Figura 5.8 tem o objetivo de distinguir conceitualmente o projeto do sistema energético/material e o projeto da comunicação com o ambiente externo, proporcionando que se ataque os dois problemas separadamente.

Em contra-partida, o modelo funcional e estrutural da Figura 5.7 provoca a inserção de novos elementos físicos no sistema, relativos aos sistemas de pedido e resposta, elementos estes que existiam fisicamente na abordagem habitual mas que não apresentavam uma representação formal quanto a sua função e estrutura. Para o caso de comunicação simples a falta de representação formal não é grande empecilho, o que não é igualmente verdade para o entendimento e concepção em presença de aspectos intrincados.

5.4 Modelo comportamental para a comunicação com o ambiente externo

Com base nos modelos propostos no item 5.3 será apresentada uma sugestão de modelo comportamental em autômatos de estados finitos, devido principalmente aos estudos realizados sobre o problema real de projeto mencionado anteriormente. Esta proposta está amparada em alguns pressupostos básicos como o fato de que um usuário ou sistema externo pode fazer escolhas (pedidos) entre aspectos predeterminados que podem ou não ser mutuamente exclusivos. Além disto, a cada pedido pode ou não haver uma resposta relacionada, dependendo das especificações de projeto, bem como podem existir respostas sem nenhum pedido relacionado, como o caso de sinais de emergência vindos do sistema energético/material, ou seja, podem existir SP, SR e SPR como visto anteriormente.

Como se pode antever, a combinação de tipos de escolhas possíveis a um usuário é virtualmente infinita, fato pelo qual serão feitos exemplos de modelos para escolhas limitadas a 2 ou 3 opções concomitantes ou mutuamente exclusivas. Outro aspecto que será introduzido é a relação destes modelos com princípios de solução físicos de modo a guiar o projetista na escolha de seus modelos em função das tecnologias de que ele dispõe para concretizar o projeto.

De modo similar, também serão apresentadas as alternativas de modelos para as respostas ao ambiente externo que, ao contrário dos pedidos, podem ser limitadas a dois tipos básicos.

5.4.1 Modelos para os pedidos do ambiente externo

Levantar, qualitativa e quantitativamente, como um usuário ou sistema externo pode fazer pedidos a um sistema automático não é uma tarefa simples, em virtude desta informação variar muito dependendo das especificações de projeto e da interpretação que o projetista atribui a elas. Deste modo, é muito difícil imaginar um conjunto padronizado de modelos que seriam adequados a todo e qualquer tipo de configuração de pedidos e, por conseguinte, os que serão apresentados a seguir são exemplos de como se pode modelar os pedidos para alguns casos específicos na esperança de que sejam suficientes para orientar a modelagem para outros casos.

Faz-se então a proposta de modelo para o caso da escolha entre duas opções, primeiramente de forma independente e logo após para o caso de exclusão mútua entre as opções. Em seguida mostra-se a extensão para o caso de três opções.

O modelo da Figura 5.11 representa o pedido que um usuário pode fazer dentre duas opções disponíveis, em que podem ser escolhidas a opção 1, a 2 ou as duas ao mesmo tempo, bem como nenhuma delas. Note que o princípio de solução para este modelo depende da tecnologia utilizada para a interface com o ambiente externo e pode ser desde um programa em uma IHM até o caso de um painel de controle implementado através de botões tipo flip-flop ou duas chaves elétricas comuns (NBR, 1973), como o exemplo mostrado nesta figura.

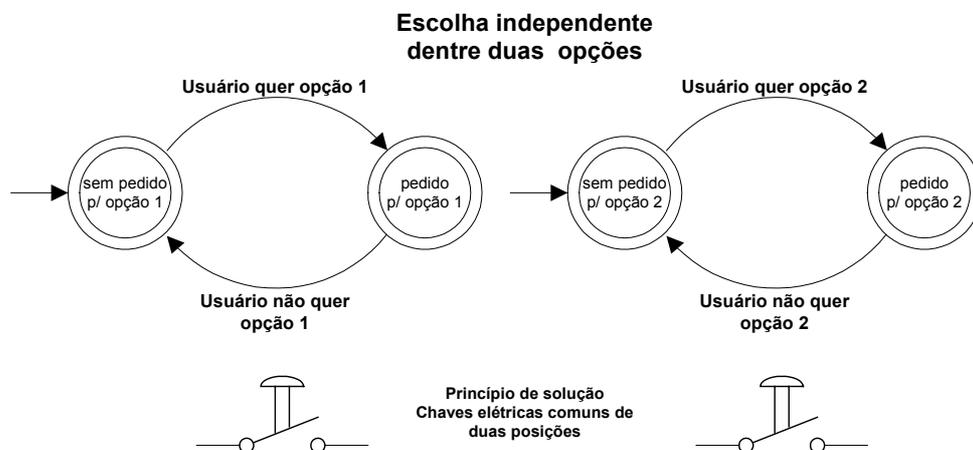


Figura 5.11 - Modelo comportamental para o pedido - duas opções independentes.

Para o caso de duas opções mutuamente exclusivas tem-se o modelo representado na Figura 5.12 onde novamente os princípios de solução dependem da tecnologia utilizada. No caso de um painel de controle o pedido pode ser manifestado através de uma chave elétrica de três posições com desligamento no meio (NBR, 1973).

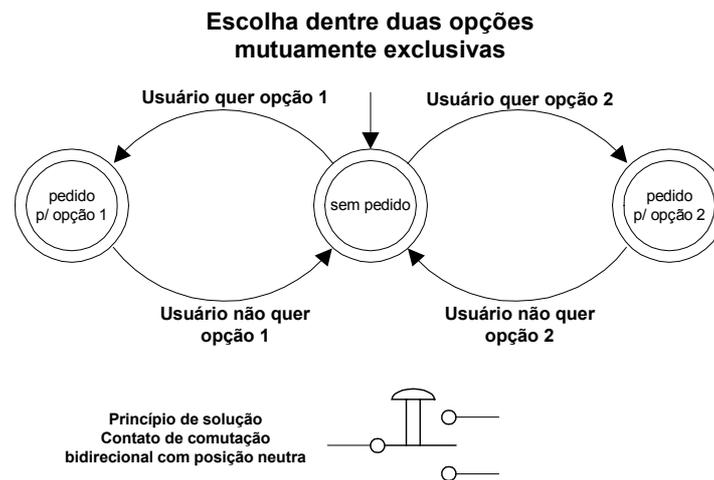


Figura 5.12 - Modelo comportamental para o pedido - duas opções mutuamente exclusivas.

Novamente os princípios de solução dependem da tecnologia utilizada mas no caso do painel de controle também pode ser resolvido através de uma chave elétrica de três posições com desligamento no meio (NBR 5274), como mostrado na Figura 5.12.

De maneira bastante direta pode-se estender o exemplo dado para o caso de três opções conforme representado na Figura 5.13 e na Figura 5.14.

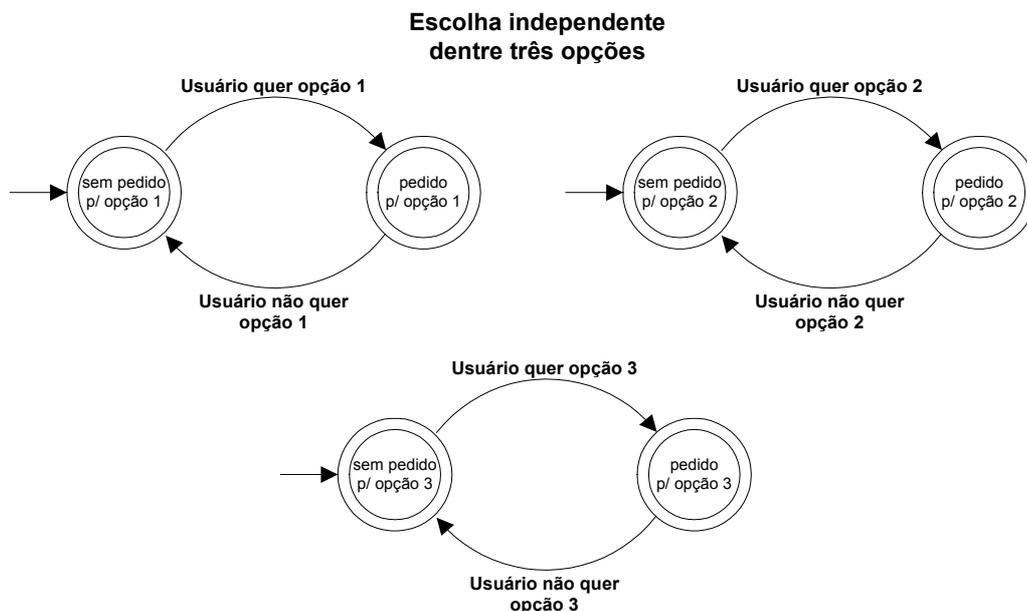


Figura 5.13 - Modelo para o pedido - três opções independentes.

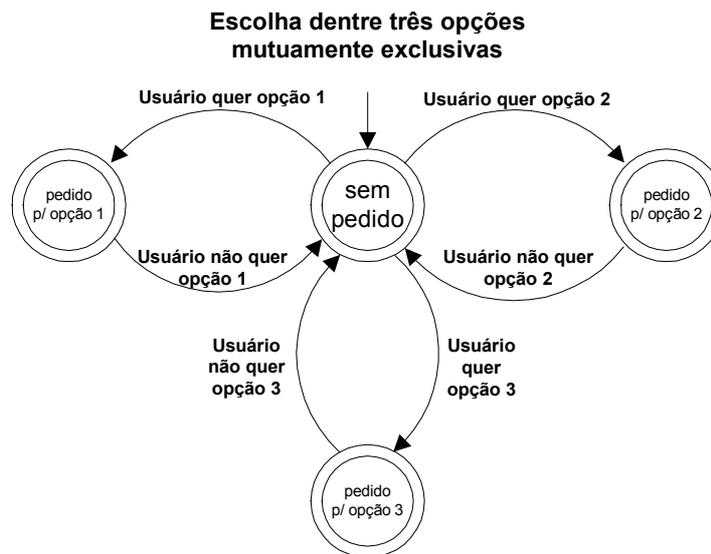


Figura 5.14 - Modelo para o pedido - três opções mutuamente exclusivas.

Os princípios de solução podem ser escolhidos de maneira similar ao caso de duas escolhas, guardadas as proporções de limitação dos componentes comerciais, como por exemplo para o caso da Figura 5.14 em que uma chave elétrica de três posições, onde o acionamento de uma posição não implica em passar por nenhuma outra, não é um componente exatamente comum. De qualquer forma, a não existência de um componente comercial não inviabiliza a utilização do modelo pois se pode criar um circuito elétrico ou eletrônico que siga este comportamento caracterizando mais um tipo de hardware possível para o sistema de pedido. O mesmo vale para o caso de mais de 3 opções e aqui cabe dizer que estes modelos de duas e três opções podem não ser únicos sendo possíveis outras formas de configurar os pedidos.

Deve-se lembrar também que se tratam de modelos cujos eventos são em sua totalidade não-controláveis. De fato, estes modelos são a rigor uma abstração dos sistemas de pedido (SP), compostos por partes de hardware e software, responsáveis por refletir a interação com o ambiente externo (por exemplo, com um usuário) quando alterados pelo fato dele querer ou não fazer certas escolhas, o que definitivamente não pode ser controlado pelo sistema automático em questão.

5.4.2 Modelos para as respostas ao ambiente externo

Ao contrário dos pedidos feitos pelo ambiente externo, as respostas dadas a ele são de caráter controlável e consideradas aqui mais restritas quanto à forma mas não quanto ao conteúdo. Propõe-se então considerar as respostas como sendo de dois tipos básicos, para efeito de simplificação e em uma tentativa de formular um padrão para o que se acredita, neste trabalho, ser representativo para a maioria dos casos – as respostas relacionadas aos pedidos e as não relacionadas aos pedidos.

5.4.2.1 Respostas relacionadas aos pedidos

As respostas são essencialmente sinais enviados ao ambiente externo sobre os quais se deseja passar uma mensagem (informação) ou conjunto delas. Uma resposta relacionada a um pedido pode se apresentar de duas maneiras: (1) quando é importante passar mensagens distintas para os casos de não haver pedido, existir pedido que não pode ser atendido e existir pedido que pode ser atendido; (2) quando o importante é somente informar se está tudo certo ou não.

A Figura 5.15 refere-se ao caso (1) onde há três mensagens distintas. A cada estado deste modelo pode ser relacionada uma mensagem distinta contemplando a não existência de pedido ou se o pedido pode ou não ser atendido. Note que este modelo é uma abstração do sistema de pedido e resposta (SPR), visto que existe um evento relacionado ao pedido (“Usuário não quer mais o pedido”).

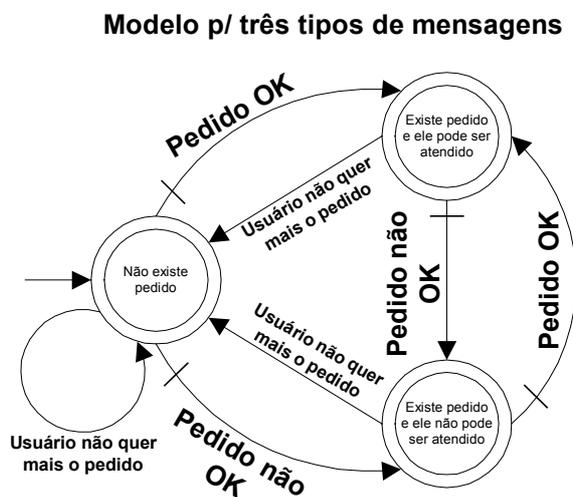


Figura 5.15 - Modelo da respostas ao ambiente externo relacionadas aos pedidos - três mensagens distintas.

Quando não é importante distinguir a não existência de pedido do fato de existir o pedido e ele estar apto a ser atendido, o modelo pode então ser reduzido a dois estados que basicamente passam a informação de se existe uma não-conformidade (erro) ou se tudo esta correto. Este caso é também um exemplo de SPR e está ilustrado na Figura 5.16 em que pode ser visto o evento relacionado a vontade do usuário de não mais querer o pedido.

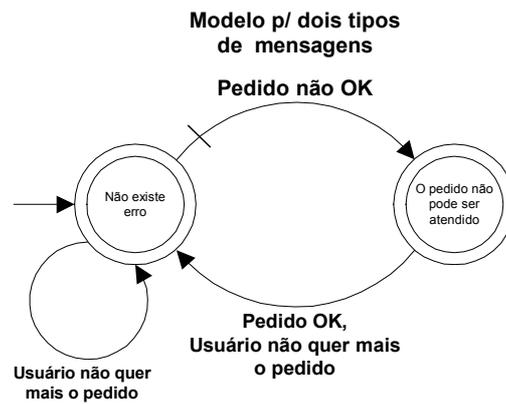


Figura 5.16 - Modelo da respostas ao ambiente externo relacionadas aos pedidos - duas mensagens distintas.

5.4.2.2 Respostas não relacionadas aos pedidos

As respostas não relacionadas aos pedidos são uma abstração dos sistemas de resposta (SR) e têm relação com as informações que se quer passar quanto aos estados do sistema automático como um todo, especialmente quanto às condições do sistema energético/material. Informações relativas ao mau funcionamento, quebras de ferramentas, término de operação, término de recursos (peças por exemplo) e etc, não são diretamente relacionadas a um pedido do ambiente externo e podem surgir à revelia deste.

Pode-se então relacionar mensagens a cada tipo de erro ou informação relevante do estado do sistema, que se quer passar ao ambiente externo, e associá-las a modelos relativos à ocorrência destes fatos. Mais uma vez é difícil estabelecer um padrão devido à variedade de configurações que os sistemas podem ter mas, em princípio, modelos simples de dois ou três estados podem resolver a maioria dos problemas. Como exemplos, pode-se imaginar um modelo de dois estados (em erro e OK) associado a cada tipo de erro ou um modelo de três ou quatro estados associado ao status de funcionamento do sistema (parado, iniciando, operando, terminando), tendo-se em ambos os casos como justificativa a informação relacionada que se quer passar (Figura 5.17).

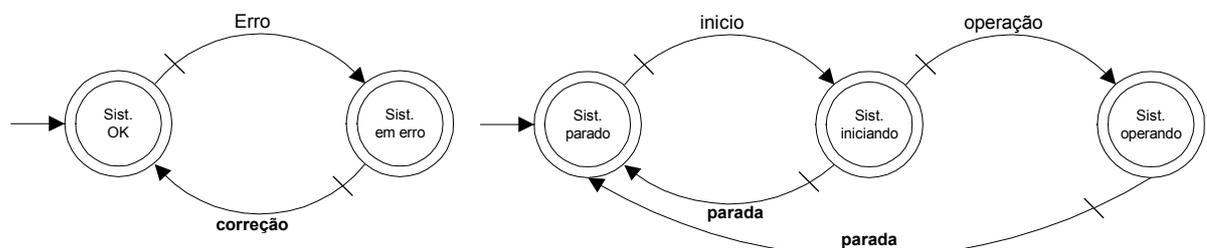


Figura 5.17 - exemplos de modelos de respostas não relacionadas aos pedidos.

5.4.3 Observações finais

Modelo do ambiente externo.

Observa-se que, embora usuários e sistemas externos sejam parte do ambiente externo, o seu tratamento pelo sistema de informações de um sistema automático é muito similar ao que acontece com os sistemas energético/materiais que são partes do ambiente interno. Os módulos do sistema produto dos sistemas de atuação e medição, concebidos para modelar atuadores e sensores elétricos, hidráulicos, pneumáticos etc., refletem as mudanças de estado ocorridas nos sistemas físicos quando as informações fluem para dentro do sistema de informação e, por sua vez, provocam alterações nos estados dos sistemas físicos a medida que comandos saem do sistema de informação em direção aos atuadores. Desta forma, pode-se dizer que os módulos do sistema produto, que são uma abstração dos SA, SM e SAM, refletem os estados do sistema físico a qual estão relacionados, hora seguindo as alterações espontâneas deste, hora forçando sua alteração.

O ambiente externo, tanto na figura de operadores humanos como máquinas adjacentes, também influencia e é influenciado pelos modelos do sistema de informação relacionados a ele. De modo análogo aos sistemas de atuação e medição, os sistemas de pedido e resposta são uma abstração de hardwares e softwares de comunicação baseados em diversas tecnologias, e refletem as mudanças de estado ocorridas no ambiente externo, físico, que têm relação com o sistema automático, quando as informações fluem para dentro do sistema de informação e, por sua vez, provocam alterações nos estados do sistema externo à medida que mensagens são enviadas através dos meios de divulgação associados, como uma tela de IHM por exemplo.

Para exemplificar esta relação pode-se supor um usuário que em princípio nada quer e muda de estado querendo utilizar uma função do sistema {estado 'não quero nada' → estado 'quero algo'}. Caso a máquina responda que o pedido vai ser atendido, o usuário passará do estado em que não está fazendo nada pra o estado em que está utilizando a função que deseja {estado 'usuário parado' → estado 'usuário utilizando função'} e, em caso contrário, ele passará para um estado em que fica esperando a máquina responder outra vez ou tenta uma outra função {estado 'usuário parado' → estado 'usuário aguardando'}.

Este exemplo é um tanto lúdico mas serve bem para representar como uma resposta de um sistema pode alterar o estado comportamental de um ser humano ou um outro sistema, nem que seja o humor do primeiro. Assim, pode-se dizer que, a exemplo dos SA, SM e SAM, os módulos do sistema produto do ambiente externo, que são uma abstração do SP,SR e SPR, refletem os estados do ambiente externo a que estão relacionados, hora seguindo as suas alterações espontâneas, hora provocando sua alteração.

Princípios de solução.

Um princípio de solução para os modelos apresentados nem sempre tem relação direta com componentes físicos, como o exemplo dos interruptores, e às vezes sua implemen-

tação tem mais afinidade com linhas de código em programas ou com uma construção mista. A arquitetura atual dos computadores pessoais tem muitos exemplos de soluções em comunicação resolvidas utilizando recursos de hardware, software ou uma combinação deles. Como exemplo, pode ser citado o periférico denominado modem (modulador e demodulador) utilizado para conexão doméstica à Internet via linha telefônica, que pode ter suas funcionalidades puramente implementadas por hardware ou pode utilizar um programa e o processador do PC para isto.

Em sistemas automáticos, a utilização de uma IHM já pode ser vista como uma implementação mista de solução em que um projetista pode programar seu hardware para que a entrada e saída e dados seja realizada em concordância com os modelos de pedido e resposta ao ambiente externo. Um circuito eletrônico, analógico ou digital, também pode ser configurado para que o fluxo de informações ocorra como previsto nos modelos, como já foi sugerido anteriormente.

Cálculo do controle e implementação.

Como será demonstrado nos capítulos seguintes, a modelagem apresentada não impõe dificuldades para a utilização dos métodos de controle supervisorio, em especial ao controle supervisorio modular local que é o método utilizado neste trabalho. A técnica de implementação em CLP, porém, sofre algumas modificações para se adequar às características da modelagem proposta.

Algumas ferramentas de software podem ser utilizadas tanto para o cálculo dos supervisores e sistema produto, como para auxiliar na implementação em CLP. Algumas destas ferramentas serão apresentadas aqui por terem sido utilizadas neste trabalho, sendo que uma em especial foi desenvolvida em virtude de um outro trabalho (SILVA NETO, 2004), realizado em parceria com o presente trabalho, para a solução de implementação de controle supervisorio modular local para sistemas automáticos com modelos de comunicação com o ambiente externo.

CAPÍTULO 6

ASPECTOS DE IMPLEMENTAÇÃO EM CLP

Como forma de testar a modelagem proposta e compará-la a outro método de projeto, foi realizada uma parceria em que o trabalho realizado por SILVA NETO (2003) forneceu subsídios importantes para que fosse atingida esta meta. O trabalho realizado por SILVA NETO resultou no projeto e implementação do controle da unidade de potência hidráulica estudada, utilizando álgebra booleana, tabelas-verdade, mapas de Karnaugh-Veitch e algoritmos numéricos para o cálculo das equações booleanas implementadas em CLP. Um simulador foi construído (Figura 6.1), utilizando o CLP original do projeto da unidade de potência, onde foi possível simular os painéis de comando, circuito hidráulico e testar o controle calculado.

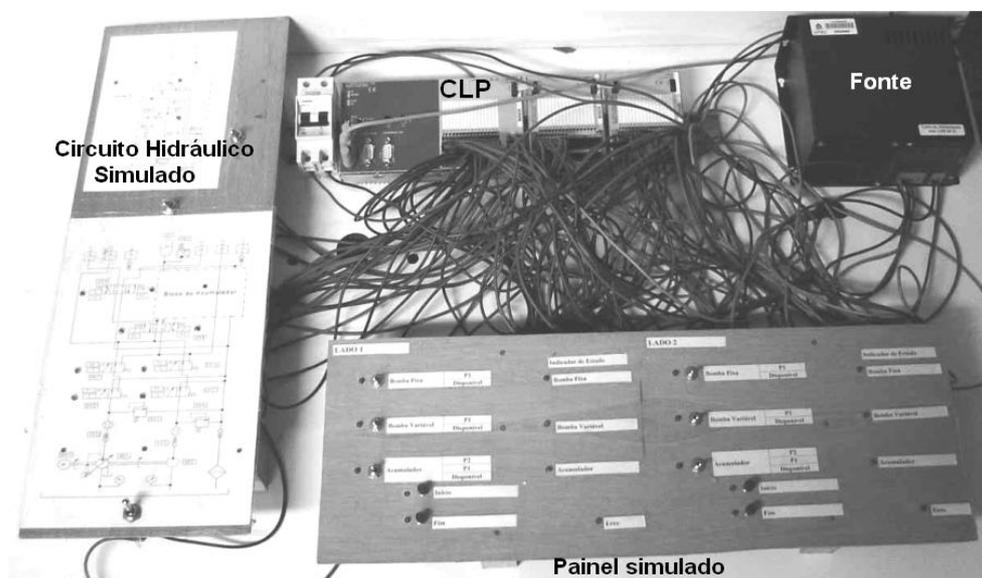


Figura 6.1 - Bancada de simulação do controle de uma unidade de potência hidráulica (adaptada de SILVA NETO, 2004).

A parceria permitiu a elaboração de uma estrutura de implementação que fosse adequada aos modelos introduzidos no Capítulo 5 e também do trabalho de comparação entre os métodos de álgebra booleana e controle supervisorio modular local (SILVA NETO et al. 2004) que será abordado na conclusão deste texto.

6.1 Programação em CLP

Devido à extensa utilização do CLP nas aplicações industriais, uma grande gama de padrões de projeto e programação foi criada para ele, resultado de pesquisas em vários campos de aplicação baseados na sua versatilidade de utilização. A evolução destes padrões pode ser vista na Figura 6.2 em que destacam-se o Grafcet, IEC 848 (IEC, 1988) e IEC 1131-3 (IEC, 1992) como as mais significativas.

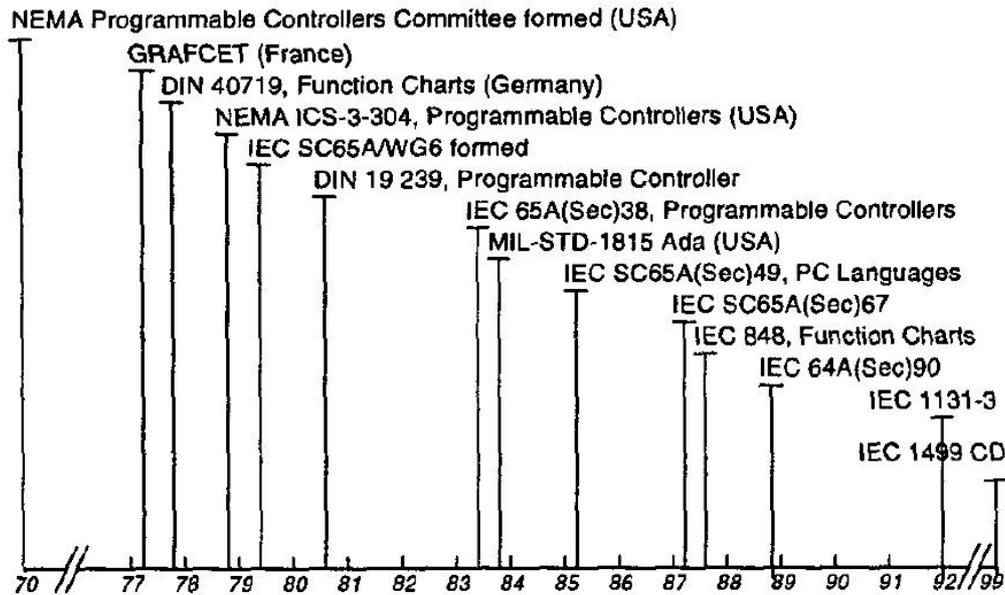


Figura 6.2 - Padronização da programação de CLP (FREY e LITZ, 2000).

Segundo FREY e LITZ (2000), o padrão mais influente seria o IEC 1131 que descreve as linguagens *IL (Instruction List)*, *SFC (Sequential Function Chart)*, *LD (Ladder Diagram)*, *FBD (Function Block Diagram)* e *ST (Structured text)*, em que pode-se ainda destacar o LD (diagrama-escada, no português) como sendo a linguagem mais abrangente no âmbito industrial. Em aspectos gerais, FREY e LITZ (1998, 2000) ilustram o processo de projeto de controle lógico em CLP como mostrado na Figura 6.3.

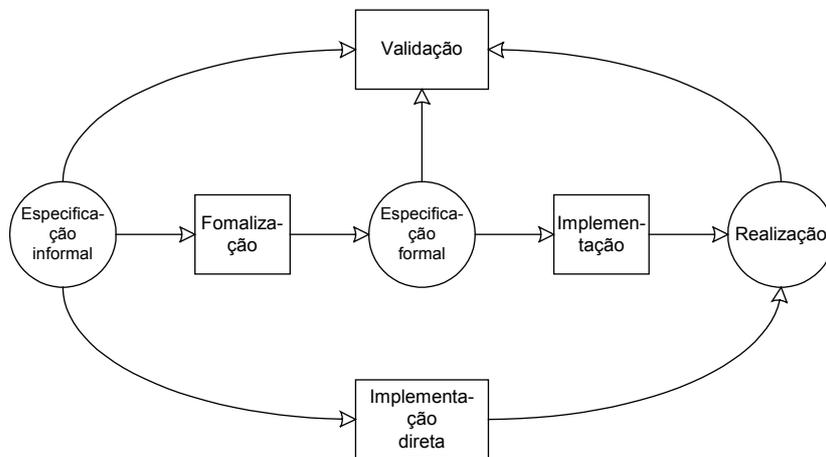


Figura 6.3 - Canal/Agência do processo de projeto de controle lógico em CLP (adaptado de FREY e LITZ, 1998, 2000).

Embora seja possível, e existam, implementações da forma direta e que são validadas utilizando especificações informais (arco inferior e superior do gráfico da Figura 6.3) acredita-se que para os projetos mais elaborados e complexos deve-se passar por uma formalização das especificações e validação formal (arcos centrais e superior direito da Figura 6.3). Neste sentido, MADER e WUPPER (2000), esboçam um esquema para aplicações em CLP (Figura 6.4) como forma de motivar a discussão sobre os quais são os diferentes níveis de abstração que uma aplicação pode ser tratada e quais são suas interconexões.

de abstração que uma aplicação pode ser tratada e quais são suas interconexões. MADER e WUPPER explicitam que a formalização das especificações é o primeiro passo para se utilizar métodos e modelos de projeto adequados e que mesmo utilizando-se aspectos formais não se deve perder a simplicidade sempre que ela for possível.

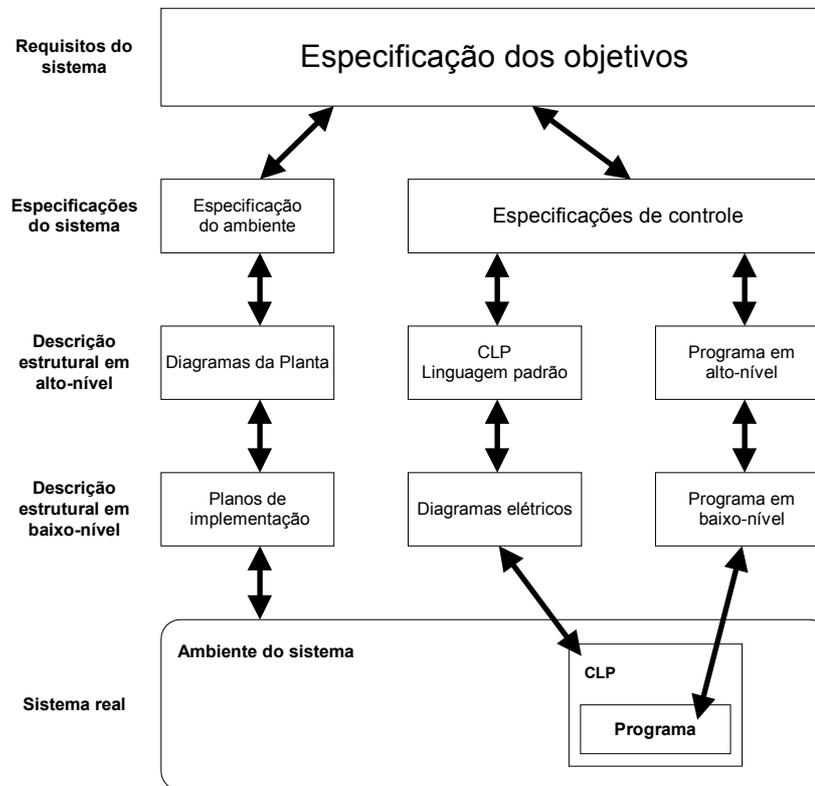


Figura 6.4 - Níveis de abstração de aplicações em CLP (MADER e WUPPER, 2000. tradução nossa).

Com respeito às linguagens de programação, MADER (2000) lembra que a estrutura de um programa pode ser projetada e verificada de maneira fragmentada e que, desta forma, não é incomum a utilização de diferentes linguagens de programação em um mesmo projeto (programa). Esta afirmação pode ser ratificada observando-se os atuais ambientes de programação em CLP que permitem a utilização simultânea de linguagens diferentes.

Assim, as implementações que serão demonstradas neste trabalho levam em consideração formalismos de projeto estrutural, funcional e comportamental, além de adotar concomitantemente linguagens de programação largamente utilizadas para representar a implementação em programa dos modelos comportamentais do sistema.

6.2 Métodos e ferramentas de cálculo do controle supervísório

A modelagem e síntese dos supervisores serão realizadas segundo a teoria de controle supervísório modular local (TCSML) (QUEIROZ e CURY, 2000), que utiliza, dentre outras coisas, uma representação da planta por sistema produto (Capítulo 3). Os modelos são então criados no formato do software livre GRAIL (UWO, 2002), que permite a modelagem e

operações simbólicas sobre os modelos facilitando o trabalho de projeto e o cálculo dos supervisores, uma vez que os arquivos são criados em código ASCII utilizando um editor de texto comum em ambiente MS-DOS. Exemplos de arquivos em formato Grail e operações realizadas com esta ferramenta estão no Apêndice 1.

Embora a TCSML tenha como objetivo principal evitar a explosão de estados, característica da teoria de controle supervísório clássica, o cálculo simbólico realizado com a ferramenta Grail exige muito recurso e tempo computacional. Para contornar este problema utiliza-se neste trabalho uma segunda ferramenta livre para modelagem e operações em SED, denominada CTCT, cuja modelagem e cálculo é realizada de forma numérica e não simbólica, diminuindo consideravelmente o esforço computacional. O CTCT, no entanto, não é tão agradável à modelagem como o Grail e torna os resultados de difícil depuração além do fato que, embora ambos sejam baseados em MS-DOS, trabalhar com o Grail é um pouco mais amigável na opinião deste autor.

Desta forma, a modelagem e cálculos básicos são realizadas em formato Grail e os cálculos mais pesados são realizados com a ferramenta CTCT. Entretanto, os formatos dos modelos nas duas ferramentas citadas não são compatíveis e para que seja possível a utilização em conjunto delas, uma terceira ferramenta livre, agora para conversão entre Grail e CTCT, foi utilizada. Trata-se de um pacote de várias funções para MS-DOS criadas no Departamento de Automação e Sistemas (DAS) da UFSC que torna possível a conversão do formato Grail para CTCT e CTCT para Grail, além de alguns outros formatos não abordados aqui. Exemplos de utilização do CTCT e do pacote de conversão estão no Apêndice 2.

Tendo calculado os supervisores modulares e o sistema produto a ser implementado, o projetista pode realizar a programação do CLP, em diagrama escada, lista de instruções (LI), blocos lógicos ou SFC (Grafcet), dependendo do método disponível pelo modelo de CLP escolhido. Para esta operação será sugerida uma estrutura de implementação, no item seguinte, adequada à TCSML e aos modelos de comunicação vistos anteriormente. Esta atividade pode ser bastante trabalhosa e passível de muitos erros quando o sistema projetado é razoavelmente grande e, neste caso, uma ferramenta de auxílio é também adequada. Assim, este trabalho utiliza uma quarta ferramenta de apoio responsável por converter os arquivos em formato Grail para lista de instruções do CLP adequado, denominada CGLI (conversor de Grail para lista de instruções) (SILVA NETO, 2004). O CGLI converte os arquivos de supervisores e módulos do sistema produto em lista de instruções seguindo o formato de estrutura de controle e as particularidades da estrutura de implementação de um sistema com modelos para a comunicação, deixando a cargo do projetista a programação de seqüências operacionais e interfaces de entrada e saída.

No Apêndice 3 pode ser visto um exemplo de utilização do CGLI para um CLP específico.

6.3 Estrutura de implementação de autômatos em diagrama escada

A estrutura apresentada a seguir é fortemente baseada na apresentada por QUEIROZ et al. (2001, 2002), VIEIRA (2003) e QUEIROZ (2004), que se entende não ser a única, mas que atende de maneira eficiente às necessidades do presente trabalho, que como não tem o principal objetivo de estudar as estruturas adequadas de implementação, acabou por não se preocupar com outras abordagens.

O programa, então, deve ser implementado de forma contínua, sendo executadas todas as linhas de código a cada ciclo do CLP e em concordância com a estrutura de programa apresentada na Figura 6.5:

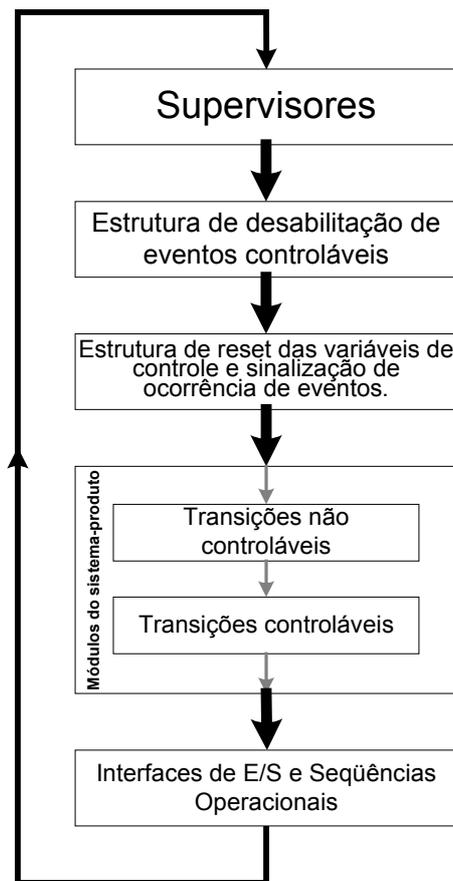


Figura 6.5 - Estrutura do programa em CLP para o controle supervísório proposto.

Observando a ordem de implementação da Figura 6.5 e tendo em mente a estrutura de controle ilustrada na Figura 3.11, pode-se descrever o ciclo de execução como:

1. Os supervisores, seguindo a lei de controle estabelecida, sinalizam quais os eventos controláveis, proibidos em relação ao estado atual do sistema produto, devem ser desabilitados através da estrutura de desabilitação de eventos controláveis;
2. Na estrutura de desabilitação de eventos controláveis, os eventos proibidos pelos supervisores são desabilitados utilizando-se variáveis lógicas binárias

que são colocadas em nível lógico alto, sinalizando a proibição da ocorrência dos eventos associados;

3. As variáveis relativas ao evento ocorrido no ciclo anterior e ao controle da execução da estrutura do programa são colocadas em nível lógico baixo sinalizando que uma nova transição pode ocorrer;
4. Um dos eventos passíveis de ocorrer no ciclo atual provoca a evolução do sistema-produto sendo que um evento não-controlável sempre tenha preferência a um evento controlável. Para assegurar isto, as linhas de código relativas a transições não-controláveis são implementadas primeiro e em seguida vem as linhas relativas as transições controláveis. Quanto ocorre uma transição, o sistema-produto imediatamente bloqueia uma nova transição e sinaliza qual evento ocorreu através de variáveis adequadas (nível lógico alto) que serão colocadas em nível lógico baixo na estrutura de *reset*, logo após a atualização do supervisor;
5. Finalmente, são executadas as interfaces de E/S e as seqüências operacionais, responsáveis por interpretar ou gerar informações de E/S e executar os detalhes operacionais dos módulos do sistema-produto, observando os eventos ocorridos e sinalizados pelo sistema-produto e os sinais vindos do ambiente externo. E o ciclo recomeça.

Nos itens seguintes, serão detalhados os diagramas escada (*Ladder Diagram*) ou SFC relativos aos blocos de programa da Figura 6.5.

6.4 Supervisores

A implementação, em diagrama escada, dos autômatos relativos aos supervisores modulares é realizada de maneira direta através de uma linha lógica para cada estado de destino existente, no formato ilustrado na Figura 6.6.

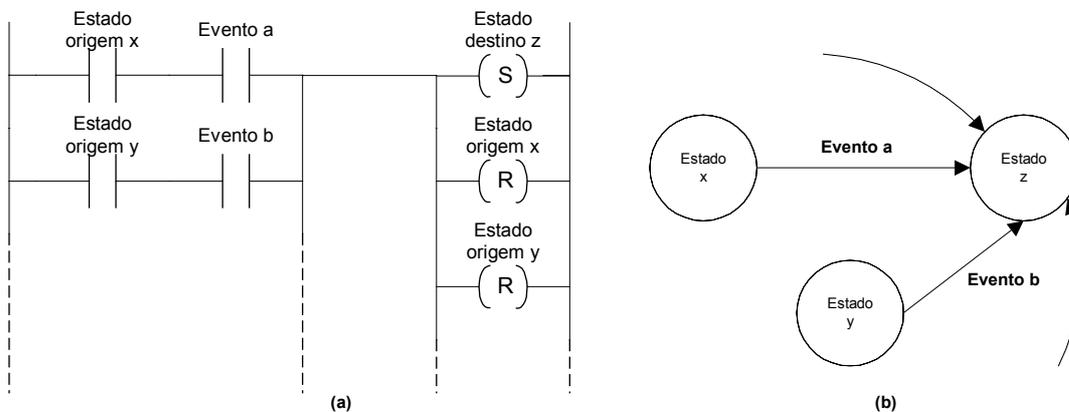


Figura 6.6 – (a) Diagrama escada genérico para implementação de supervisores modulares originalmente em (b) autômatos.

A lógica em diagrama-escada tem correspondência direta em blocos lógicos, LI ou SFC, que são as linguagens de programação mais comuns em CLP sendo, portanto, facilmente traduzida de uma linguagem para outra como já fazem alguns ambientes de programação para CLP.

6.5 Estrutura de desabilitação de eventos controláveis

A cada estado dos supervisores pode haver um conjunto de eventos controláveis que devem ser desabilitados e estes eventos, por sua vez, podem ser desabilitados por mais de um supervisor. Isto significa que um evento, controlável e passível de ocorrer em certo instante, só ocorrerá quando não existir nenhum estado de qualquer dos supervisores sinalizando sua desabilitação. Esta condição é implementada utilizando variáveis de sinalização de desabilitação para cada evento controlável (D_evento) e a estrutura em diagrama escada mostrada a seguir.

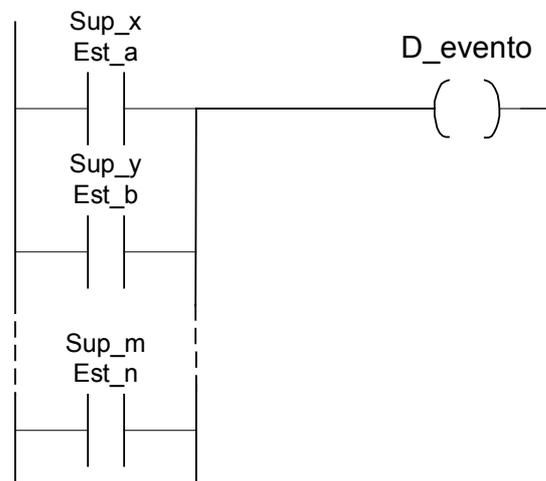


Figura 6.7 - Diagrama escada para a estrutura de desabilitação.

6.6 Estrutura de *reset* das variáveis de controle e variáveis de sinalização de ocorrência de eventos

As variáveis que sinalizam a ocorrência de eventos são de valores atribuídos com retenção, como será mostrado mais adiante, e para que tenham um caráter próximo ao instantâneo devem ser colocadas em nível lógico baixo após serem observadas pelos supervisores, não permitindo que permaneçam ativas por mais de um ciclo de varredura do programa. A estrutura de *reset* realiza esta tarefa e também habilita uma nova evolução do sistema-produto impedida por uma variável de intertravamento que impede a ocorrência de mais de uma evolução do sistema-produto sem a conseqüente atualização dos supervisores.

A cada variável relatada acima tem-se então uma operação lógica a exemplo do diagrama mostrado na Figura 6.8.



Figura 6.8 - Diagrama escada para a estrutura de *reset*.

6.7 Sistema produto

Os módulos de sistema-produto são compostos de transições controláveis e/ou não-controláveis e, para evitar inconsistência de controle as transições não-controláveis, são separadas das controláveis e implementadas de modo que as operações lógicas referentes a elas sempre sejam executadas antes das referentes às transições controláveis. Isto é realizado devido à lógica de implementação de maneira direta para os eventos não-controláveis e indireta para os controláveis como será explicado nos itens seguintes.

6.7.1 Transições não-controláveis

A lógica referente às transições não-controláveis executa a transição de maneira direta quando recebe um sinal da seqüência operacional (SO) (ou da interface de entrada) referente a um evento vindo do ambiente externo do qual não se tem controle sobre sua ocorrência. O sinal aparece sob a forma de uma variável de entrada vinculada a um contato normalmente aberto como visto na ilustração da Figura 6.9.

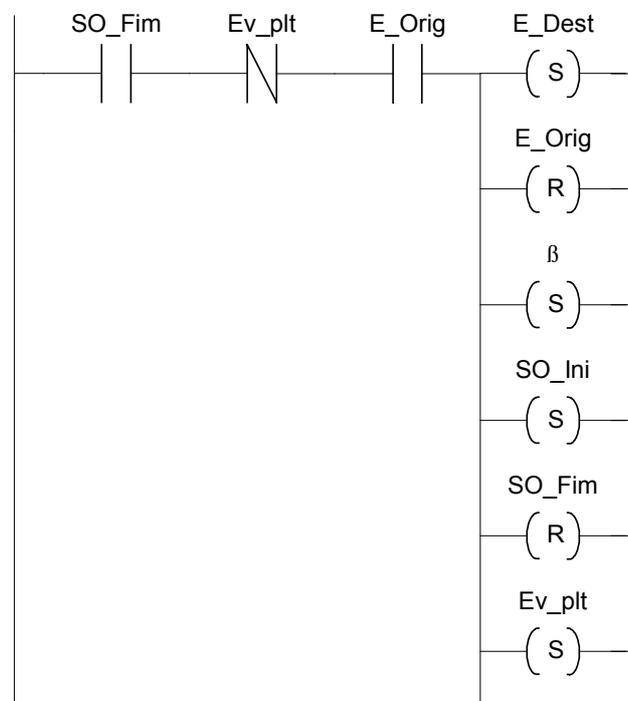


Figura 6.9 – Diagrama-escada para implementação de transição não-controlável entre dois estados diferentes.

A seqüência operacional relacionada sinaliza que terminou ($SO_fim = 1$) e caso a planta não esteja impedida de evoluir ($Ev_plt=0$), ocorrerá a evolução do estado de origem ($E_Orig=0$) para o de destino ($E_Dest=1$). A seqüência operacional é preparada para nova operação ($SO_fim=0, SO_ini=1$) ou para receber estímulos do ambiente externo, o evento não controlável relacionado é sinalizado ($\beta=1$) e a planta é impedida de nova evolução até que o supervisor seja atualizado marcando a variável de intertravamento ($Ev_plt=1$).

As seqüências operacionais (ou interfaces de entrada) sempre sinalizam a ocorrência de final de atividade e este sinal deverá ser tratado mesmo que este fato não esteja relacionado a mudanças de estado no sistema produto (condição de *self-loop*). Isto acontece por que, nesta implementação, o sinal ficará ativo até que ocorra o seu tratamento pois a sinalização é feita com retenção. Os autolaços (self-loops) não são implementados nos supervisores e o sistema ignora a ocorrência de eventos que não mudam seu estado. No entanto, para o sistema produto existe uma estrutura que sempre sinaliza as entradas detectadas, sendo necessário implementar os autolaços para tratar estes sinais.

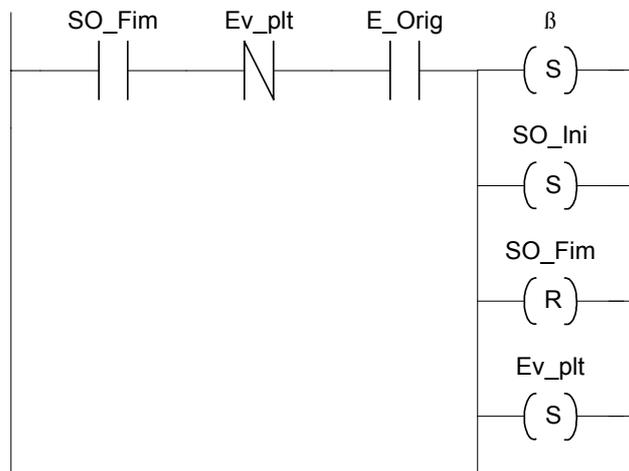


Figura 6.10 - Diagrama escada para implementação de transição não controlável - Autolaço (self-loop).

Para tratar o sinal relativo a um autolaço pode-se implementar uma estrutura similar a da Figura 6.9 mas sem as linhas referentes à mudança de estado, como visto na Figura 6.10. Note que, em ambos os diagramas, todas as variáveis envolvidas estão sendo manipuladas com retenção dos resultados para garantir que serão observadas por todas as estruturas pertinentes do programa e pelo tempo adequado.

6.7.2 Transições controláveis

A teoria de controle supervisorio pressupõe a espontaneidade da ocorrência de eventos, o que é razoável para o caso de eventos não-controláveis mas não para o caso de eventos controláveis, os quais são quase sempre comandos ou sinais que devem ser gerados pelo sistema de controle, não ocorrendo espontaneamente. No entanto, a solução deste problema é bastante simples, sendo que basta vincular a transição controlável de estados à

negação da proibição da ocorrência e não diretamente a ocorrência do evento e assim sempre que for possível e não estiver proibida a transição ocorrerá e o evento será sinalizado ao supervisor.

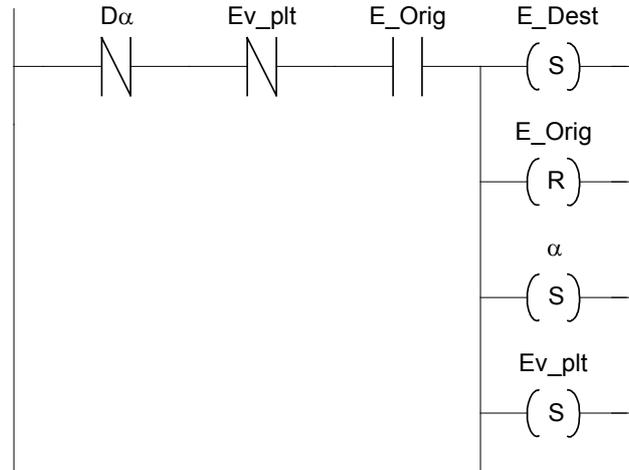


Figura 6.11 - Diagrama escada para implementação de transição controlável.

A Figura 6.11 ilustra a operação lógica em que não estando desabilitado o evento controlável ($D\alpha=0$) e não estando o sistema produto impedido de evoluir ($Ev_plt=0$), ocorrerá a transição entre o estado de origem e o de destino ($E_Orig=0, E_Dest=1$), será sinalizada a ocorrência do evento controlável ($\alpha=1$) e a planta será impedida de evoluir até a atualização dos supervisores ($Ev_plt=1$). Os autolaços de eventos não-controláveis não são implementados visto que são estímulos gerados pelo sistema de controle e não provocam mudança nos estados do sistema físico podendo ser descartados.

6.8 Interfaces de entrada e saída e seqüências operacionais

As seqüências operacionais são muito características de cada subsistema da planta envolvida e dificilmente podem ser generalizadas. Entretanto, um projeto e implementação bastante adequados podem ser realizados utilizando SFC para as operações a exemplo da Figura 3.12, em que um evento controlável (comando) inicia a seqüência que termina com a sinalização de um evento não controlável. Esta é a forma tradicional que leva em conta que um módulo do sistema produto é sempre composto por eventos controláveis e não controláveis sendo que o último sempre é uma consequência da ocorrência do primeiro.

Para modelos compostos somente por eventos controláveis ou não controláveis a estrutura deve ser um pouco diferente, visto que sua ocorrência é totalmente imprevisível para o caso não controlável e desprovida de resposta ou confirmação de execução de operações no caso controlável. Este último caso não representa grande dificuldade, pois se pode estabelecer como serão realizadas as operações e estipular condições para suprir a falta de informação sobre os resultados inerentes destas operações, fazendo com que a implementação não difira muito da tradicional. Já para os modelos totalmente não-controláveis deve-se tomar o cuidado de não se perder informações de eventos ocorridos, bem como

evitar sinalizar várias vezes o mesmo evento antes que o sistema produto possa tratar o primeiro deles.

Como exemplo de implementação de entrada de eventos relativos a modelos totalmente não-controláveis, demonstra-se a seguir uma interface de entrada para uma chave comum (interruptor liga/desliga), particularmente pertinente ao sistema estudado no escopo deste trabalho. Os eventos são relacionados ao sinal para ligar e a negação deste mesmo sinal, sendo que este modelo pode ser utilizado também para o caso de sensores digitais binários.

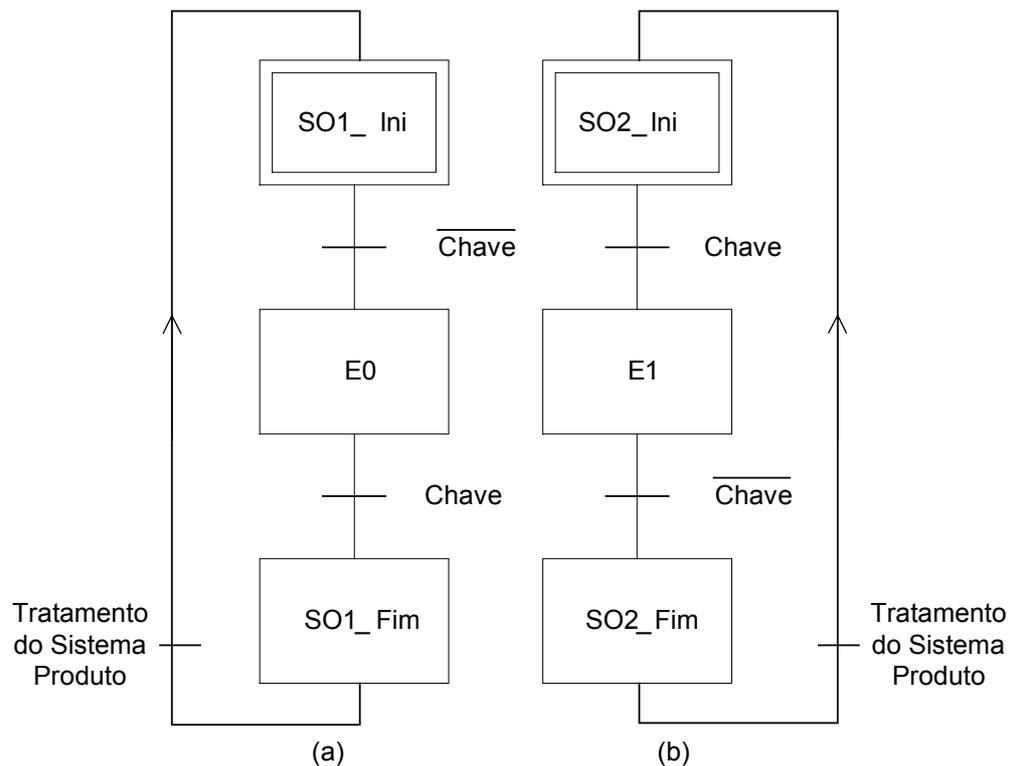


Figura 6.12 - SFC relativo a entrada de sinais de uma chave - (a) Informa quando a chave é ligada, (b) informa quando a chave é desligada.

Os SFC representados na Figura 6.12 sinalizam o funcionamento de um interruptor elétrico sendo que a ilustração (a) sinaliza quando a chave é acionada e (b) quando ela é desligada. No início da execução do programa as variáveis relativas aos passos iniciais de cada estrutura (SO1_Ini, SO2_Ini) são iniciadas em nível lógico alto e caso a chave não esteja acionada (\overline{Chave}) a estrutura (a) passa ao passo E0 e a estrutura (b) fica no passo inicial aguardando que a chave seja acionada. Em caso de chave acionada no início ocorre uma operação similar mas agora (b) fica no passo E1 e aguarda que a chave seja desligada.

De qualquer modo, a posição inicial da chave é ignorada para efeito de controle e a partir do início do programa todas as movimentações serão sinalizadas desde que haja tempo hábil entre a operação da chave pelo usuário e a atualização do sistema produto pela lógica de controle, o que é esperado devido à grande velocidade do ciclo de varredura dos CLPs em comparação ao tempo de acionamento de chaves mecânicas. Assim sendo, os

passos finais (SO1_Fim, SO2_Fim) sinalizam ao sistema-produto a ocorrências de acionamento e desacionamento da chave, respectivamente, e este por sua vez reinicia a interface correspondente (Figura 6.9) colocando-a em condições de detectar um novo estímulo externo (SOx_Fim=0, SOx_Ini=1). O SFC da Figura 6.12 é escrito em diagrama-escada como mostrado na Figura 6.13.

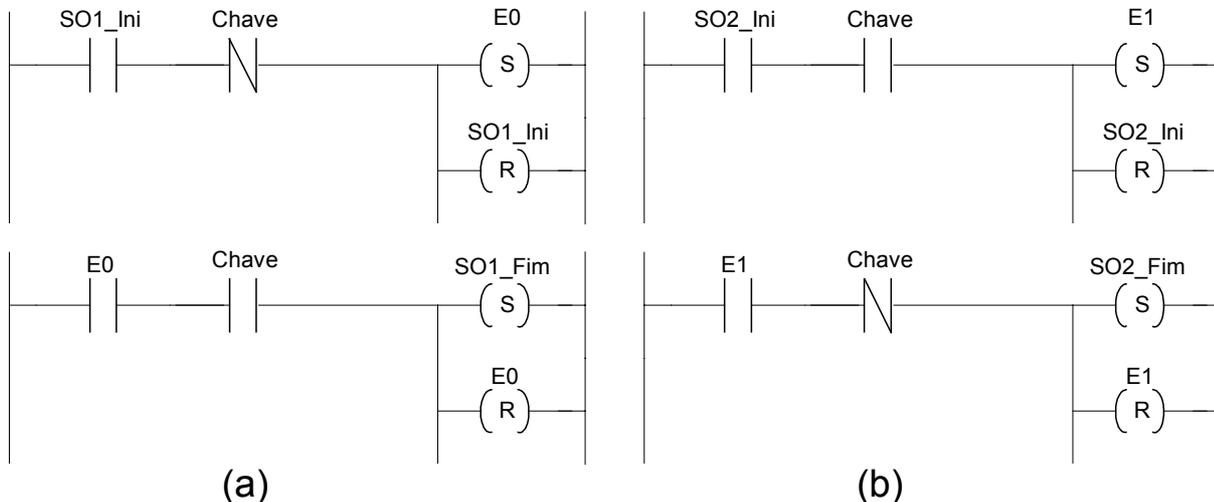


Figura 6.13 - Diagrama escada da interface de entrada de uma chave comum.

6.9 Observações importantes

A estrutura de implementação mostrada não chega a ser um método formal para a implementação de controle supervísório em CLP, mas atende aos objetivos deste projeto, devido à simplicidade de aplicação. Existem outras abordagens formais utilizando outras ferramentas de modelagem como as Redes de Petri, a exemplo de trabalhos como os realizados por HSU e LEE (2000), LIU e DARABI (2002) e HELLGREN (2002).

No presente contexto de controle, os eventos controláveis e não controláveis não são exclusivamente relacionados a comandos e respostas (resultados dos comandos), como nos exemplos de SANTOS (2003) e VIEIRA (2003, 2004). Assim, a implementação vista neste capítulo não impõe limitações à planta livre que pode ser composta por modelos com eventos totalmente não controláveis, controláveis ou qualquer combinação destes casos. Deve-se lembrar, no entanto, que as hipóteses citadas anteriormente de que o sistema é de estados finitos e sem restrição quanto a observabilidade de seus eventos.

Um detalhe observado durante a concepção da estrutura de implementação apresentada, é a necessidade da utilização de variáveis de controle da evolução dos supervisores modulares, de maneira similar à variável 'Ev_plt', para os casos de supervisores com duas ou mais transições simultâneas com o mesmo evento. As variáveis devem ser exclusivas de cada supervisor modular, sendo que também devem estar na estrutura de *reset* das variáveis de controle e variáveis de sinalização de ocorrência de eventos.

A implementação também pressupõe que todas as variáveis relativas aos estados iniciais dos supervisores, ao sistema-produto e aos passos das SFC das seqüências operacionais e das interfaces de E/S, serão inicializadas corretamente no início da execução do programa. As SO e interfaces de E/S têm sua estrutura de programação particularmente dependente dos detalhes e tecnologias envolvidas do hardware do sistema real e, assim, sua inicialização adequada depende desta estrutura.

CAPÍTULO 7

PROJETO DE UM SISTEMA EXEMPLO

7.1 Introdução

Como forma de exemplificar a modelagem proposta no Capítulo 5 e a implementação do Capítulo 6, será apresentado o projeto de um sistema fictício baseado no sistema real estudado, ou seja, um sistema de controle de direcionamento de energia hidráulica com recurso compartilhado. Neste exemplo pressupõe-se a pré-realização de uma fase de projeto informacional que forneceria o conjunto de especificações de projeto para as etapas seguintes de projeto conceitual, preliminar e detalhado. Mostrar-se-á, então, as atividades de modelagem funcional, estrutural e comportamental, síntese de supervisores modulares e projeto de Interfaces de E/S e de seqüências operacionais, relativas ao projeto conceitual. Dentre as atividades relativas às fases de projeto preliminar e detalhado, serão apresentadas somente as etapas de implementação em CLP do controle calculado na fase de projeto conceitual. Estas atividades foram realizadas até a validação do programa em CLP, utilizando-se para isto a bancada de simulação descrita no Capítulo 6.

No transcurso dos estudos realizados e da implementação deste sistema exemplo, um procedimento foi elaborado como forma sistematizar as atividades de projeto, sendo que suas etapas serão descritas a seguir juntamente com a atividade de concepção do sistema exemplo.

7.2 Procedimento de projeto

Descreve-se a seguir o procedimento de projeto utilizado no sistema em questão:

1. Modelagem do sistema energético/material;
 - a. Obter modelo estrutural e funcional;
 - b. Obter modelos de comportamento da planta livre;
 - c. Identificar princípios de solução;
 - d. Especificar comportamento controlado;
2. Modelagem da comunicação com o ambiente externo;
 - a. Obter modelo estrutural e funcional;
 - b. Obter modelos de comportamento relacionados aos pedidos;
 - c. Obter modelos de comportamento relacionados às respostas;
 - d. Identificar princípios de solução para pedidos e respostas;
 - e. Especificar comportamento controlado das respostas;
 - f. Realizar especificações complementares quando necessário;
3. Cálculo do controlador;

- a. Gerar arquivos de modelos e especificações;
 - b. Obter representação por sistema produto;
 - c. Calcular os supervisores modulares;
 - d. Obter supervisores reduzidos;
4. Implementação em CLP;
- a. Obter supervisores e sistema produto em linguagem padrão para CLP;
 - b. Mapear entradas e saídas do CLP em relação ao sistema físico;
 - c. Implementar interfaces de E/S;
 - d. Implementar seqüências operacionais;
 - e. Teste e validação.

Lembra-se que estas atividades não são forçosamente seqüenciais e que têm caráter recursivo, isto quer dizer que algumas atividades podem ser realizadas paralelamente bem como provocarem modificações em outras já realizadas.

7.3 Especificações de projeto

O sistema consiste em duas fontes de energia, na forma de fluido hidráulico pressurizado, que devem ser direcionadas para duas linhas de trabalho, cada uma disponível a um usuário. O funcionamento deste sistema deve obedecer às especificações de projeto seguintes:

- Cada fonte pode ser direcionada para uma única saída por vez, ou seja, uma fonte não pode estar ligada às duas saídas ao mesmo tempo;
- Os usuários devem poder escolher qualquer uma das fontes, nenhuma delas ou as duas ao mesmo tempo;
- O acionamento do sistema deve ocorrer somente através de um comando de início;
- Sempre que possível o sistema deve operar simultaneamente para os dois usuários;
- Caso uma ou mais escolhas não sejam possíveis (recurso indisponível) o usuário deve ser avisado e o sistema não deve ser acionado;
- Em caso de escolha possível o sistema deve ser acionado e o usuário avisado;
- Sempre que um recurso estiver em uso, ambos os usuários devem ser avisados.

7.4 Modelagem do sistema energético/material

7.4.1 Modelagem estrutural e funcional

Utilizando as especificações de projeto, pode-se inicialmente representar o sistema em rede Canal/Agência como mostrado na Figura 7.1.

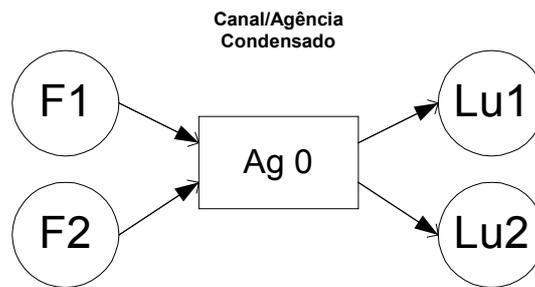


Figura 7.1 - Representação em C/A do sistema energético/material.

Tem-se na figura acima, F1 e F2, respectivamente Fonte 1 e 2, Lu1 e Lu2, respectivamente, linha de trabalho do usuário 1 e 2. Interligando os canais de entrada e saída tem-se a agência geral (Ag0) que deve promover o direcionamento da energia de F1 e F2, para Lu1 e Lu2 segundo as especificações de projeto estabelecidas. Para evoluir em variantes de estrutura e função, refina-se a rede de maneira a se poder dispor de alternativas úteis ao projeto e, para este sistema, refina-se como visto na Figura 7.2. Outras estruturas de refinamento não são apresentadas pela simplicidade do exemplo e assim toma-se o refinamento apresentado como sendo o mais adequado ao projeto.

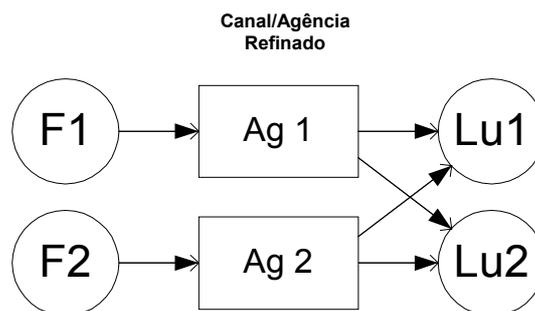


Figura 7.2 - C/A refinado do sistema exemplo.

Nesta estrutura, a agência Ag1 é responsável pelo direcionamento da fonte F1 e a agência Ag2 pelo direcionamento da fonte F2, sendo ambas habilitadas a direcionar as fontes às duas linhas de trabalho.

7.4.2 Modelagem comportamental da planta livre

Com base na representação da Figura 7.2 se pode relacionar os estados possíveis das agências e estipular os eventos que podem compor o modelo comportamental como visto na Tabela 3, onde α_{xy} representa o evento em que a agência x direciona a energia ao usuário y e γ_{xy} o evento em que a agência x interrompe o direcionamento de energia ao usuário y. Os estados são relacionados ao direcionamento das fontes 1 e 2 aos usuários 1 e 2, aos dois usuários ou a nenhum deles.

Tabela 3 - Estimativa de estados e eventos do modelo comportamental.

Estados		Eventos	
Ag1	Ag2	Ag1	Ag2
0 - fechado	0 - fechado	$\Sigma_{ag1} =$ $\{\alpha_{11}, \alpha_{12}, \gamma_{11}, \gamma_{12}\}$	$\Sigma_{ag2} =$ $\{\alpha_{21}, \alpha_{22}, \gamma_{21}, \gamma_{22}\}$
1 - F1 p/ u1	1 - F2 p/ u1		
2 - F1 p/ u2	2 - F2 p/ u2		
3 - F1 p/ u1 e u2	3 - F2 p/ u1 e u2		

Em princípio, considera-se que as agências operam com eventos totalmente controláveis e, modelando-se as sub-funções da agência Ag1, têm-se os seguintes autômatos:

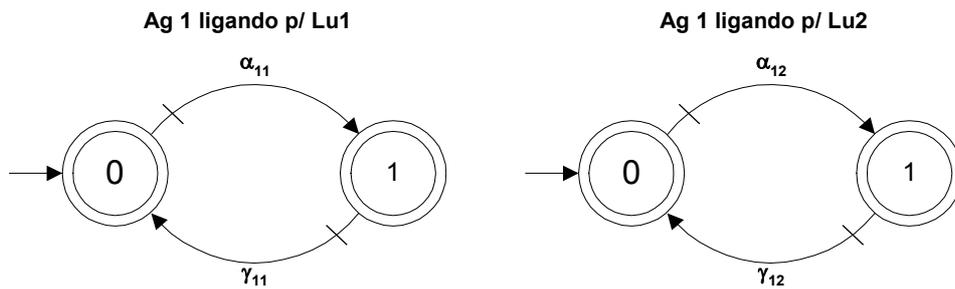


Figura 7.3 - Modelos comportamentais das sub funções de Ag1.

Realizando a composição síncrona dos modelos da Figura 7.3, obtêm-se o modelo de comportamento da função global de Ag1 ilustrado na Figura 7.4.

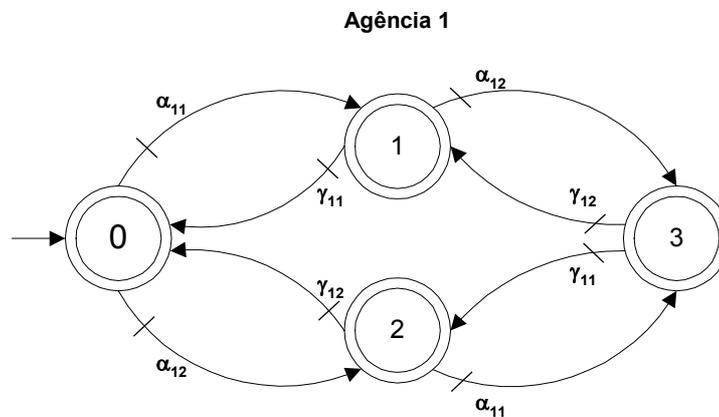


Figura 7.4 - Autômato do comportamento global de Ag1.

Os modelos apresentados não têm um comportamento livre que atenda às especificações de projeto, pois o estado 3 do modelo global de Ag1 fere a proibição de que uma fonte seja ligada às duas linhas de trabalho ao mesmo tempo. Seguindo o procedimento descrito no item 7.2, um dos passos seguintes será a especificação do comportamento controlado e, como o procedimento é recursivo, esta etapa poderá provocar alterações no modelo da planta livre, embutindo-se parte do controle na configuração estrutural e funcional do sistema físico. Para exemplificar esta operação, adianta-se aqui uma especificação de controle para evitar a ocorrência do estado proibido através de dois autômatos Esp1 e Esp2.

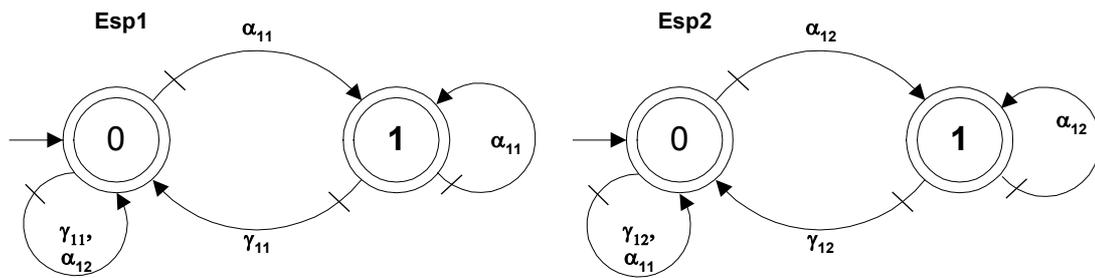


Figura 7.5 - Especificação preliminar do comportamento de Ag1.

As especificações Esp1 e Esp2 evitam que ocorra α_{11} e α_{12} ou α_{12} e α_{11} simultaneamente, impedindo que Ag1 atinja o estado 3. Entretanto, pode-se utilizar estas especificações para realizar uma composição síncrona com Ag1 que resulta em um novo modelo comportamental de Ag1, que denomina-se de Ag1', que pode ser tomado como modelo livre da agência 1.

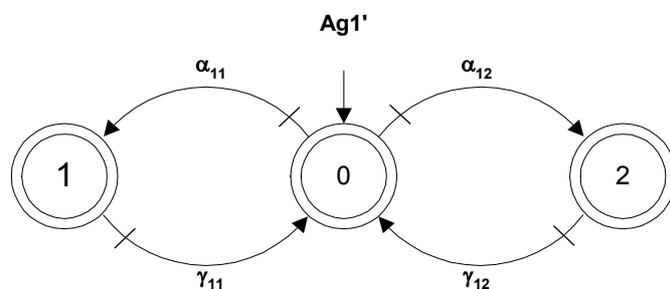


Figura 7.6 - Novo modelo de comportamento da agência 1.

O mesmo procedimento pode ser estendido a agência 2 cujo modelo de comportamento fica:

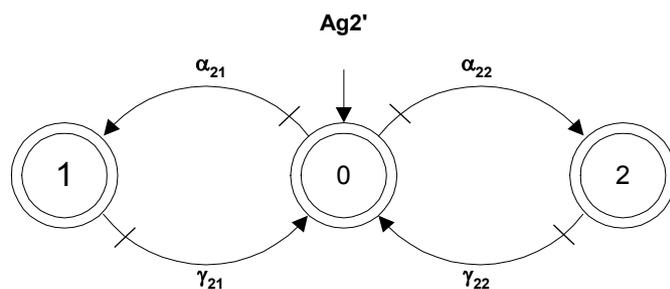


Figura 7.7 - Modelo de comportamento da agência 2.

Nota-se que a estimativa de estado, eventos e a modelagem preliminar do comportamento já é uma forma de restringir o comportamento da planta livre, que de outra forma poderia ser provida de mais estados e eventos que provocassem um comportamento mais complexo, como por exemplo a existência de um evento α_{112} que promovesse o direcionamento da fonte F1 para os dois usuários simultaneamente e não primeiro para um e depois para outro. Desta forma, implantar as restrições de comportamento no modelo da planta livre é uma opção de resolver parte do problema de controle por *hardware* e não por *software*.

7.4.3 Identificação de princípios de solução

Os diferentes modelos gerados no item anterior resultam em diferentes princípios de solução e o projetista pode se valer deste fato para ficar mais cômodo em relação à escolha dos componentes disponíveis para o projeto. Como se está projetando um sistema hidráulico, a solução aos modelos de sub funções das agências (Figura 7.3) pode ser associada a duas válvulas direcionais de duplo solenóide para cada agência, como os tipos mostrados na Figura 7.8.

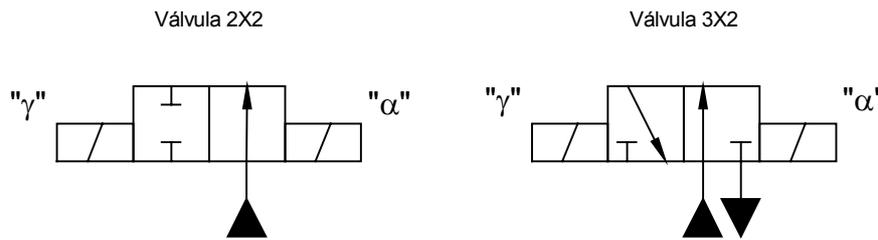


Figura 7.8 - Princípios de solução para os modelos de sub funções.

As válvulas direcionais 2/2 (2 vias e 2 posições) e 3/2 (3 vias e 2 posições) podem ter os acionamentos de seus solenóides associados aos eventos α e γ e, caso fossem de acionadas por simples solenóide e com retorno por mola, bastaria associar γ a negação de α . Estes princípios de solução exigem que o comportamento proibido relacionado ao estado 3 do modelo da Figura 7.4 seja inibido através de controle por software via supervisores modulares, o que por outro lado não é necessário se for possível a utilização de um princípio de solução associado ao modelo da Figura 7.6. Esta solução pode ser implementada através de uma válvula direcional 4/3 (4 vias e 3 posições) ou 5/3 (5 vias e 3 posições), centrada por molas e com centro fechado, como ilustrado na Figura 7.9.

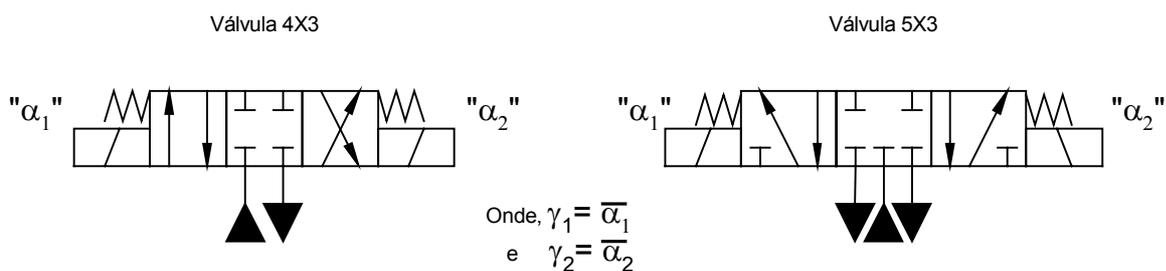


Figura 7.9 - Princípios de solução para Ag1' .

Escolheu-se então o modelo Ag1' e Ag2', com o princípio de solução relacionado a válvula direcional 4/3 mostrada no exemplo. Ressalta-se o fato de que o procedimento descrito não tem a pretensão de ser tomado como um método de projeto sistemático para circuitos hidráulicos. O presente procedimento almeja contribuir paralelamente com a atividade de projeto de circuitos hidráulicos, ou outros como elétricos e pneumáticos, promovendo a melhor integração do projeto físico e de controle.

7.4.4 Especificação do comportamento controlado

Para especificar o comportamento adequado das agências Ag1 e Ag2 (toma-se agora, por simplicidade, esta denominação em detrimento de Ag1' e Ag2') responde-se a seguinte pergunta: Quem determina quando acontece o acionamento de Ag1 e Ag2? A resposta: São os usuários.

Assim, determina-se que uma agência é acionada direcionando a fonte relacionada a certo usuário se este usuário quer que esta agência trabalhe para ele e, o contrário se ele não quer. Respeitando-se as especificações de projeto, deve haver um comando de ligar independente para cada usuário. Desta forma, pode-se prever os eventos que determinarão o funcionamento das agências como mostra a Tabela 4.

Tabela 4 - Eventos que determinam o funcionamento das agências 1 e 2.

	Eventos					
	Quer agência 1	Não quer agência 1	Quer agência 2	Não quer agência 2	Ligar	Desligar
Usuário 1 (u1)	u1qa1	u1nqa1	u1qa2	u1nqa2	u1liga	u1desl
Usuário 2 (u2)	u2qa1	u2nqa1	u2qa2	u2nqa2	u2liga	u2desl

As especificações então são divididas em dois grupos, um que determina o acionamento de acordo com a vontade do usuário e o outro que o acionamento só pode ocorrer após o comando de ligar, como ilustrado na Figura 7.10 e Figura 7.11, respectivamente.

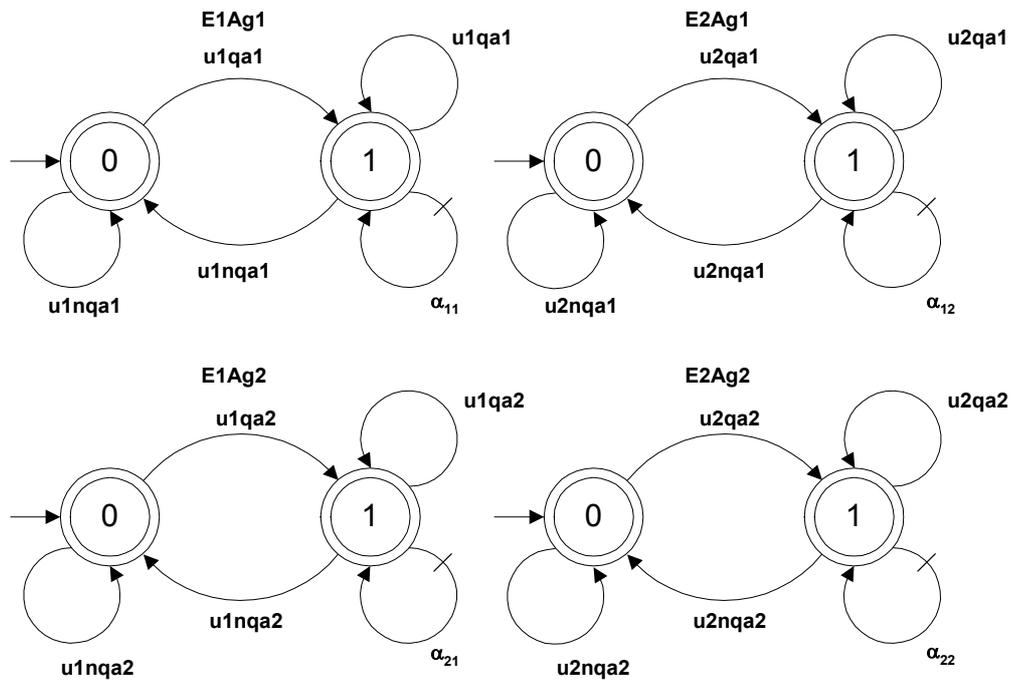


Figura 7.10 - Especificações de acionamento de Ag1 e Ag2, relativas às escolhas dos usuários.

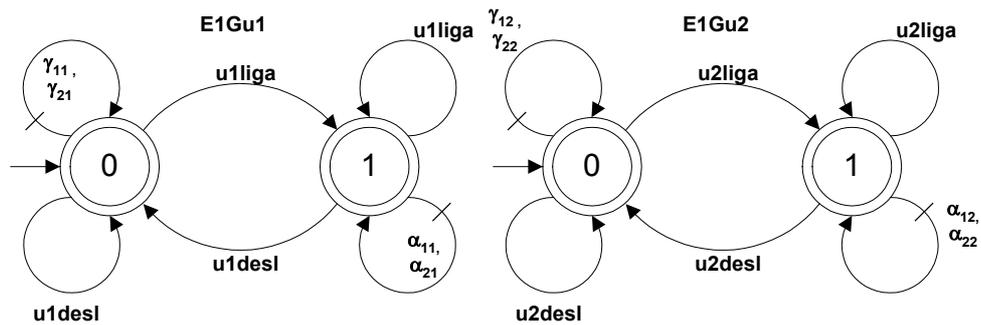


Figura 7.11 - Especificações de acionamento de Ag1 e Ag2, relativas aos comandos de ligar dos usuários.

Nota-se que, os eventos ligados aos usuários fazem parte de um sistema de pedido (SP) ou de um sistema de pedido e resposta (SPR) e, por tanto, são não controláveis.

7.5 Modelagem da comunicação com o ambiente externo

7.5.1 Modelagem estrutural e funcional

O modelo em C/A para o sistema atual pode conter SP, SR e SPR, bem como implementação de entradas e saídas de informação sem modelagem formal, caracterizando uma implementação mista como comentado no Capítulo 5. De fato, este sistema pode ser projetado de maneira mista e, assim, o modelo estrutural e funcional da comunicação com o ambiente externo fica como mostrado na Figura 7.12.

Na figura mencionada, o modelo em C/A refere-se basicamente à estrutura de controle do sistema de informação e nesta fase de projeto já se pode incluir os módulos relativos ao controle do sistema energético/material para que se possa ter uma visão do sistema como um todo. Esta representação permitiu a decisão de não modelar as respostas ao usuário que informam quando um recurso está sendo usado, ou seja, quando uma válvula for acionada em uma dada seqüência operacional (SO); esta mesma SO se encarrega de informar, através de um LED, por exemplo, que aquela agência está em uso.

A escolha de modelar ou não uma resposta, nesta fase, pode ou não ser definitiva dependendo ainda das atividades de formulação das seqüências operacionais e interfaces de E/S, cuja estrutura pode mostrar-se ou não adequada para conter entradas e saídas diretas do ambiente externo.

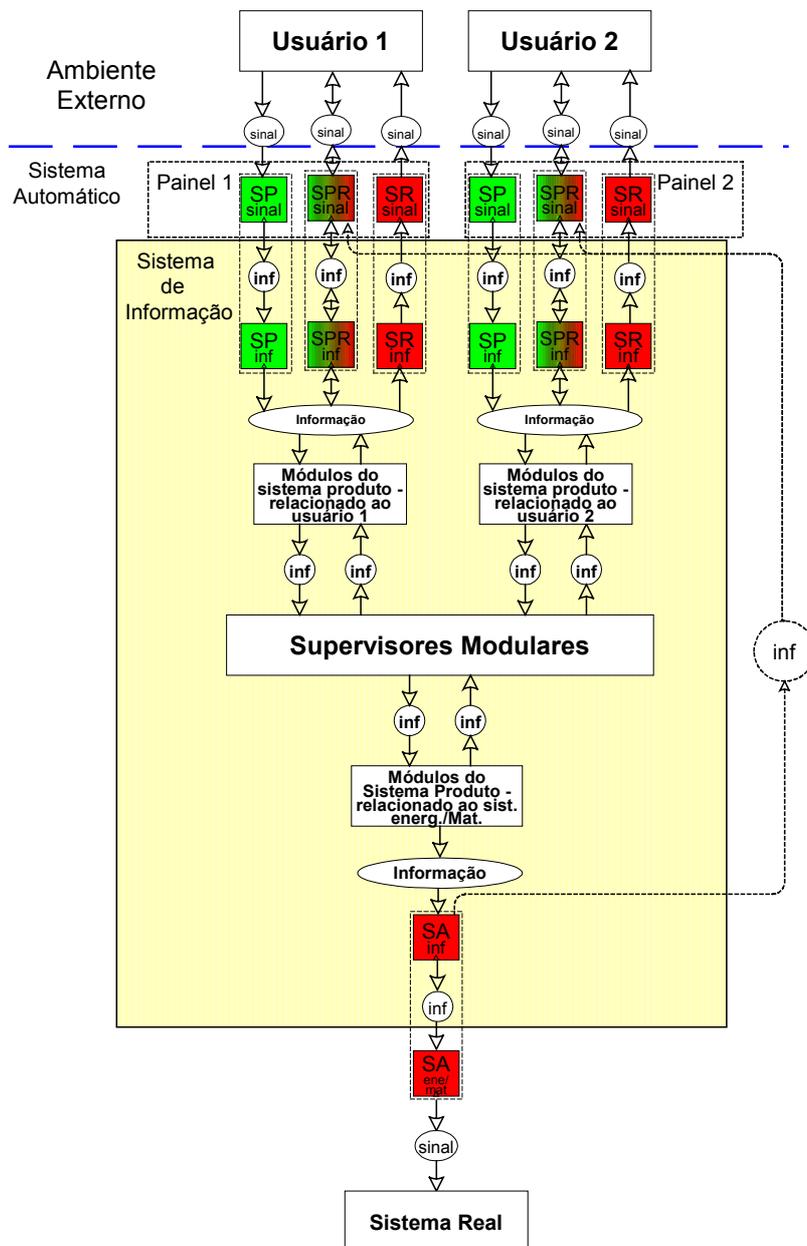


Figura 7.12 - C/A do sistema de informação do projeto teste.

7.5.2 Modelagem de comportamento relacionada aos pedidos

Baseando-se nos eventos da Tabela 4, pode-se formular o modelo de comportamento dos pedidos que reflete os estados do usuário relativos à sua predisposição de fazer escolhas. Embora este modelo seja não controlável, ele de certa forma limita as entradas do usuário a um formato interessante ao sistema e em concordância com as especificações de projeto, utilizando para isto o formato da interface de entrada e o princípio de solução escolhido para promover o fluxo de informação. Escolheu-se aqui um conjunto de modelos em que cada usuário é capaz de fazer escolha entre as agências de maneira independente como descrito na Figura 7.13.

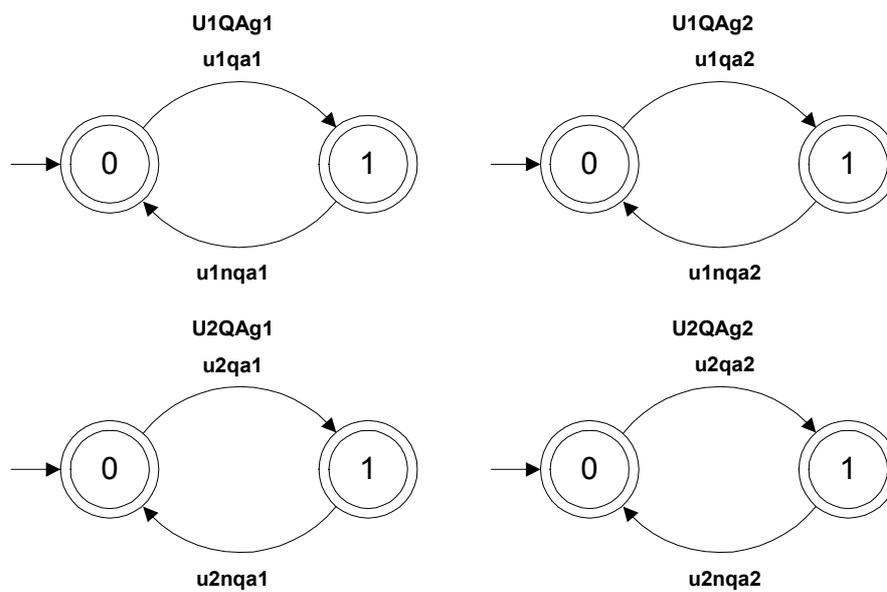


Figura 7.13 - Modelos de comportamento dos pedidos dos usuários 1 e 2 - escolha de agências.

Da mesma maneira formula-se os modelos relativos aos comandos de ligar e desligar.

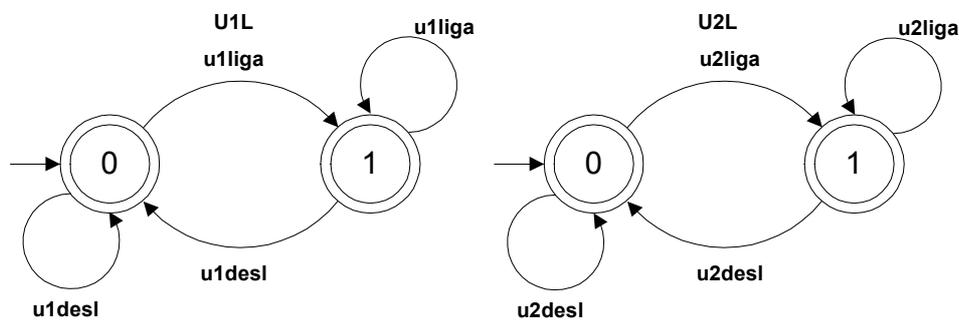


Figura 7.14 - Modelos de comportamento dos pedidos dos usuários 1 e 2 - comando liga/desliga.

Em princípio os modelos podem ser arbitrários, mas é conveniente que sejam dependentes da tecnologia disponível ao projeto, como é o caso dos exemplos acima que são relacionados a princípios de solução específicos e disponíveis no painel simulado da banca-

da de teste. Observa-se também que estes modelos podem não ser únicos como o caso de U1L e U2L da Figura 7.14, cuja linguagem gerada por cada um pode ser representada por um autômato mínimo de um único estado com autolaços de todos os eventos. Os modelos da Figura 7.13 e Figura 7.14 caracterizam-se como modelos de sistemas de pedido (SP).

7.5.3 Modelagem de comportamento relacionada às respostas

Segundo as especificações de projeto, os usuários devem ser avisados do resultado de seus pedidos, ou seja, devem saber se o pedido está apto a ser atendido (pedido OK) ou se existe algum tipo de erro e o pedido não está apto a ser atendido (pedido não OK), além de ambos os usuários ser avisados de quais recursos estão em uso. Esta última informação pode ser implementada de maneira direta como mencionado no item 7.5.1, e assim, não será formulado um modelo de resposta para ela. Para este exemplo escolheu-se o modelo de resposta de três estados, introduzido no Capítulo 5, pois se quer passar informações diferentes para o caso de OK, não OK e nenhuma informação quando não existem pedidos.

Para tanto, determinam-se os eventos relativos como mostrado na Tabela 5, criando-se os modelos relativos às respostas conforme a Figura 7.15.

Tabela 5 - Eventos relativos aos modelos das respostas a u1 e u2.

Modelos	Alfabeto
Resposta ao pedido de Ag1 por u1 (U1RAg1)	$\Sigma_{U1RAg1} = \{u1oka1, u1noka1, u1nqa1\}$
Resposta ao pedido de Ag2 por u1 (U1RAg2)	$\Sigma_{U1RAg2} = \{u1oka2, u1noka2, u1nqa2\}$
Resposta ao pedido de Ag1 por u2 (U2RAg1)	$\Sigma_{U2RAg1} = \{u2oka1, u2noka1, u2nqa1\}$
Resposta ao pedido de Ag2 por u2 (U2RAg2)	$\Sigma_{U2RAg2} = \{u2oka2, u2noka2, u2nqa2\}$

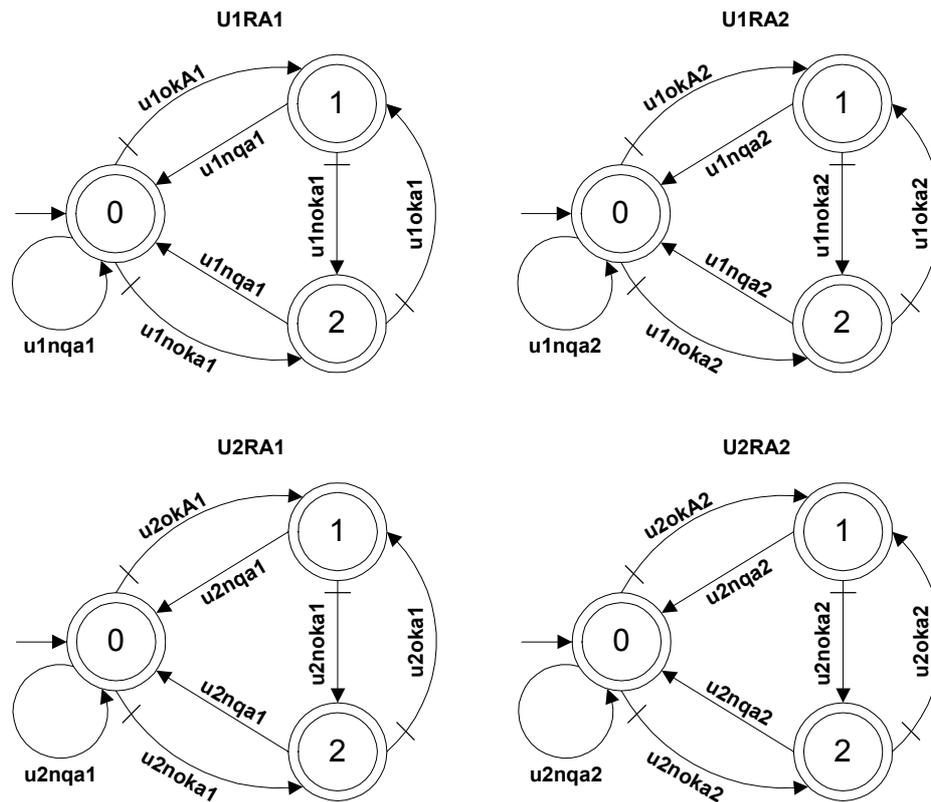


Figura 7.15 - Modelos de comportamento das respostas aos usuários u1 e u2.

Nota-se que nos modelos da Figura 7.15 existem eventos relacionados aos pedidos (u_xnqay), o que os associam com sistemas de pedido e resposta (SPR).

7.5.4 Identificação de princípios de solução para pedidos e respostas

Conforme mencionado anteriormente, a modelagem dos pedidos foi guiada pelos componentes disponíveis na bancada de teste que, para o caso das escolhas de agências, ficou relacionada a interruptores elétricos comuns, uma para cada modelo e, para o caso dos comandos de liga e desliga, ficou relacionada a botões de impulso (do tipo *push-bottom*) que não memorizam estados, dois botões para cada modelo, um para ligar e um para desligar.

Para implementar as respostas escolheu-se os LEDs dispostos nos painéis, um para as respostas OK e outro para as respostas de erro (não OK), em cada painel. Deste modo, todas as interfaces de resposta irão se ocupar apenas com dois elementos de saída de cada painel e, no caso da resposta direta do sistema energético/material que informa a utilização dos recursos, as SO correspondentes acionam em cada painel um LED para cada agência, sempre que comandarem o acionamento de um solenóide qualquer da válvula, sinalizando a indisponibilidade do recurso.

7.5.5 Especificação do comportamento controlado das respostas

Para cada modelo de resposta foram realizadas duas especificações de comportamento controlado, uma determinando que para ocorrer uma resposta OK ou não OK é necessário primeiro querer uma atividade e a outra determinando que a resposta é sempre OK a não ser que a agência não possa atender o pedido. O primeiro caso é implementado através das especificações iniciadas pelo prefixo E1 e o segundo pelo prefixo E2 nos autômatos da Figura 7.16 e Figura 7.17, respectivamente.

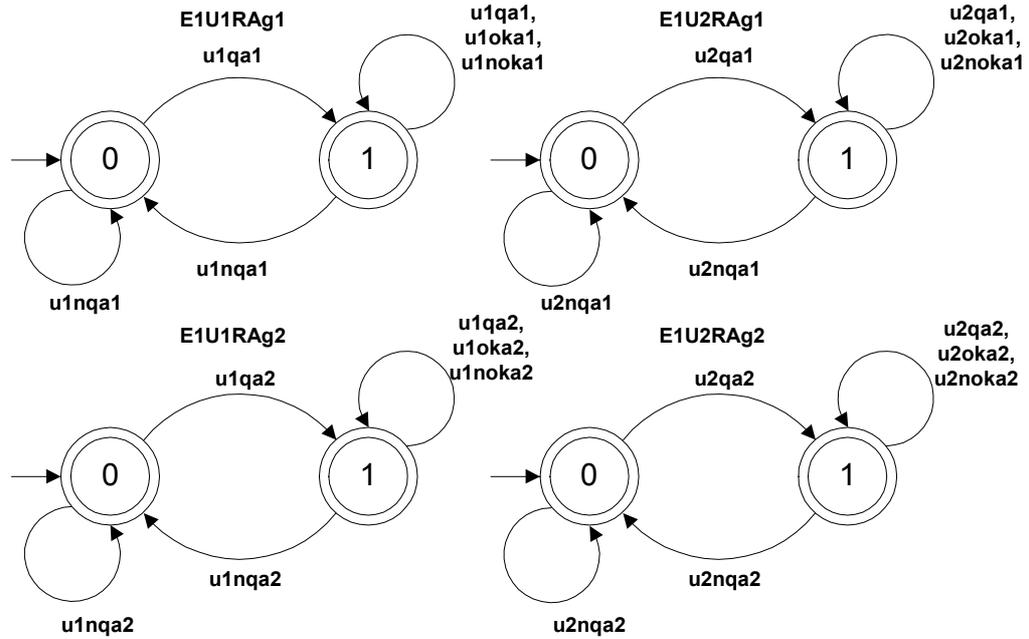


Figura 7.16 - Especificações para as respostas a u1 e u2 - respostas em caso de pedido.

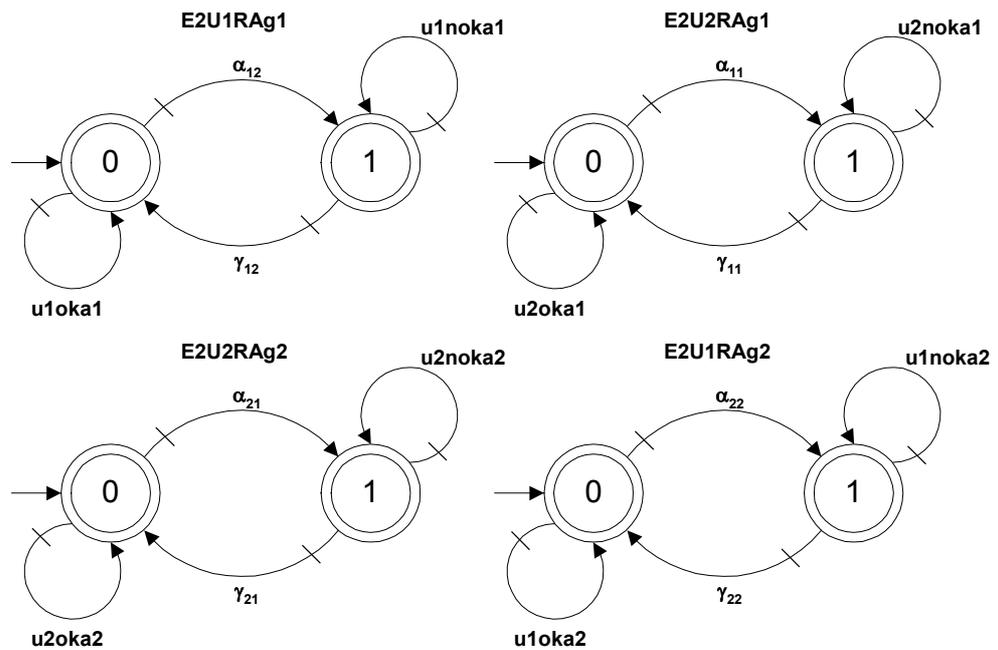


Figura 7.17 - Especificações para as respostas a u1 e u2 - determina o tipo de resposta.

7.5.6 Realização de especificações complementares

O controle projetado até agora atende a quase todas as especificações de projeto, no entanto ainda permite que o sistema acione de forma parcial em caso de um pedido para os dois recursos onde um deles não esta disponível. Neste caso a resposta de “não Ok” será transmitida, no entanto um recurso disponível seria acionado para o usuário enquanto que o outro recurso não. Para evitar atender o pedido de forma incompleta foi modelado um conjunto de especificações complementares para evitar este acionamento parcial, conforme mostrado na Figura 7.18.

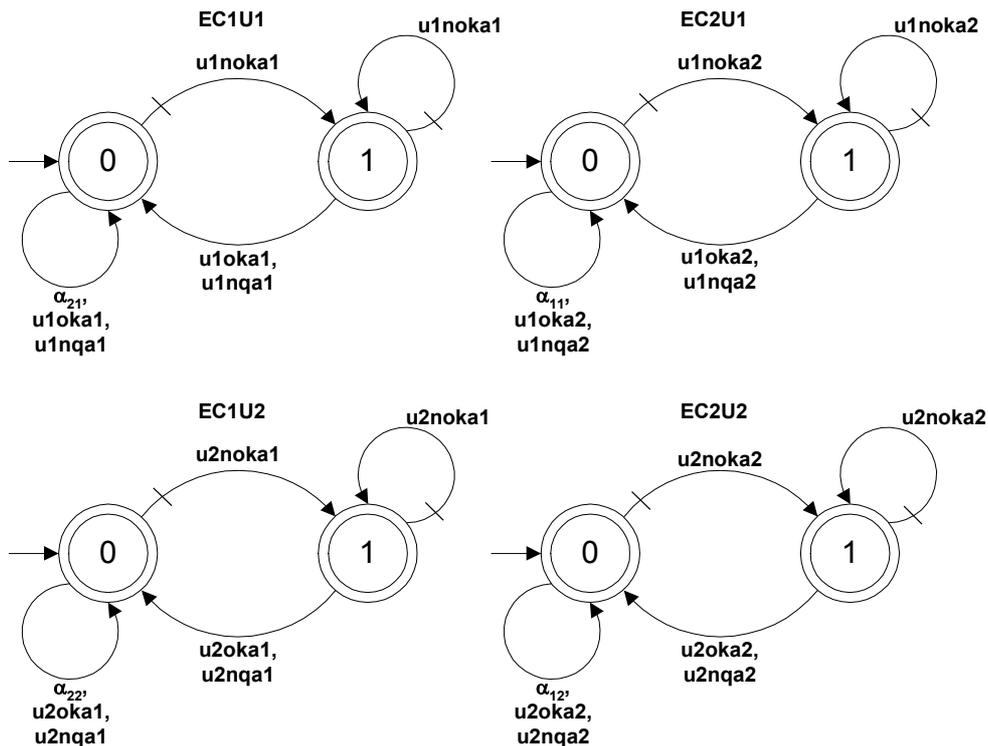


Figura 7.18 - Especificações complementares para evitar acionamento parcial.

Estas especificações impedem o acionamento das agências para um dado usuário sempre que existir uma resposta de erro relacionada a uma das agências.

7.6 Cálculo do controlador

7.6.1 Gerar arquivos de modelos e especificações

Utilizando o formado Grail, foram gerados os arquivos digitais dos modelos e especificações.

Tabela 6 - Relação de modelos e especificações em arquivos Grail.

	quantidade	Relação (nome do arquivo (.txt))
Modelos	12	Ag1, Ag2, U1QAg1, U1QAg2, U2QAg1, U2QAg2, U1L,

		U2L, U1RAg1, U1RAg2, U2RAg1, U2RAg2.
Especificações	18	E1Ag1, E2Ag1, E1Ag2, E2Ag2, E1LU1, E1LU2, E1U1RAg1, E2U1RAg1, E1U1RAg2, E2U1RAg2, E1U2RAg1, E2U2RAg1, E1U2RAg2, E2U2RAg2, EC1U1, EC2U1, EC1U2, EC2U2.

7.6.2 Obter representação por sistema produto

Rapidamente, identificou-se que a intersecção entre dos modelos que compõem a planta e a comunicação não é vazia entre os modelos dos pedidos e suas respostas relacionadas. Assim tem-se a seguinte representação por sistema produto, obtida por operações com funções do Grail:

Tabela 7 - Representação por sistema produto do exemplo de teste.

Sistema Produto	Modelos relacionados
Módulo 1 – MSP1	U1QAg1 // U1RAg1
Módulo 2 – MSP2	U1QAg2 // U1RAg2
Módulo 3 – MSP3	U2QAg1 // U2RAg1
Módulo 4 – MSP4	U2QAg2 // U2RAg2
Módulo 5 – MSP5	Ag1
Módulo 6 – MSP6	Ag2
Módulo 7 – MSP7	U1L,
Módulo 8 – MSP8	U2L

7.6.3 Calcular e reduzir os supervisores modulares

Executando os passos de obtenção das plantas locais, produto síncrono dos módulos do sistema produto relacionados através de uma especificação, obteve-se posteriormente os supervisores locais através do cálculo da máxima linguagem controlável, utilizando-se para isto as próprias plantas locais e os resultados do produto síncrono delas com cada especificação relacionada. Posteriormente os supervisores foram reduzidos e os resultados são mostrados na Tabela 8, sendo que para estas operações foram utilizados os softwares Grail, cálculo de plantas locais e especificações locais, o pacote de conversão Grail-CTCT, para converter plantas e especificações locais de Grail para CTCT e depois os supervisores reduzidos de CTCT para Grail, e o CTCT para calcular e reduzir os supervisores modulares.

Tabela 8 - Resultado dos supervisores locais e supervisores locais reduzidos.

Especificações relacionadas	Supervisores modulares		Supervisores modulares reduzidos	
	Arquivo	Nº de estados (E) e transições (T)	Arquivo	Nº de estados (E) e transições (T)
E1Ag1	Sup1	18 E 75 T	Supr1	2 E 15 T
E2Ag1	Sup2	18 E 75 T	Supr2	2 E 15 T
E1Ag2	Sup3	18 E 75 T	Supr3	2 E 15 T
E2Ag2	Sup4	18 E 75 T	Supr4	2 E 15 T
E1LU1	Sup5	18 E 72 T	Supr5	2 E 16 T
E1LU2	Sup6	18 E 72 T	Supr6	2 E 16 T
E1U1RAg1	Sup7	4 E 10 T	Supr7	2 E 4 T
E2U1RAg1	Sup8	4 E 10 T	Supr8	2 E 4 T
E1U1RAg2	Sup9	18 E 60 T	Supr9	2 E 10 T
E2U1RAg2	Sup10	18 E 60 T	Supr10	2 E 10 T
E1U2RAg1	Sup11	4 E 10 T	Supr11	2 E 4 T
E2U2RAg1	Sup12	4 E 10 T	Supr12	2 E 4 T
E1U2RAg2	Sup13	18 E 60 T	Supr13	2 E 10 T
E2U2RAg2	Sup14	18 E 60 T	Supr14	2 E 10 T
EC1U1	Sup15	18 E 76 T	Supr15	2 E 15 T
EC2U1	Sup16	18 E 76 T	Supr16	2 E 15 T
EC1U2	Sup17	18 E 76 T	Supr17	2 E 15 T
EC2U2	Sup18	18 E 76 T	Supr18	2 E 15 T

Realizou-se também o teste de modularidade dos supervisores locais onde foi constatada a igualdade abaixo:

Sendo,

$$SUPm = (Sup1 \parallel Sup2 \parallel Sup3 \parallel Sup4 \parallel Sup5 \parallel Sup6 \parallel Sup7 \parallel Sup8 \parallel Sup9 \parallel Sup10 \parallel Sup11 \parallel Sup12 \parallel Sup13 \parallel Sup14 \parallel Sup15 \parallel Sup16 \parallel Sup17 \parallel Sup18);$$

Verificou-se que,

$$SUPm = Trim(SUPm).$$

Ou seja, como produto síncrono de todos os supervisores modulares é igual ao TRIM do mesmo produto, os supervisores são modulares e, portanto, não provocam conflito.

Este exemplo é particularmente especial devido à marcação dos estados nos modelos utilizados. No sistema mostrado aqui, e da mesma maneira o projeto descrito no próximo capítulo, todos os estados foram considerados como sendo de tarefa completa, pois os modelos não possuem eventos de sinalização de conclusão de uma seqüência operacional. Esta consideração não é única, sendo totalmente dependente da interpretação do projetista. A marcação de todos os estados permite que o sistema sempre conclua ao menos uma ta-

refa e isto faz com que ele seja sempre modular, ou seja, o teste acima se torna desnecessário.

7.7 Implementação em CLP

Para a implementação em CLP foi utilizado o software CGLI em que os arquivos Grail do sistema produto e supervisores foram convertidos em Lista de instruções (IL no inglês) para o modelo de CLP Bosch CL 151-DP, instalado na bancada de teste. Para esta implementação foram testados os supervisores originais e os reduzidos, e a estrutura utilizada seguiu a proposta do Capítulo 6, sendo as seqüências operacionais realizadas por acionamento direto dos solenóides das válvulas pela variável correspondente ao estado do módulo do sistema produto adequado. Juntamente com os solenóides foi acionado um LED para avisar o recurso em uso, caracterizando a resposta direta de maneira tradicional.

De forma similar, as interfaces de saída também acionaram os LEDs correspondentes observando as variáveis dos módulos do sistema produto relacionados às respostas. As linhas de código de SO e interfaces de E/S foram implementadas manualmente já que o CGLI não gera esta parte do programa. O mapa de variáveis de entradas e saídas do CLP e seus rótulos de chamada no programa já estavam realizados para o teste do controle da unidade hidráulica por álgebra booleana e foram aproveitados neste teste.

7.8 Resultados e conclusões

Após as etapas de projeto, cálculo do controlador e implementação, o funcionamento do sistema foi avaliado e confrontado com as especificações de projeto. Como a gama de opções de funcionamento deste projeto teste é pequena, devido ao número reduzido de componentes no sistema, todas as possíveis formas de acionamento foram testadas, descartadas as opções em que ocorrem falhas físicas do equipamento, como quebra de válvulas ou rompimento de condutores elétricos.

Constatou-se que as especificações foram atendidas por completo no teste da bancada de simulação onde foi possível testar a interação do operador com o sistema através do painel simulado e o acionamento do circuito hidráulico no momento correto através da utilização de duas válvulas do circuito simulado. Durante a implementação, depuração e execução do programa, ficou claro que a estrutura de implementação de controle supervisorio é mais compreensível que o caso de álgebra booleana, mas também pôde ser notado que ela exige mais recursos de *hardware* do CLP.

O *hardware* também foi bastante exigido na etapa de cálculo e redução dos supervisores locais que, mesmo para este exemplo pequeno, requereu a utilização de softwares diferentes para contornar as limitações de memória e reduzir o tempo de processamento.

CAPÍTULO 8

PROJETO DE UMA UNIDADE DE POTÊNCIA HIDRÁULICA

Conforme se citou nos capítulos anteriores, o objeto do estudo realizado neste trabalho foi o projeto de uma unidade de potência hidráulica (UPCH) que fornecerá recursos para uma bancada de trabalho destinada ao ensino e pesquisa em hidráulica, idealizada no Laboratório de Sistemas hidráulicos e Pneumáticos (LASHIP) da UFSC através do Projeto Plataforma Hidráulica Proporcional. Os trabalhos de projeto foram divididos por equipes responsáveis pelas partes mecânicas, hidráulicas, elétricas e de controle, além do projeto das atividades de ensino da bancada e software de apoio a estas atividades (SOUZA, 2003, 2005).

No presente trabalho, as atividades de projeto foram concentradas no controle da unidade hidráulica de potência, cuja equipe de trabalho desenvolveu paralelamente um controle baseado em Álgebra Booleana, utilizando tabelas-verdade e algoritmos numéricos para resolver as expressões booleanas e implementá-las em CLP (SILVA NETO, 2004). Desta forma, mostrar-se-á, nos itens que seguem, somente o projeto do controle da unidade de potência utilizando uma técnica de controle mais refinada (Teoria de controle supervisorio modular local - TCSML), visto que todo o projeto mecânico, hidráulico e elétrico já está concluído, tendo sido realizado concomitantemente com o controlador baseado em Álgebra Booleana citado acima.

8.1 Introdução

A unidade de potência hidráulica da Plataforma Hidráulica Proporcional é composta pelos seguintes recursos que devem ser compartilhados por dois grupos de usuários da bancada de trabalho:

- Uma bomba hidráulica de vazão variável;
- Uma bomba hidráulica de vazão fixa;
- Um acumulador;

As bombas são acionadas por um motor de indução elétrico e o sistema completo da unidade tem ainda sensores de pressão, filtros com sensor de sujeira, sensor de nível do óleo, entre outros, que são componentes comuns a uma unidade hidráulica automática.

A estrutura mecânica da unidade foi projetada de tal modo que as duas bombas serão acionadas pelo mesmo motor elétrico e todos os componentes serão instalados sobre o reservatório de óleo, tais como válvulas, o acumulador, o painel elétrico etc, como ilustrado na Figura 8.1.

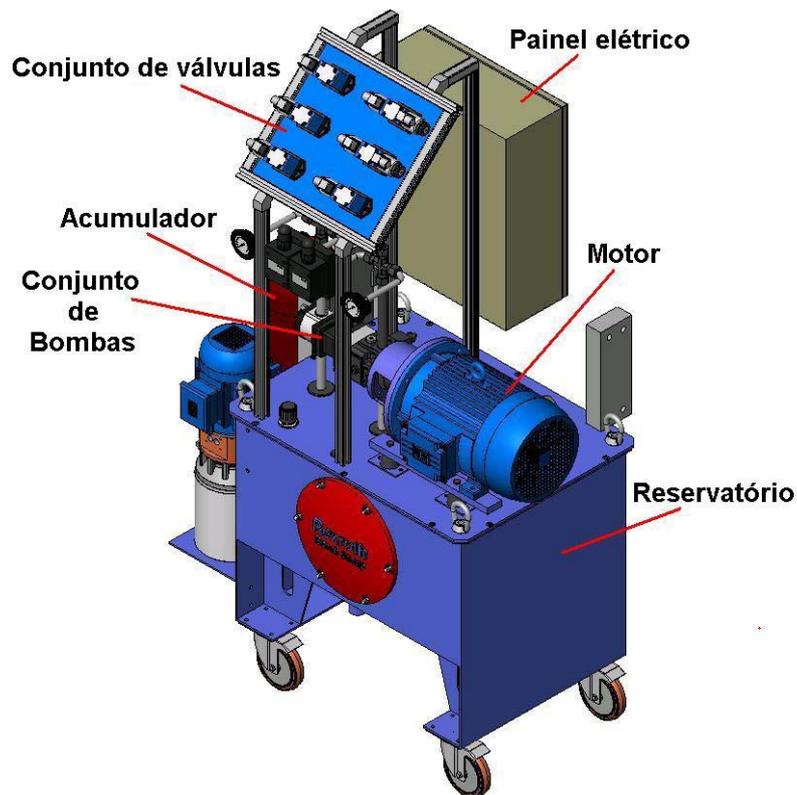


Figura 8.1 - Desenho da unidade hidráulica de potência.

O circuito hidráulico da unidade de potência foi projetado de maneira a fornecer energia (óleo pressurizado) para dois grupos de usuários através de duas saídas em cada lado da bancada de trabalho (saídas P1 e P2). A função dos principais componentes do circuito é descrita na Tabela 9.

Tabela 9 - Componentes do circuito hidráulico e suas funções.

Componente	Função
Motor (0M1)	Acionamentos do conjunto de bombas
Conj. de bombas (0P1, 0P2)	Fornece vazão ao circuito hidráulico
Válvulas (0V3, 0V7)	Direcionam a vazão da bomba variável para as bancadas
Válvulas (0V16, 0V17)	Direcionam a vazão da bomba fixa para as bancadas
Válvula (0V8)	Direciona a vazão do conj. de bombas para o acumulador
Válvula (0V4)	Direciona a vazão do acumulador para as bancadas
Sensor de pressão do acumulador (0S1)	Sinaliza quando o acumulador está cheio
Válvula do acumulador (0V11)	Habilita a carga e descarga do acumulador
Sensor de pressão do filtro (0S2)	Sinaliza quando o filtro está sujo

O circuito hidráulico completo está descrito na Figura 8.2, onde pode-se observar também os componentes auxiliares do circuito como válvulas de alívio, engates rápidos etc.

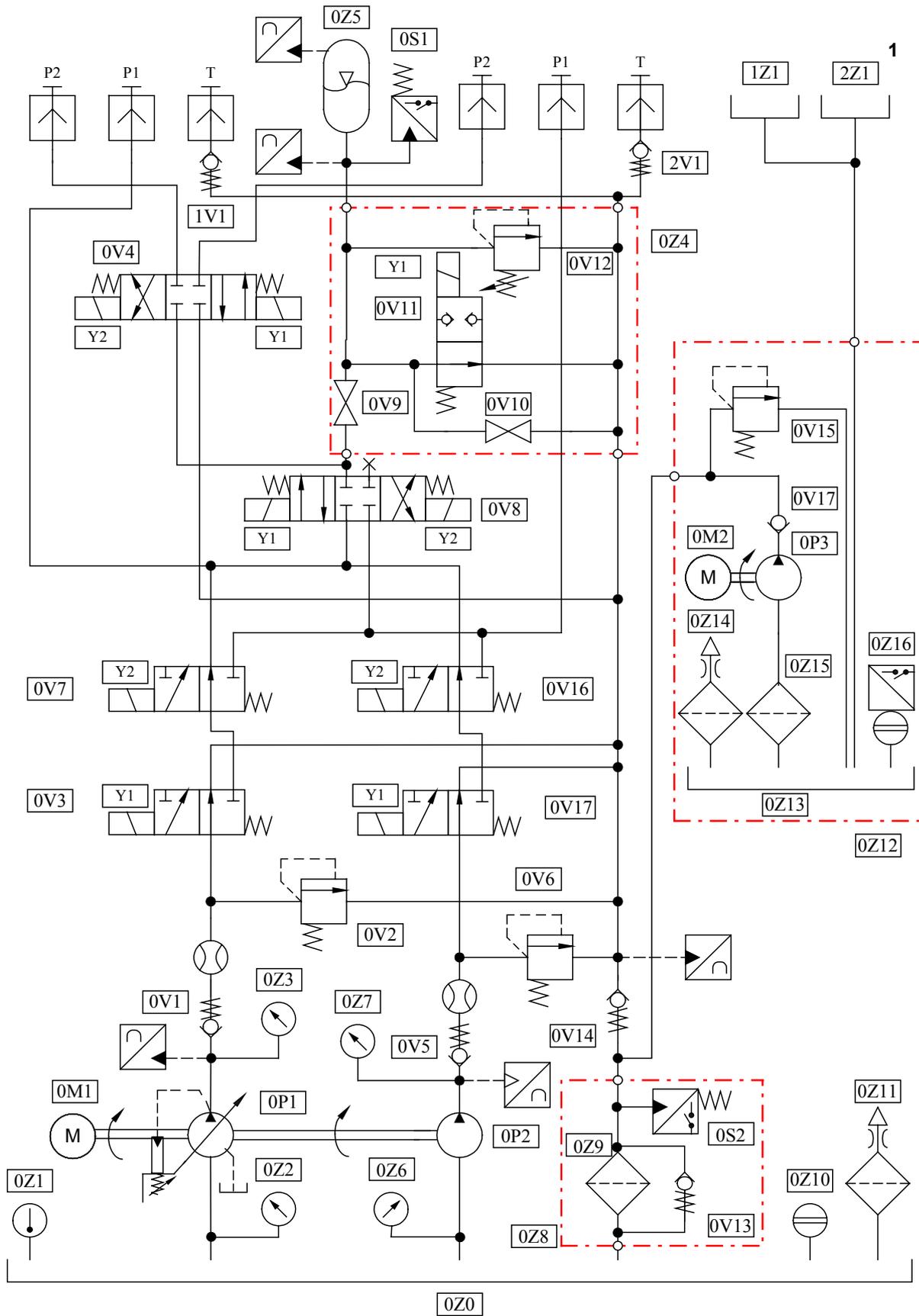


Figura 8.2 - Circuito hidráulico da unidade de potência.

Cada lado da bancada é provido de um painel de controle, por onde os grupos de alunos podem selecionar os recursos da unidade de potência, necessários à experiência que irão realizar, e também receber informações do sistema sobre as condições de funcionamento. As entradas serão proporcionadas por chaves elétricas adequadas e as saídas por sinalizadores luminosos. O desenho esquemático do painel pode ser observado na Figura 8.3.

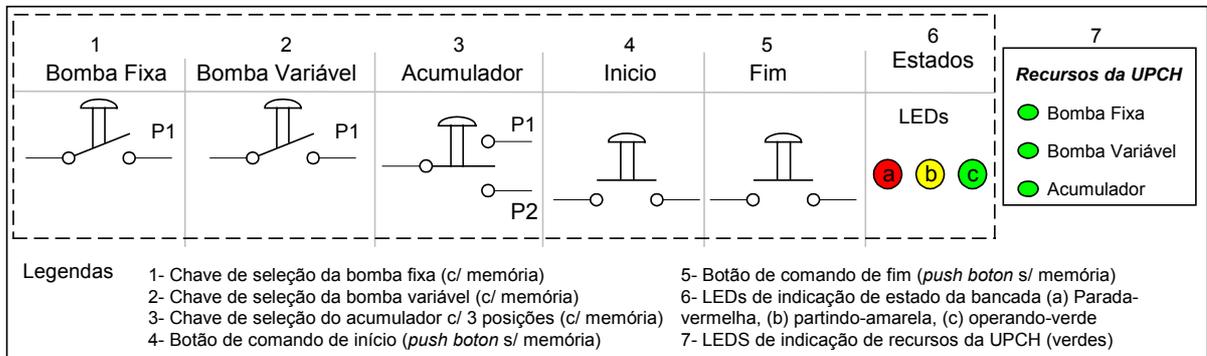


Figura 8.3 - Esquema do painel de comando para as bancadas (adaptado de SOUZA, 2003).

8.2 Especificações de projeto

As especificações mostradas a seguir não foram obtidas através de uma fase formal de projeto informacional, mas para a aplicação da proposta de projeto deste trabalho elas têm o mesmo efeito de especificações obtidas através do método. Como o projeto conceitual em questão está limitado ao projeto do controle da unidade de potência, as especificações apresentadas também serão limitadas aos aspectos de controle das funcionalidades da UPCH, ao compartilhamento dos recursos e ao controle da comunicação com os usuários das bancadas.

As especificações de controle da UPCH e comunicação através dos painéis são:

1. As bombas não devem ser ligadas diretamente às saídas P2 de nenhum dos lados da bancada;
2. As bombas só devem ser utilizadas por um lado de cada vez, ou seja, uma única bomba não pode estar direcionada para os dois lados da bancada ao mesmo tempo;
3. O acumulador só deve ser utilizado em um lado da bancada de cada vez, ou seja, ele não pode estar conectado aos dois grupos de usuários ao mesmo tempo;
4. Para utilizar o acumulador uma das bombas deve enchê-lo sendo que a bomba variável tem preferência nesta atividade;
5. Para utilizar o acumulador na saída P1 o usuário deve escolher também pelo menos uma bomba;
6. Os recursos escolhidos pelos usuários somente serão destinados a eles após o comando de início das atividades (ligar);

7. Os recursos destinados aos usuários somente serão liberados após o comando de fim das atividades (desligar);
8. Quando o acumulador for escolhido por um usuário para a saída P2 deve existir ao menos uma bomba disponível para enchê-lo e esta será acionada temporariamente para esta atividade, ficando posteriormente o recurso disponível para o outro usuário;
9. Os estados de cada bancada devem ser informados ao usuário através de sinalizadores luminosos (parada-vermelha, partindo-amarela, operando-verde) sendo: 'parada' para a bancada desligada, 'partindo' para bancada ligada com acumulador enchendo ou bombas temporariamente indisponíveis e 'operando' para bancada ligada com recursos solicitados em operação;
10. As escolhas de recursos indisponíveis ou escolha errada (acumulador em P1 sem bombas) devem ser informadas ao usuário através das luzes de sinalização de estado da bancada (vermelha, amarela, verde), com as três luzes piscando;
11. Os recursos em uso (indisponíveis) devem ser informados às duas bancadas simultaneamente (sinalizadores luminosos verdes);
12. Não deve haver atendimento parcial de pedidos dos usuários, ou seja, caso um dos recursos solicitados não esteja disponível nenhum pode ser acionado;
13. Erros críticos devem ser sinalizados aos dois painéis simultaneamente com o acionamento intermitente de todas as luzes dos painéis. São erros críticos: a parada do motor das bombas, o sinal de filtro sujo e o sinal de acumulador cheio sem o acionamento do acumulador.

8.3 Modelagem do sistema energético/material

8.3.1 Modelagem funcional e estrutural

O circuito hidráulico da UPCH, descrito na Figura 8.2, pode ser modelado em rede C/A condensada como mostrado na Figura 8.4.

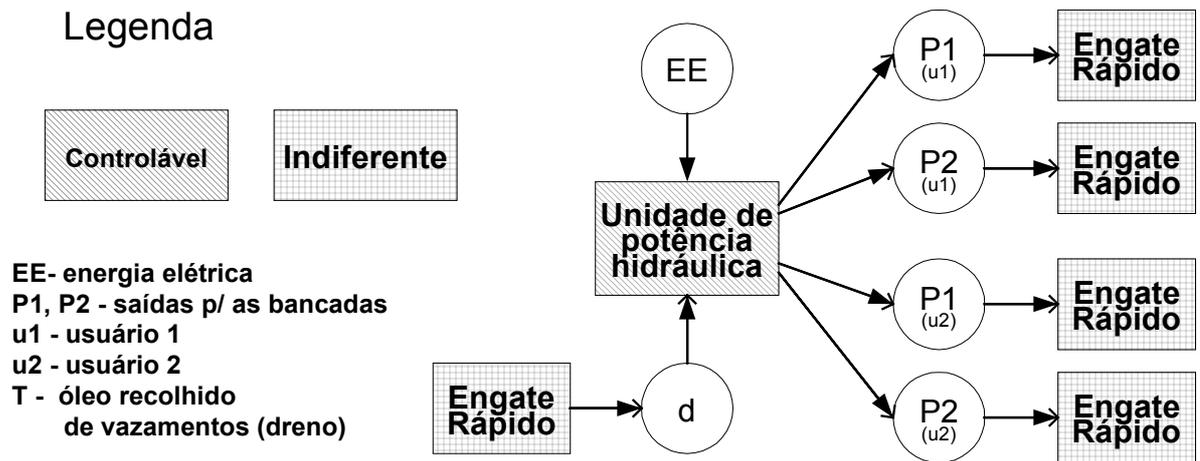


Figura 8.4 - Modelo em rede C/A condensada da UPCH .

Pode-se refinar o diagrama da Figura 8.4 de modo a se obter uma representação mais detalhada. Para tanto, identificou-se as estruturas do circuito hidráulico que são importantes para a modelagem e controle do sistema, mostradas na Figura 8.5, possibilitando criar a Rede C/A mostrada na Figura 8.6. A Tabela 10 descreve com mais detalhes os componentes identificados na Figura 8.5.

Tabela 10 - Elementos componentes dos conjuntos importantes ao controle.

Conjunto	Elementos componentes
Conjunto de bombas	0M1, 0P1, 0P2
Filtro	0Z9, 0S2
Direcionador da bomba variável (BV)	0V3, 0V7
Direcionador da bomba fixa (BF)	0V16, 0V17
Direcionador 2	0V8
Direcionador 3	0V4
Acumulador	0Z5, 0S1, 0V11

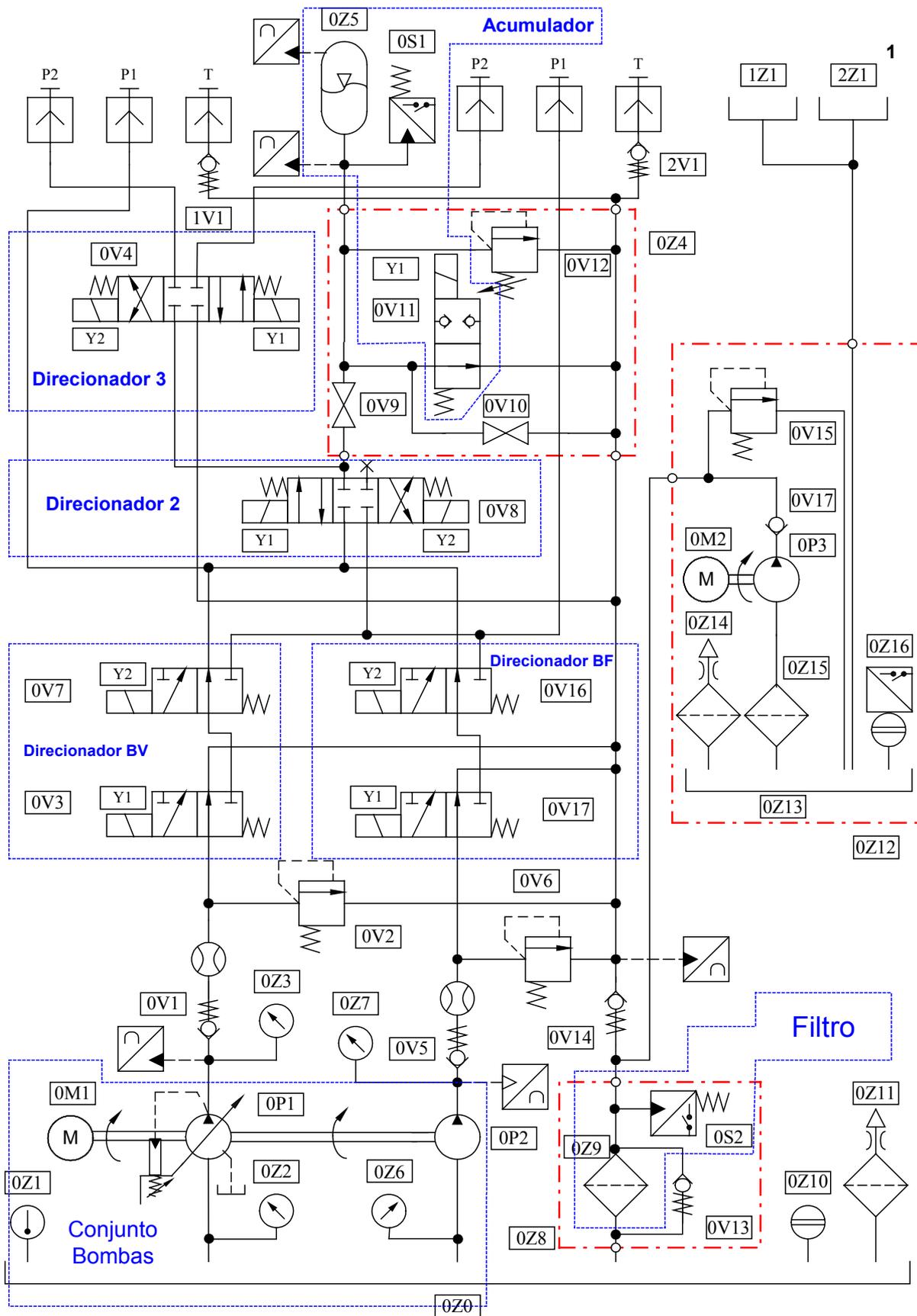


Figura 8.5 - Entidade do circuito hidráulico importantes ao controle.

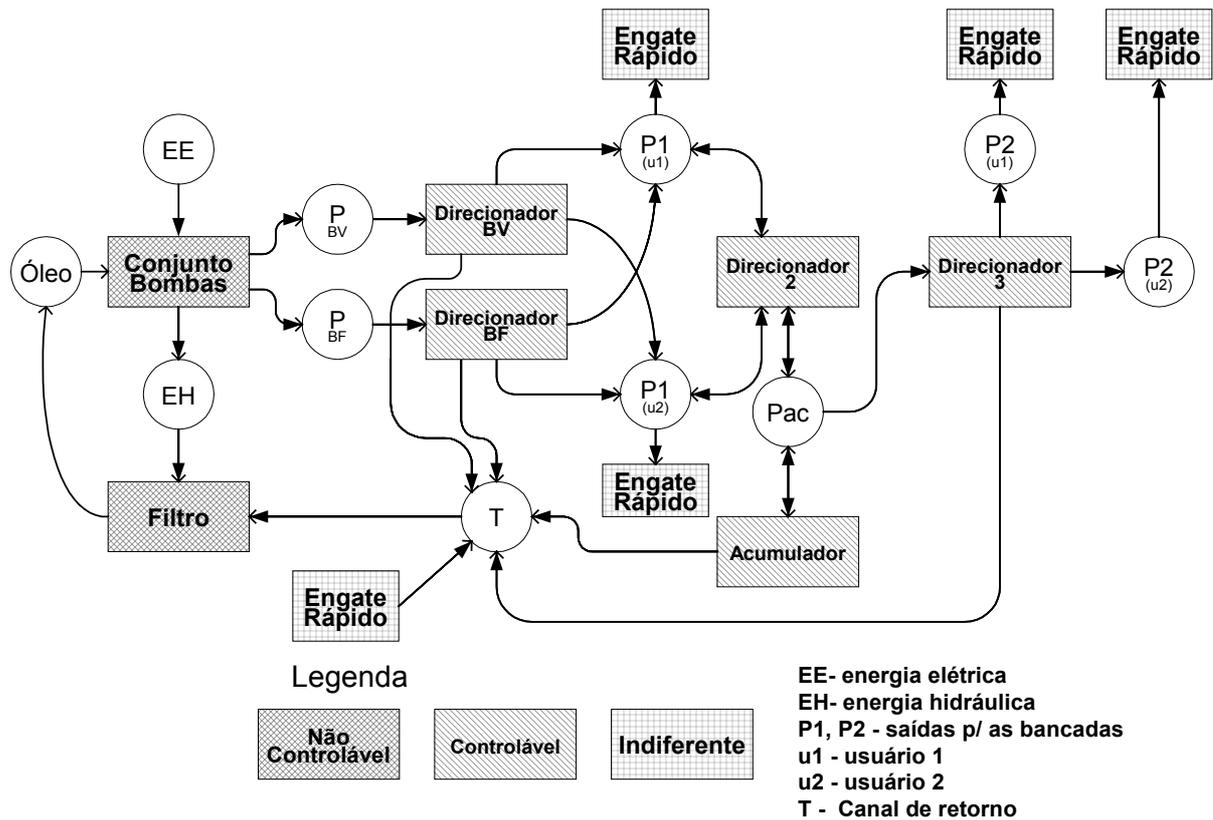


Figura 8.6 - Rede C/A refinada da UPCH.

Os engates rápidos são elementos puramente mecânicos desconsiderados para efeito de controle. O motor que aciona as bombas hidráulicas é acionado por uma chave de partida comandada diretamente pelo operador (especificação de projeto) mas como emite um sinal informando seu estado, caracteriza-se como um sistema de medição (SM). De forma semelhante, como o filtro possui um sensor de ensujamento, este é tratado como um sistema de medição (SM). As agências denominadas direcionador BV, BF, 2 e 3 são sistemas puramente de atuação (SA), já que não são munidas de nenhum sensor de confirmação do acionamento. O acumulador, que contém um componente de atuação controlável (válvula) e um sensor que indica quando ele está cheio (pressostato), caracteriza-se como um sistema de atuação-medição (SAM).

8.3.2 Modelagem comportamental

Como o projeto do circuito hidráulico já está realizado, a escolha dos princípios de solução também já foi realizada e, assim, a modelagem comportamental limita-se a identificar o comportamento dos componentes físicos escolhidos e modelá-los em autômatos. Para efeito de simplificação, os passos descritos no item 7.4.2 sobre especificação através dos componentes físicos serão excluídos, sendo mostrados os modelos finais escolhidos para compor o sistema de controle.

Iniciando pelos sistemas de medição (SM), tem-se os modelos do conjunto de bombas e do filtro, descritos na Figura 8.7. As referências de todas as figuras de modelos e especificações a seguir com arquivos de texto (.txt) são feitas em relação aos arquivos Grail gerados para o cálculo dos supervisores modulares.

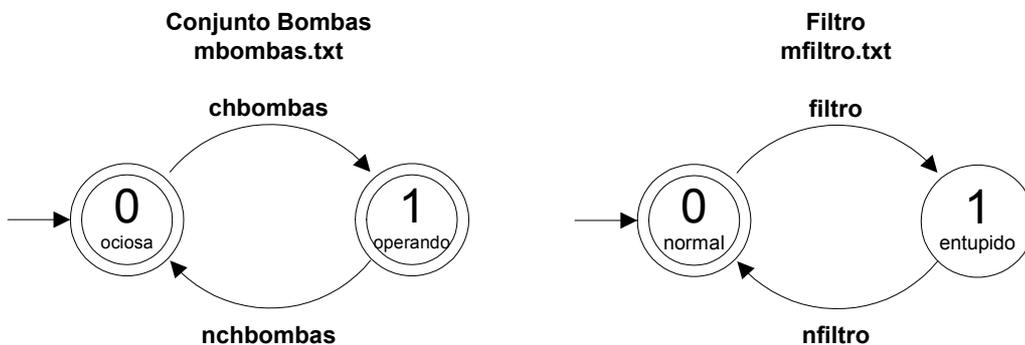


Figura 8.7 - Autômatos do conjunto de bombas e do filtro.

Os eventos {chbombas, nchbombas} estão relacionados ao sinal da chave de partida rápida do motor de indução que aciona as bombas e os eventos {filtro, nfiltro} estão relacionados ao pressostato que indica quando o filtro está obstruído.

Os modelos dos sistemas de atuação (SA) direcionadores, ficam na forma da Figura 8.8, Figura 8.9, Figura 8.10 e Figura 8.11. A Tabela 11 apresenta o significado dos eventos empregados.

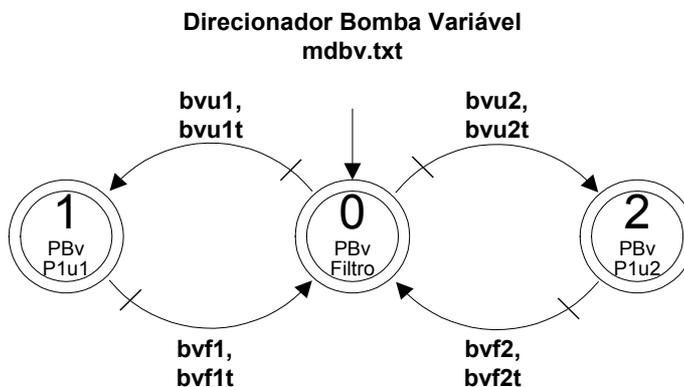


Figura 8.8 - Autômato do direcionador da bomba variável.

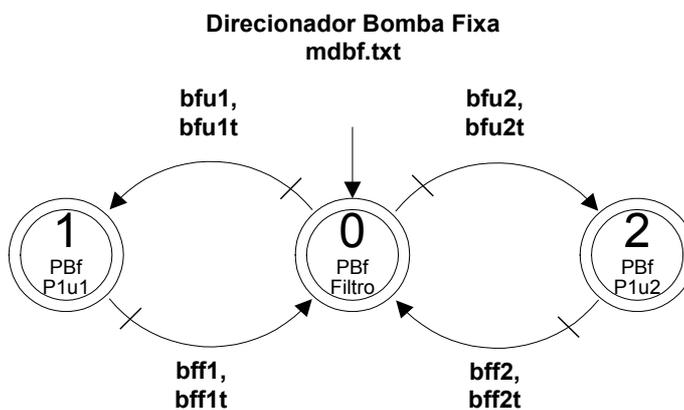


Figura 8.9 - Autômato do direcionador da bomba fixa.

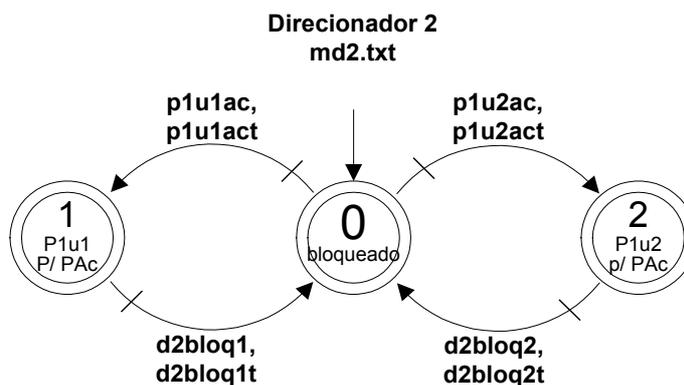


Figura 8.10 - Autômato do direcionador 2.

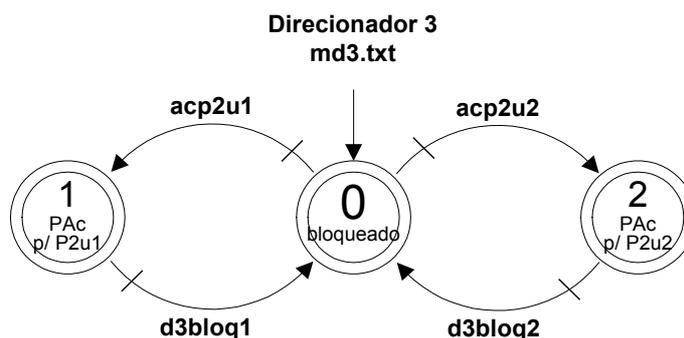


Figura 8.11 - Autômato do direcionador 3.

Tabela 11 - Eventos do sistema de atuação (SA).

Eventos $x=\{1,2\}$	ação
bvux	Ligar BV para usuário x permanentemente
bvuxt	Ligar BV para usuário x temporariamente
bvfx	Fechar BV aberta para usuário x permanent.
bvfmt	Fechar BV aberta para usuário x tempor.
bfux	Ligar BF para usuário x permanentemente
bfuxt	Ligar BF para usuário x temporariamente
bffx	Fechar BF aberta para usuário x permanent.
bffxt	Fechar BF aberta para usuário x tempor.
p1uxac	Ligar P1 do usuário x ao acumulador perman.
p1uxact	Ligar P1 do usuário x ao acumulador temp.
d2bloqx	Fechar direcionador 2 aberto para P1 de U- suário x permanentemente.
d2bloqxt	Fechar direcionador 2 aberto para P1 de U- suário x temporariamente.
acp2ux	Ligar o acumulador a P2 do usuário x
d3bloqx	Fechar direcionador 3 aberto para P2 do u-

	suário x.
--	-----------

O último modelo relacionado ao sistema energético material é um sistema de atuação-medição (SAM), o acumulador, descrito na Figura 8.12.

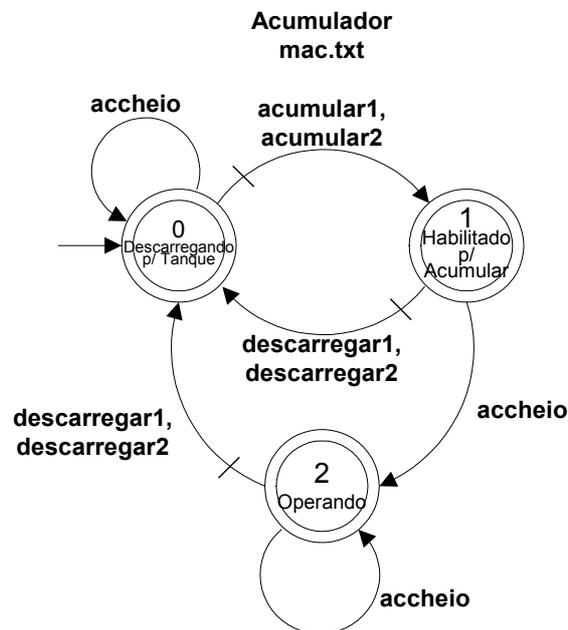


Figura 8.12 – Autômato do acumulador.

Os eventos {acumular1, acumular2} comandam o acumulador a se habilitar a acumular para os usuários 1 e 2, respectivamente, e, de maneira similar, os eventos {descarregar1, descarregar2} comandam a descarga para reservatório. O evento {accheio} está relacionado ao sinal do pressostato do acumulador que sinaliza quando este está cheio.

8.3.3 Especificação do comportamento controlado

Para atender às especificações listadas no item 8.2 e fazer com que a planta livre (modelos do item 8.3.2) as obedeça, modela-se as especificações de projeto em autômatos. A Figura 8.13 descreve as especificações que atendem ao sub item 1 do item 8.2 .

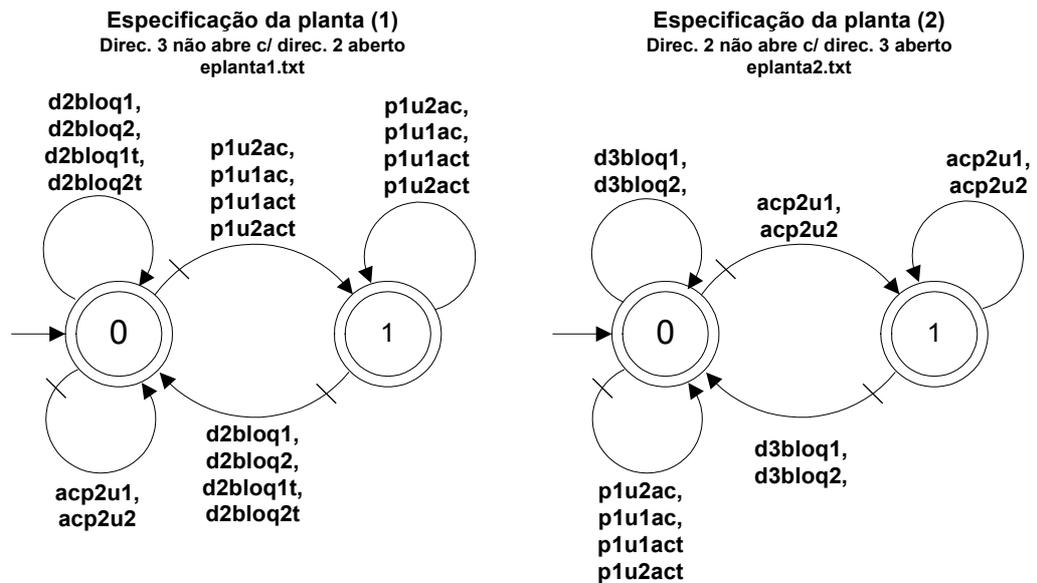


Figura 8.13 - Especificações da planta sem a comunicação com o ambiente externo.

A preferência da bomba variável para encher o acumulador, especificação citada no sub item 4, está modelada nos autômatos da Figura 8.14.

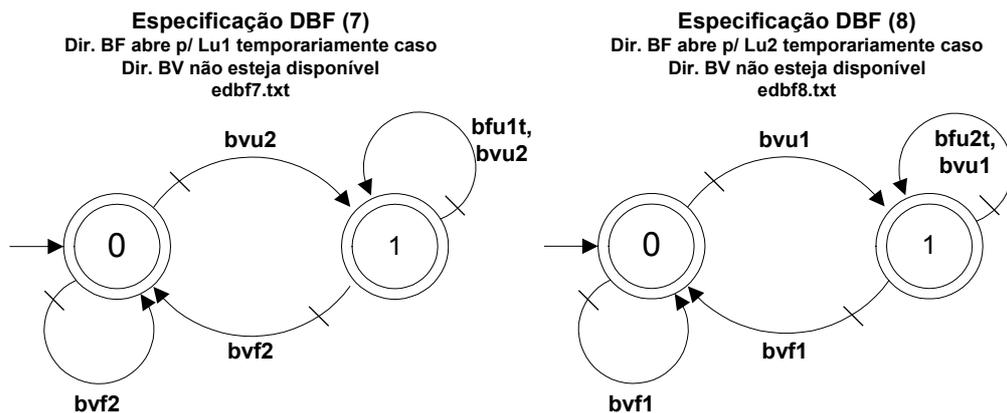
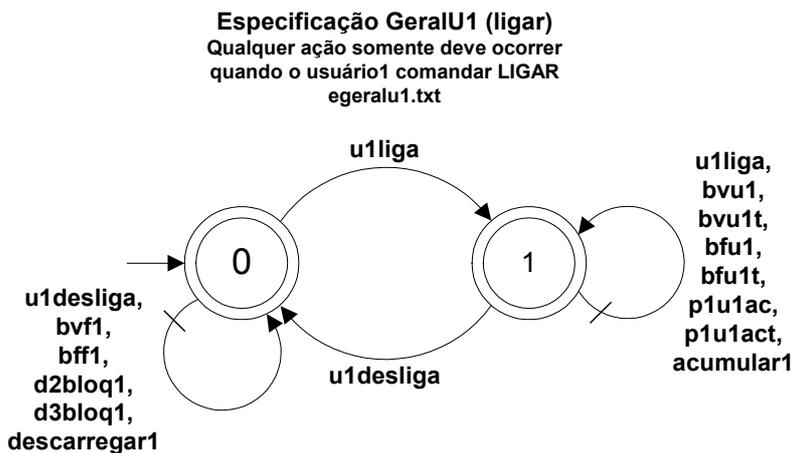


Figura 8.14 - Especificação para preferência de BV no enchimento do acumulador.

As demais especificações estão modeladas em relação a cada modelo das agências cujo acionamento tem relação com os pedidos dos usuários aos quais os modelos ainda são desconhecidos. As especificações a seguir descrevem os sub itens do item 8.2 relacionados ao funcionamento das agências e ao compartilhamento dos recursos, sem levar ainda em conta as respostas. Como as especificações em relação aos dois usuários (dois lados da bancada) são simétricas, mostrar-se-á apenas as relacionadas ao usuário 1 e as compartilhadas pelos dois usuários.



Especificação para a o direcionador da Bomba Variável

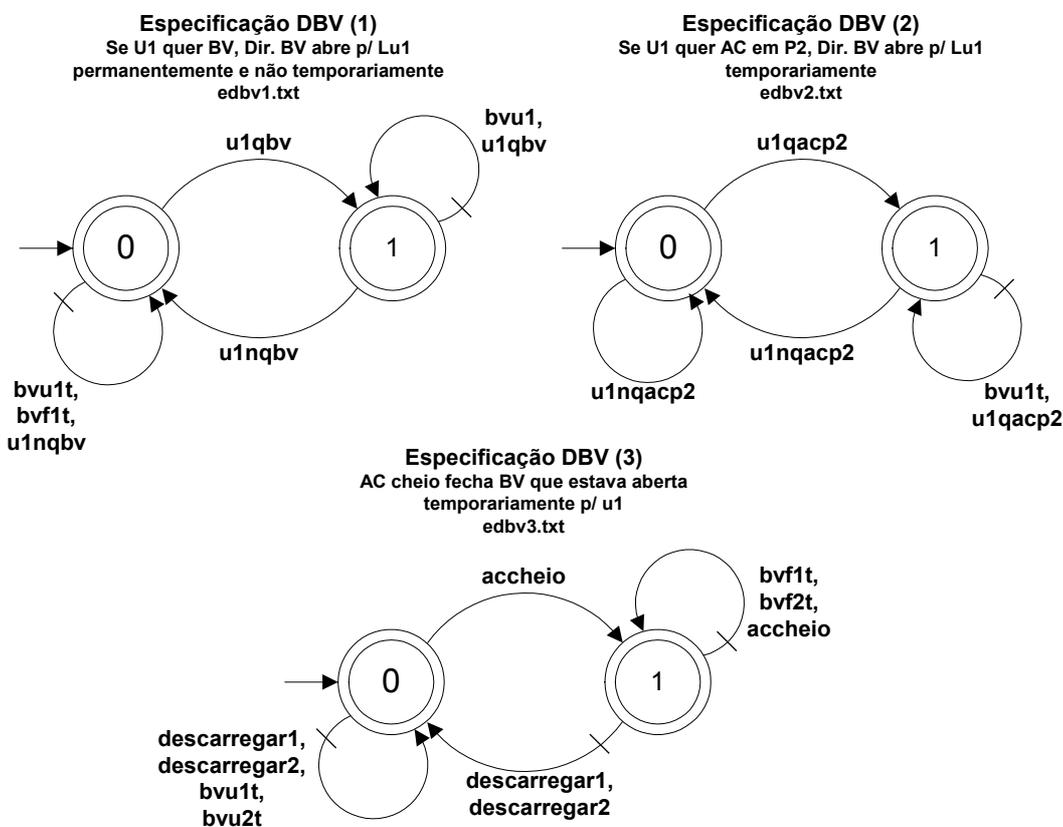
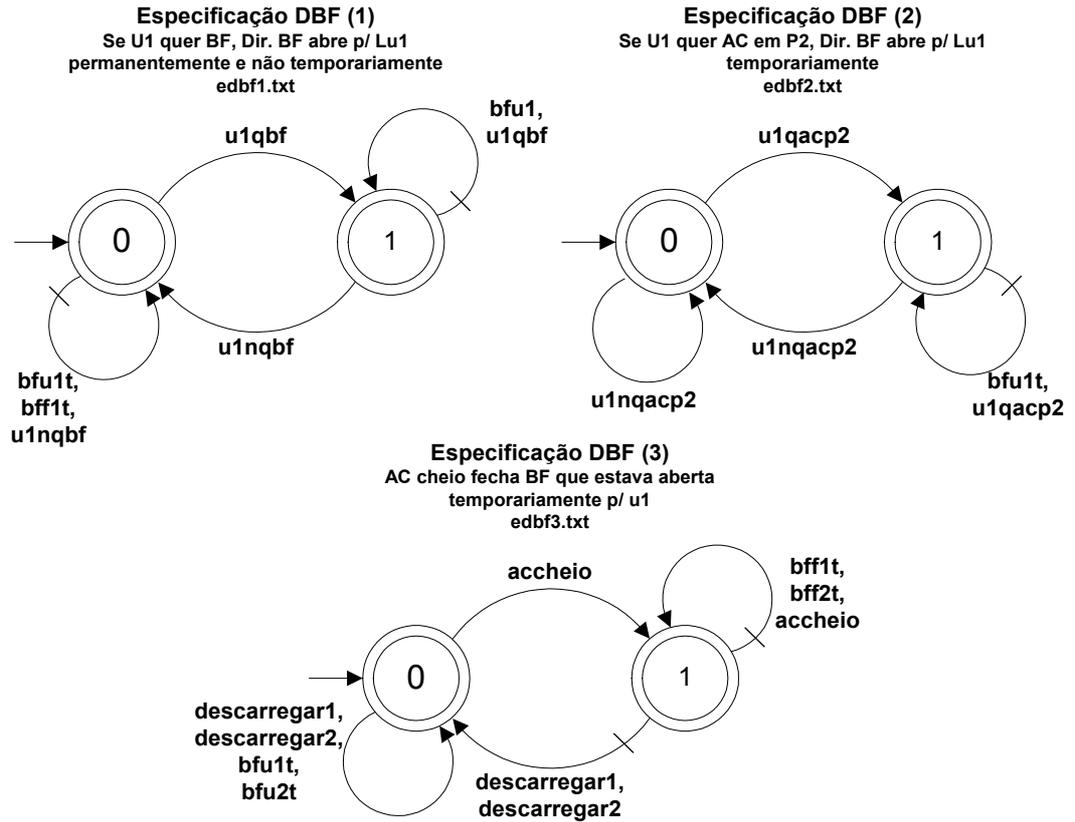


Figura 8.15 - Especificações geral e para a bomba variável.

Especificação para a o direcionador da Bomba Fixa



Especificações para a o direcionador 2

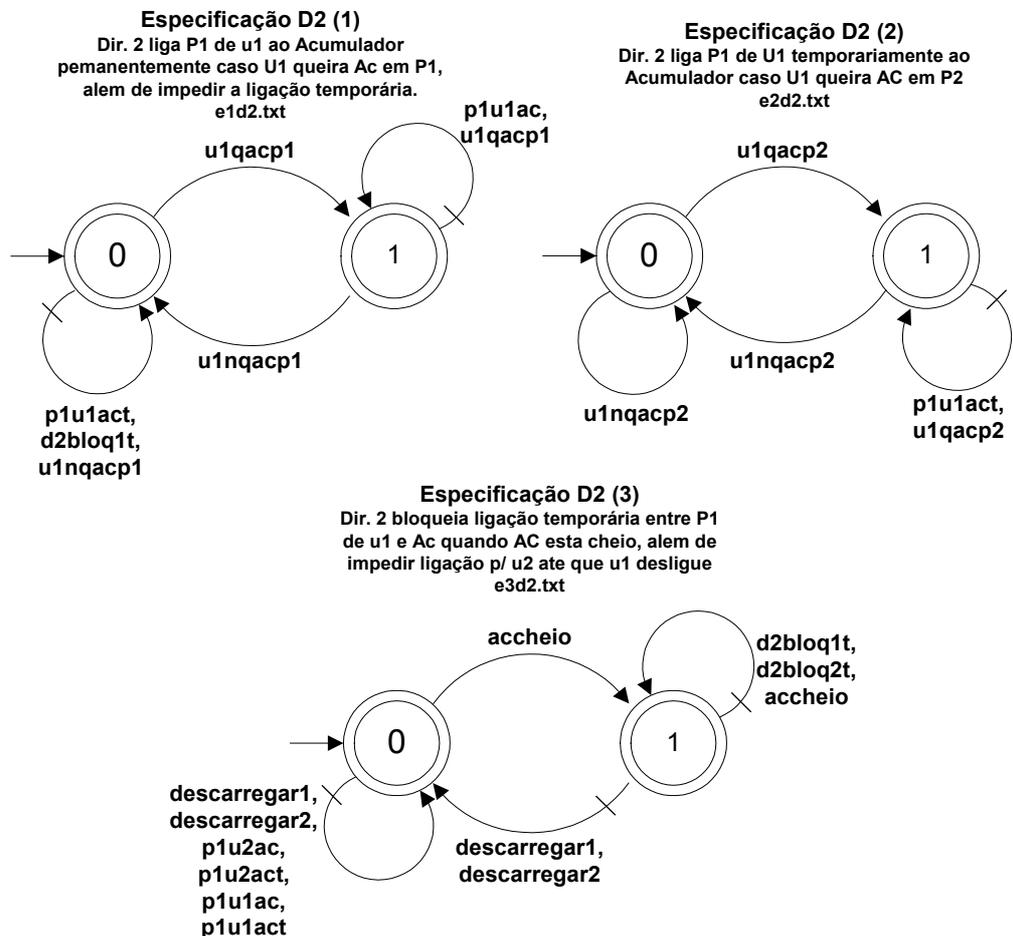


Figura 8.16 – Especificações para direcionador da bomba fixa e direcionador 2.

Especificações para a o direcionador 3

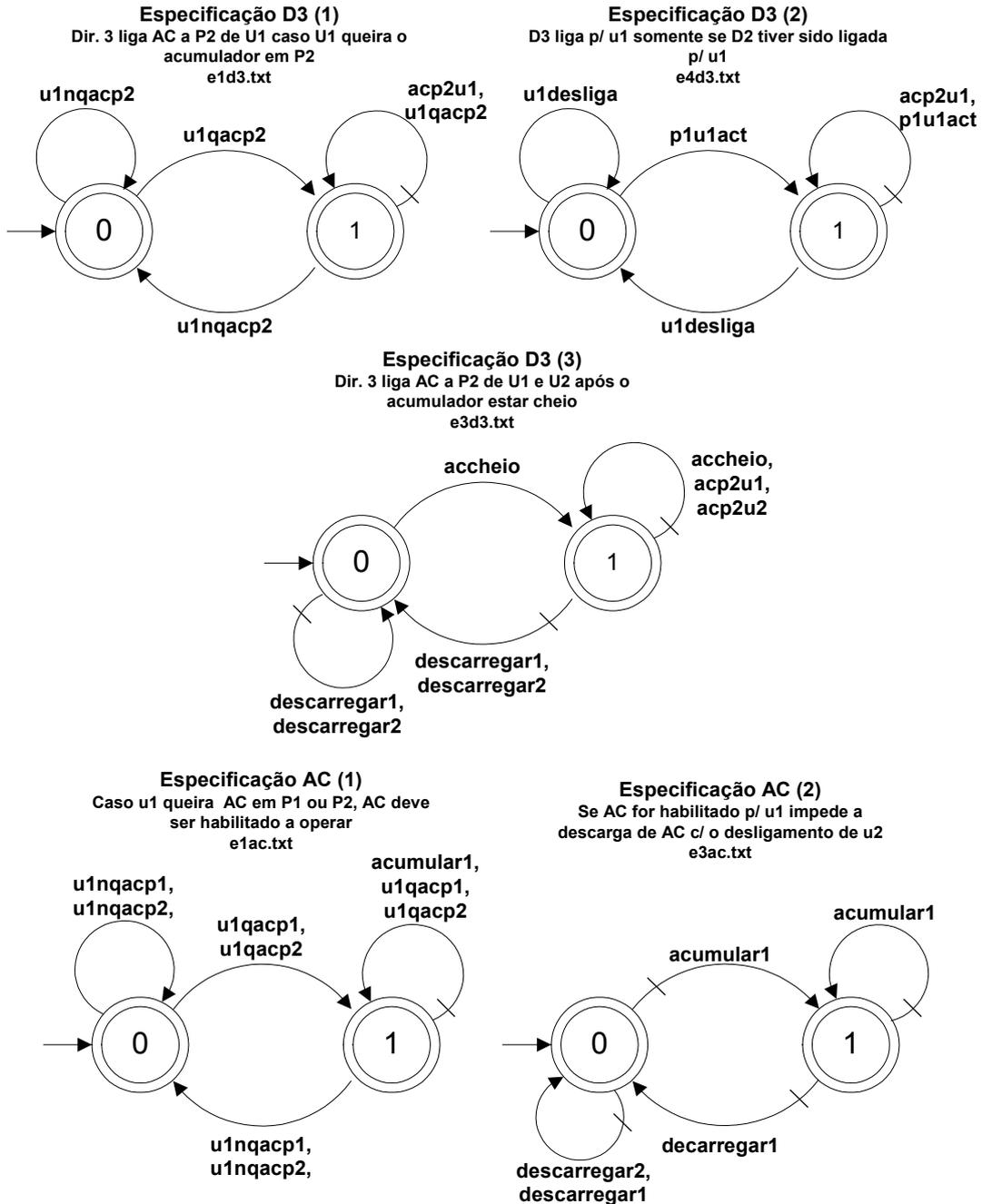


Figura 8.17 - Especificação do direcionador 3 e do acumulador.

8.4 Modelagem da comunicação com o ambiente externo

De maneira similar ao realizado nos itens 7.5.2 e 7.5.3 , apresentar-se-á nos itens seguintes os modelos relacionados aos pedidos e respostas trocados com o ambiente externo ao sistema automático de controle da UPCH. Novamente os modelos dos usuários 1 e 2 (lados 1 e 2 da bancada de trabalho) são simétricos e, desta forma, serão mostrados os modelos relacionados ao usuário 1 para efeito de simplificação.

8.4.1 Modelagem de comportamento relacionada aos pedidos

De acordo com as especificações de projeto, os usuários podem escolher os recursos independentemente, sendo que: as bombas são escolhidas somente para a saída P1 e o acumulador pode ser escolhido para a saída P1 ou P2. Assim os modelos dos pedidos, no caso do usuário 1, ficam como descrito na Figura 8.18.

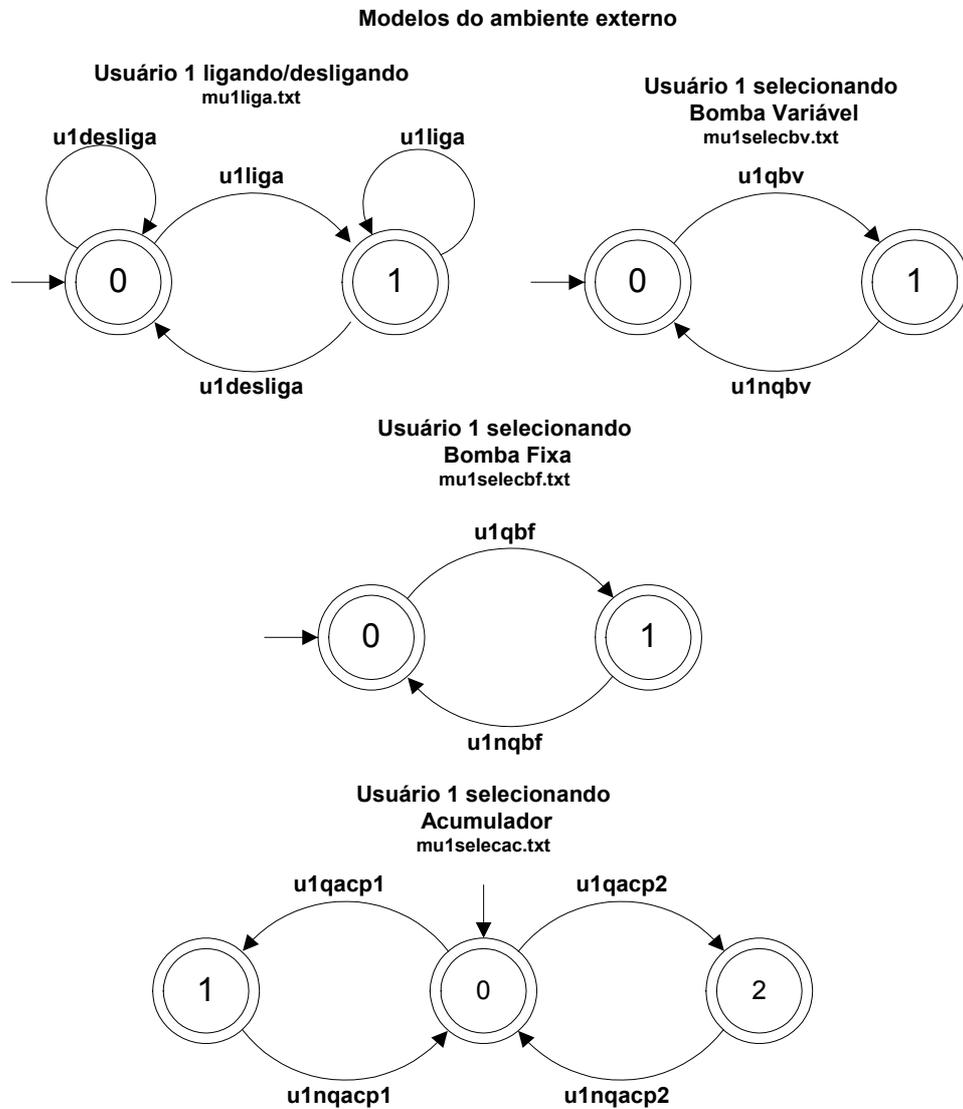


Figura 8.18 - Modelos dos pedidos do usuário 1.

A Tabela 12 mostra o significado dos eventos relacionados aos modelos de pedido.

Tabela 12 - Eventos dos modelos de pedido.

Evento	Ação
u1liga	Comando de ligar do usuário 1
u1desliga	Comando de desligar do usuário 1
u1qbv	Usuário 1 querendo a bomba variável
u1nqbv	Usuário 1 não quer mais a bomba variável

u1qbf	Usuário 1 querendo a bomba fixa
u1nqbf	Usuário 1 não quer mais a bomba fixa
u1qacp1	Usuário 1 querendo acumulador em P1
u1nqacp1	Usuário 1 não quer mais acumulador em P1
u1qacp2	Usuário 1 querendo acumulador em P2
u1nqacp1	Usuário 1 não quer mais acumulador em P2

8.4.2 Modelagem de comportamento relacionada às respostas

Embora não exista a necessidade de informar da maneira diferente a não existência de pedido e o pedido correto (ver item 5.4.2), o que caracteriza a utilização de modelos de respostas com dois estados, escolheu-se para este caso uma implementação com dois e três estados para efeito de experimentação do modelo proposto. O sistema automático em questão é composto de sistemas de pedido (SP), resposta (SR) e pedido-resposta (SPR)

Os modelos mostrados na Figura 8.19 são relativos a sistemas de pedido-resposta (SPR).

Modelos das respostas aos usuários

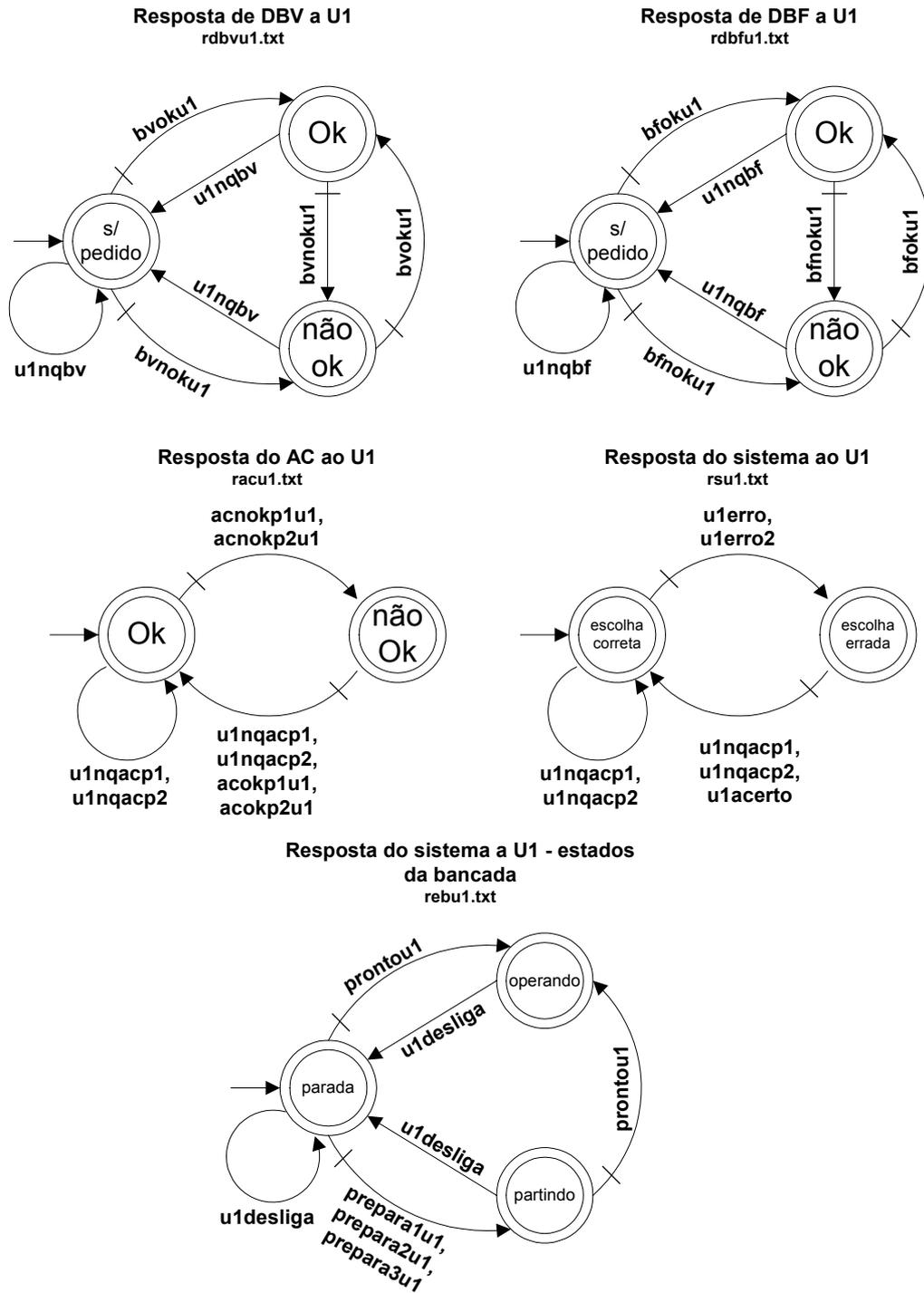


Figura 8.19 - Modelos das respostas relacionadas aos pedidos do usuário 1.

Os modelos de resposta descritos na Figura 8.20 não são diretamente relacionados aos pedidos ou mesmo não são relacionados aos pedidos, caracterizando os sistemas de resposta (SR).

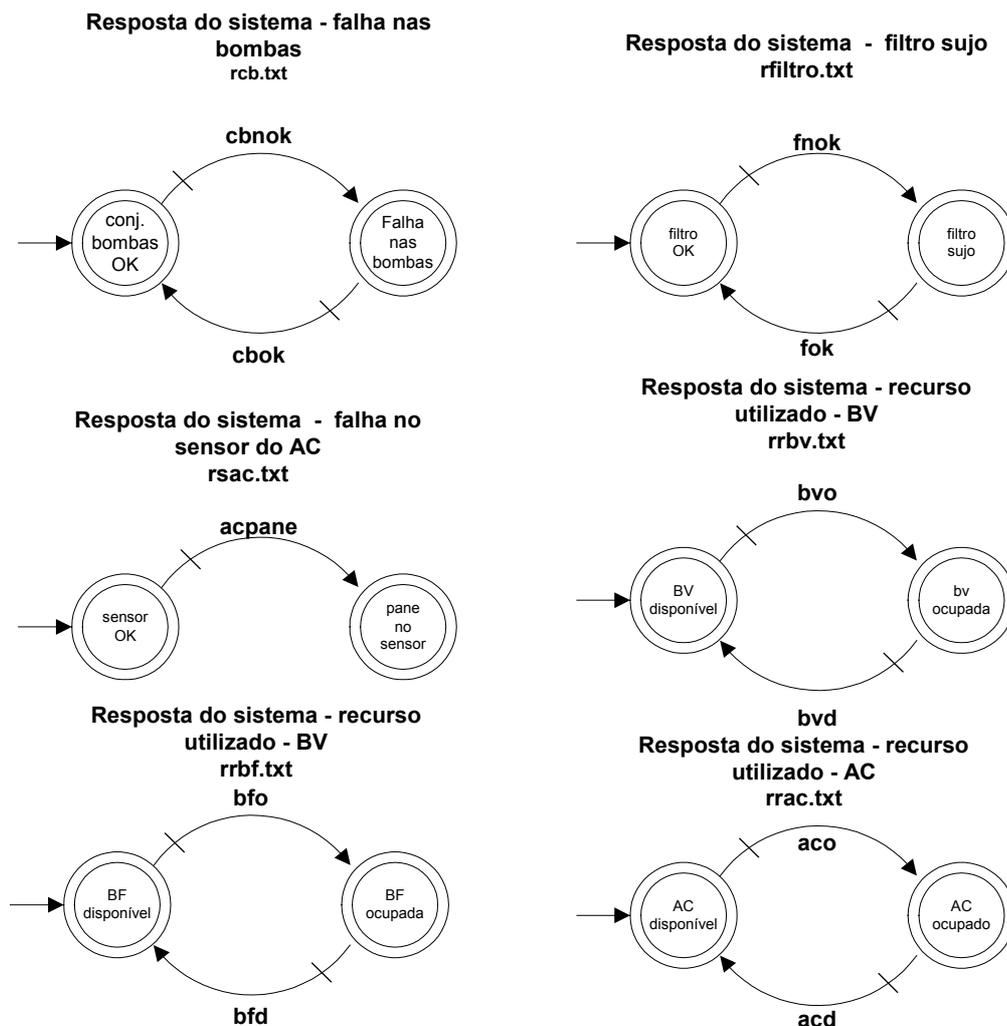


Figura 8.20 - Modelos de respostas de erros críticos e recursos utilizados.

Os modelos de respostas mostrados não foram concebidos de imediato na forma em que estão apresentados. Como foi comentado, o processo de projeto é recursivo e assim os modelos foram sendo modificados juntamente com as especificações mostradas mais adiante, de modo a tornar o controle adequado às especificações de projeto. Mais uma vez, alguns modelos são desnecessários, como os de respostas sobre os recursos utilizados, já que poderia ser realizada uma implementação mista como a vista no item 7.5.1 , entretanto escolheu-se esta implementação para teste do modelo proposto.

A Tabela 13 mostra o significado dos eventos dos modelos de respostas.

Tabela 13 - Eventos do modelos de resposta.

Eventos	Ação
bvoku1	Bomba variável OK para usuário 1
bvnoku1	Bomba variável não OK para usuário 1
bfoku1	Bomba fixa OK para usuário 1
bfnoku1	Bomba fixa não OK para usuário 1
acokp1u1	Acumulador OK em P1 do usuário 1

acnokp1u1	Acumulador não OK em P1 do usuário 1
Acokp2u1	Acumulador OK em P2 do usuário 1
Acnokp2u1	Acumulador não OK em P2 do usuário 1
u1erro	Usuário 1 escolheu AC em P1 sem bombas
u1erro2	U1 escolheu AC sem bombas disponíveis
u1acerto	A escolha de U1 está correta
prontou1	Bancada pronta para operar
prepara1u1	Aguardar liberação da bomba variável
prepara2u1	Aguardar liberação da bomba fixa
prepara3u1	Aguardar o sinal de acumulador cheio
cbnok	Conjunto de bombas desligado
cbok	Conjunto de bombas ligado
fok	Filtro OK
fnok	Filtro sujo
acpane	Pane no sensor do acumulador
bvo	Bomba variável ocupada
bvd	Bomba variável disponível
bfo	Bomba fixa ocupada
bfd	Bomba fixa disponível
aco	Acumulador ocupado
acd	Acumulador disponível

8.4.3 Especificação do comportamento controlado das respostas

As especificações para os modelos das respostas dão origem a supervisores que observam os pedidos do usuário e os estados da planta, permitindo uma resposta adequada dentro das especificações de projeto. As figuras abaixo, Figura 8.21, Figura 8.22, Figura 8.23, Figura 8.24, Figura 8.25 e Figura 8.26, descrevem as especificações para as respostas.

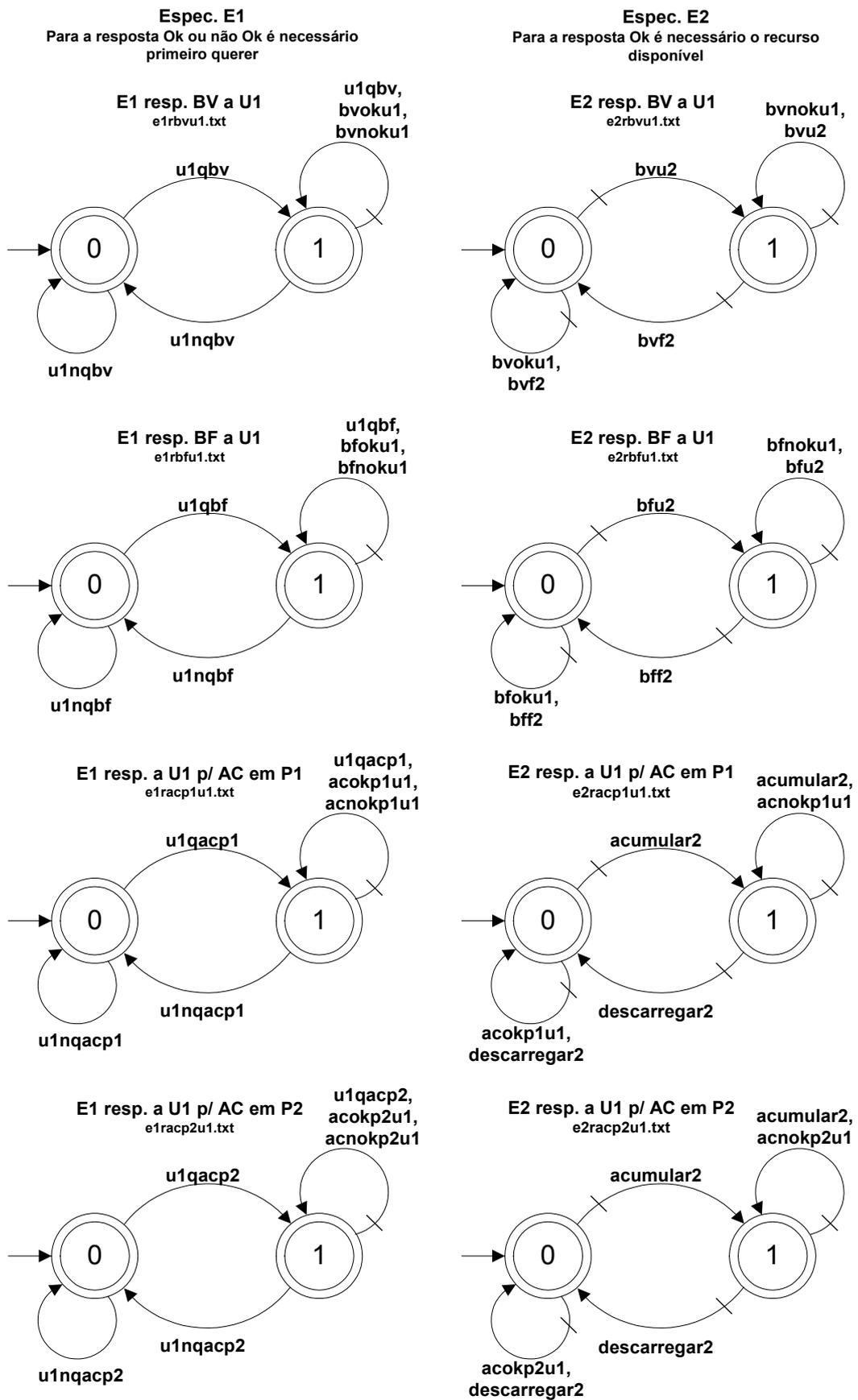


Figura 8.21 - Especificações das respostas de BV, BF e AC.

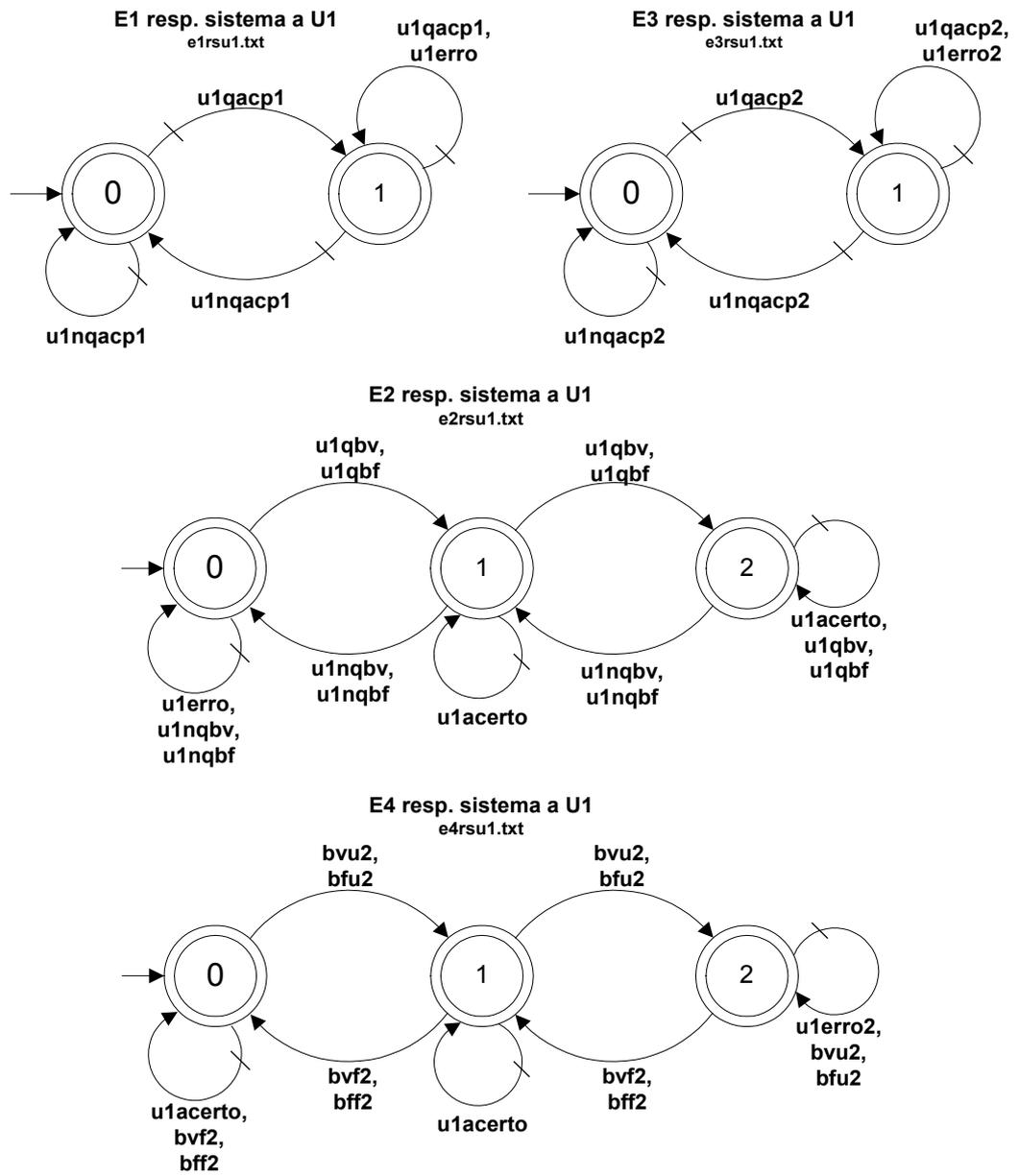


Figura 8.22 - Especificações para as respostas de erro de escolha.

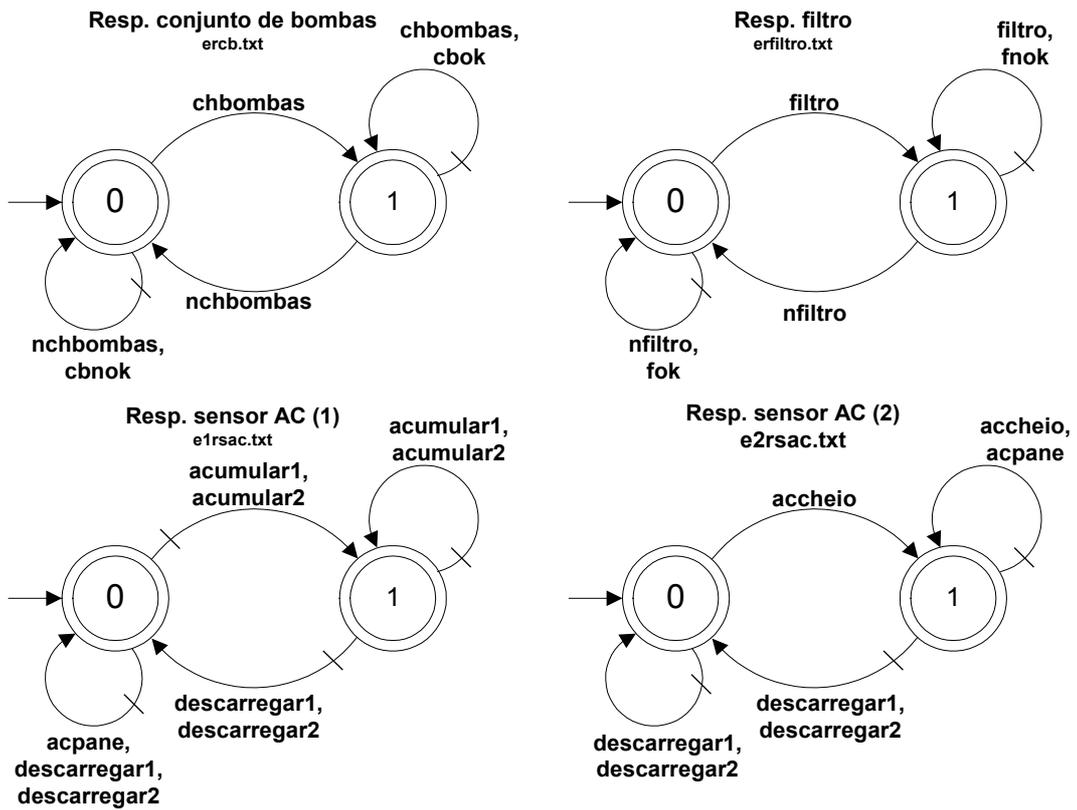


Figura 8.23 - Especificações para a resposta de erros críticos.

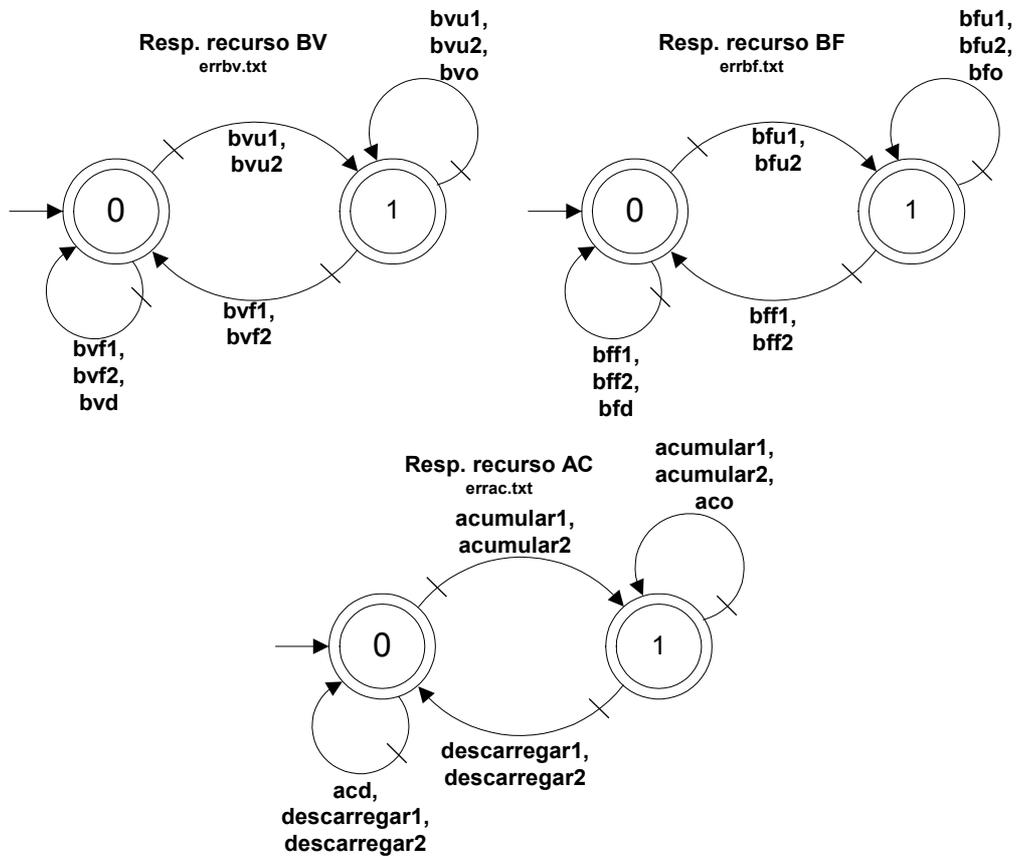


Figura 8.24 - Especificações para as respostas de recursos utilizados.

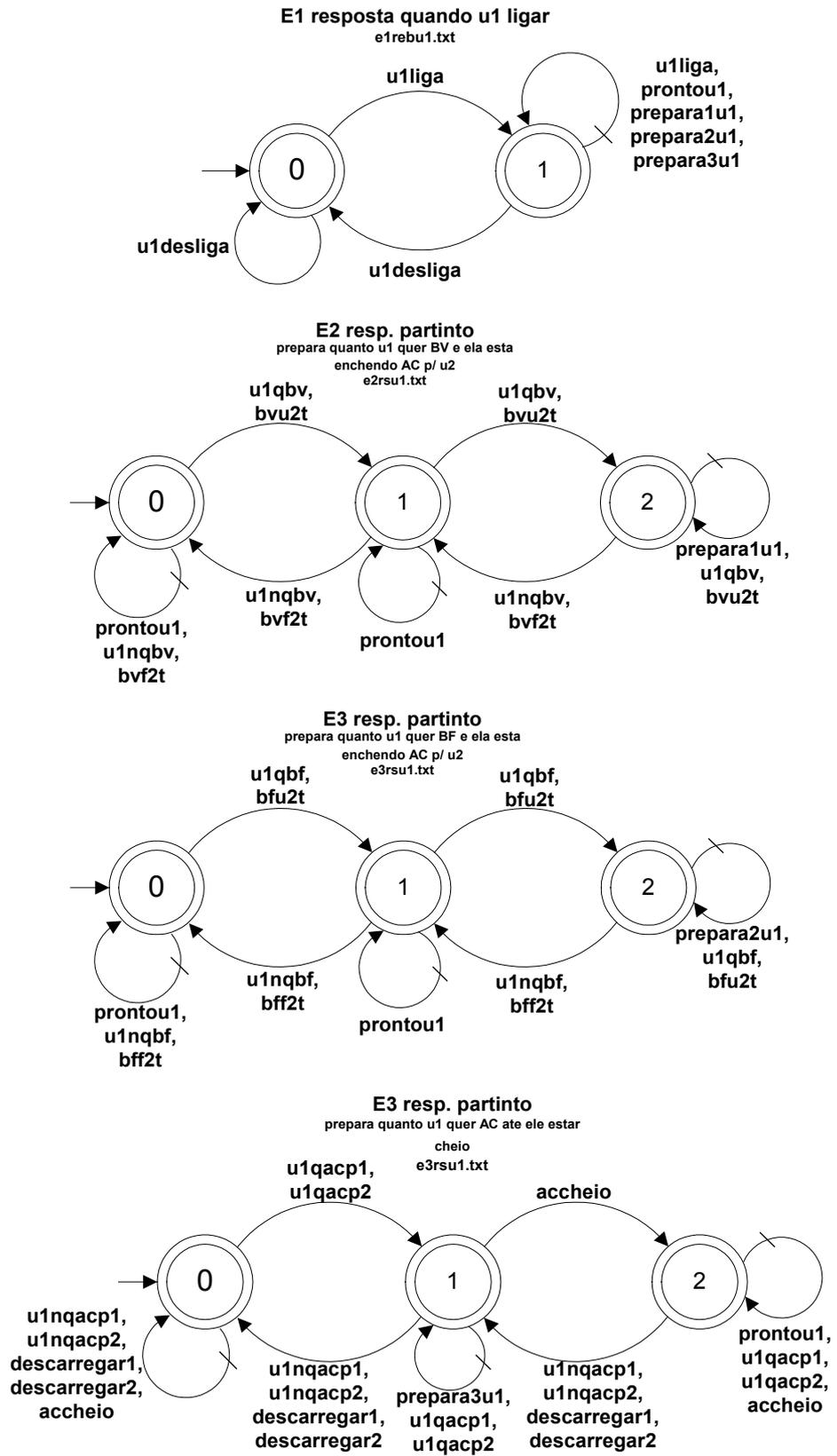


Figura 8.25 - Especificações para as respostas de estados da bancada (parada, partindo, operando).

Para assegurar o controle correto em relação às especificações de projeto, são necessárias algumas especificações complementares para impedir o acionamento parcial do sistema ou para bloqueá-lo em caso de falhas.

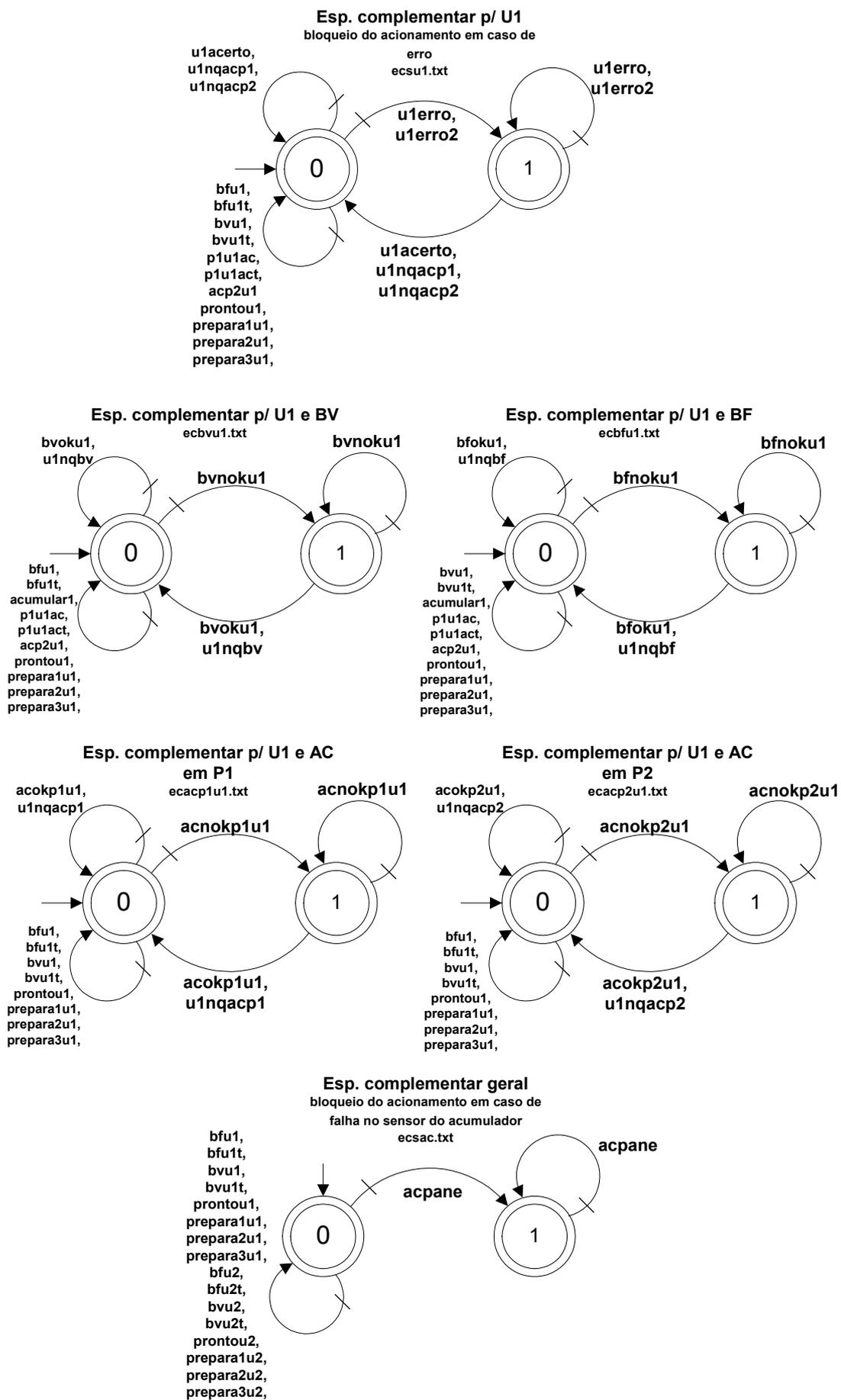


Figura 8.26 - Especificações complementares.

8.5 Cálculo do controlador

Para evitar problemas de limites computacionais, alguns modelos e especificações foram divididos de modo a não resultarem em supervisores modulares muito grandes e inviabilizarem a aplicação. Esta manobra permitiu o cálculo dos supervisores com sucesso mas, no entanto, provocou um número excessivo de supervisores modulares, fato que também não comprometeu a aplicação devido a redução realizada nos supervisores.

8.5.1 Gerar arquivos de modelos e especificações

Os arquivos gerados para o cálculo dos supervisores modulares podem ser observados na Tabela 14.

Tabela 14 - Lista de arquivos Grail gerados para o cálculo do controlador.

Grupo de arquivos	Elementos	Quantidade
Modelos do sistema energético material	mac.txt, mbombas.txt, md2.txt, md3.txt, mdbf.txt, mdbv.txt, mfiltro.txt.	7
Modelos dos sistemas de pedido	mu1liga.txt, mu1selecac.txt, mu1selecbf.txt, mu1selecbv.txt, mu2liga.txt, mu2selecac.txt, mu2selecbf.txt, mu2selecbv.txt.	8
Modelos dos sistemas de resposta	racu1.txt, racu2.txt, rcb.txt, rdbfu1.txt, rdbfu2.txt, rdbvu1.txt, rdbvu2.txt, rebu1.txt, rebu2.txt, result.txt, result1.txt, rfiltro.txt, rrac.txt, rrbf.txt, rrbv.txt, rsac.txt, rsu1.txt, rsu2.txt.	18
Especificações	e11csu1.txt, e11csu2.txt, e12csu1.txt, e12csu2.txt, e13csu1.txt, e13csu2.txt, e1ac.txt, e1cACP1u1.txt, e1cACP1u2.txt, e1cACP2u1.txt, e1cACP2u2.txt, e1cbfu1.txt, e1cbfu2.txt, e1cbvu1.txt, e1cbvu2.txt, e1d2.txt, e1d3.txt, e1racP1u1.txt, e1racP1u2.txt, e1racP2u1.txt, e1racP2u2.txt, e1rbfu1.txt, e1rbfu2.txt, e1rbvu1.txt, e1rbvu2.txt, e1rebu1.txt, e1rebu2.txt, e1rsac.txt, e1rsu1.txt, e1rsu2.txt, e21cACP1u1.txt, e21cACP1u2.txt, e21cACP2u1.txt, e21cACP2u2.txt, e21cbfu1.txt, e21cbfu2.txt, e21cbvu1.txt, e21cbvu2.txt, e21csu1.txt, e21csu2.txt, e22cACP1u1.txt, e22cACP1u2.txt, e22cACP2u1.txt,	113

	e22cACP2u2.txt, e22cbfu1.txt, e22cbfu2.txt, e22cbvu1.txt, e22cbvu2.txt, e22csu1.txt, e22csu2.txt, e23cbfu1.txt, e23cbfu2.txt, e23cbvu1.txt, e23cbvu2.txt, e23csu1.txt, e23csu2.txt, e2ac.txt, e2d2.txt, e2d3.txt, e2racp1u1.txt, e2racp1u2.txt, e2racp2u1.txt, e2racp2u2.txt, e2rbfu1.txt, e2rbfu2.txt, e2rbvu1.txt, e2rbvu2.txt, e2rebu1.txt, e2rebu2.txt, e2rsac.txt, e2rsu1.txt, e2rsu2.txt, e3ac.txt, e3d2.txt, e3d3.txt, e3rebu1.txt, e3rebu2.txt, e3rsu1.txt, e3rsu2.txt, e4ac.txt, e4d3.txt, e4rebu1.txt, e4rebu2.txt, e4rsu1.txt, e4rsu2.txt, e5d2.txt, e5d3.txt, e6d2.txt, eCSAcu1.txt, eCSAcu2.txt, edbf1.txt, edbf2.txt, edbf3.txt, edbf5.txt, edbf6.txt, edbf7.txt, edbf8.txt, edbv1.txt, edbv2.txt, edbv3.txt, edbv5.txt, edbv6.txt, egeral2u1.txt, egeral2u2.txt, egeralu1.txt, egeralu2.txt, eplanta1.txt, eplanta2.txt, ercb.txt, erfiltro.txt, errac.txt, errbf.txt, errbv.txt,	
Total de arquivos		146

8.5.2 Obter representação por sistema produto

A representação por sistema produto do sistema em questão fica como mostrado na Tabela 15.

Tabela 15 - Representação por sistema produto dos modelos do sistema da UPCH.

Sistema Produto	Modelos relacionados
Módulo 1 – MSP1	mbombas.txt
Módulo 2 – MSP2	mfiltro.txt
Módulo 3 – MSP3	mdbv.txt
Módulo 4 – MSP4	mdbf.txt
Módulo 5 – MSP5	md2.txt
Módulo 6 – MSP6	md3.txt
Módulo 7 – MSP7	mac.txt
Módulo 8 – MSP8	mu1liga.txt rebu1.txt
Módulo 9 – MSP9	mu2liga.txt rebu2.txt

Módulo 10 – MSP10	mu1selecbv.txt rdbvu1.txt
Módulo 11 – MSP11	mu2selecbv.txt rdbvu2.txt
Módulo 12 – MSP12	mu1selecbf.txt rdbfu1.txt
Módulo 13 – MSP13	mu2selecbf.txt rdbfu2.txt
Módulo 14 – MSP14	mu1selecac.txt racu1.txt rsu1.txt
Módulo 15 – MSP15	mu2selecac.txt racu2.txt rsu2.txt
Módulo 16 – MSP16	rcb.txt
Módulo 17 – MSP17	rfiltro.txt
Módulo 18 – MSP18	rsac.txt
Módulo 19 – MSP19	rrbv.txt
Módulo 20 – MSP20	rrbf.txt
Módulo 21 – MSP21	rrac.txt

8.5.3 Calcular e reduzir os supervisores modulares

O cálculo dos supervisores modulares foi realizado de maneira similar ao relatado no item 7.6.3 , utilizando-se para tal os softwares Grail, CTCT e o pacote de conversão entre eles. Os resultados estão descritos na Tabela 16.

Tabela 16 - Resultado dos supervisores locais e supervisores locais reduzidos da UPCH.

Especificações relacionadas	Supervisores modulares		Supervisores modulares reduzidos	
	Supervisor	(Nº de estados, Nº de transições)	Supervisor	(Nº de estados, Nº de transições)
e11csu1.txt	Sup1	(36,258)	Supr1	(2, 33)
e11csu2.txt	Sup2	(36,258)	Supr2	(2, 33)
e12csu1.txt	Sup3	(36,258)	Supr3	(2, 33)
e12csu2.txt	Sup4	(36,216)	Supr4	(2, 33)
e13csu1.txt	Sup5	(36,276)	Supr5	(2, 28)
e13csu2.txt	Sup6	(72,552)	Supr6	(2, 28)
e1ac.txt	Sup7	(36,278)	Supr7	(2, 27)
e1cacp1u1.txt	Sup8	(72,516)	Supr8	(2, 36)
e1cacp1u2.txt	Sup9	(72,516)	Supr9	(2, 36)
e1cacp2u1.txt	Sup10	(72,516)	Supr10	(2, 36)
e1cacp2u2.txt	Sup11	(72,516)	Supr11	(2, 36)
e1cbfu1.txt	Sup12	(54,414)	Supr12	(2, 30)
e1cbfu2.txt	Sup13	(54,414)	Supr13	(2, 30)
e1cbvu1.txt	Sup14	(54,414)	Supr14	(2, 30)
e1cbvu2.txt	Sup15	(54,414)	Supr15	(2, 30)
e1d2.txt	Sup16	(36,254)	Supr16	(2, 31)
e1d3.txt	Sup17	(36,214)	Supr17	(2, 29)
e1racp1u1.txt	Sup18	(12,50)	Supr18	(2, 16)
e1racp1u2.txt	Sup19	(12,50)	Supr19	(2, 16)

e1racp2u1.txt	Sup20	(12,50)	Supr20	(2, 16)
e1racp2u2.txt	Sup21	(12,50)	Supr21	(2, 16)
e1rbfu1.txt	Sup22	(4,8)	Supr22	(2, 4)
e1rbfu2.txt	Sup23	(4,8)	Supr23	(2, 4)
e1rbvu1.txt	Sup24	(4,8)	Supr24	(2, 4)
e1rbvu2.txt	Sup25	(4,8)	Supr25	(2, 4)
e1rebu1.txt	Sup26	(4,13)	Supr26	(2, 8)
e1rebu2.txt	Sup27	(4,13)	Supr27	(2, 8)
e1rsac.txt	Sup28	(6,19)	Supr28	(2, 7)
e1rsu1.txt	Sup29	(12,54)	Supr29	(2, 17)
e1rsu2.txt	Sup30	(12,54)	Supr30	(2, 17)
e21cacp1u1.txt	Sup31	(72,516)	Supr31	(2, 30)
e21cacp1u2.txt	Sup32	(72,516)	Supr32	(2, 30)
e21cacp2u1.txt	Sup33	(72,516)	Supr33	(2, 30)
e21cacp2u2.txt	Sup34	(72,516)	Supr34	(2, 30)
e21cbfu1.txt	Sup35	(18,86)	Supr35	(2, 21)
e21cbfu2.txt	Sup36	(18,86)	Supr36	(2, 21)
e21cbvu1.txt	Sup37	(18,86)	Supr37	(2, 21)
e21cbvu2.txt	Sup38	(18,86)	Supr38	(2, 21)
e21csu1.txt	Sup39	(36,258)	Supr39	(2, 33)
e21csu2.txt	Sup40	(36,258)	Supr40	(2, 33)
e22cacp1u1.txt	Sup41	(144,1104)	Supr41	(2,30)
e22cacp1u2.txt	Sup42	(144,1104)	Supr42	(2, 30)
e22cacp2u1.txt	Sup43	(144,1104)	Supr43	(2, 30)
e22cacp2u2.txt	Sup44	(144,1104)	Supr44	(2, 30)
e22cbfu1.txt	Sup45	(18,64)	Supr45	(2, 14)
e22cbfu2.txt	Sup46	(18,64)	Supr46	(2, 14)
e22cbvu1.txt	Sup47	(18,64)	Supr47	(2, 14)
e22cbvu2.txt	Sup48	(18,64)	Supr48	(2, 14)
e22csu1.txt	Sup49	(36,216)	Supr49	(2, 26)
e22csu2.txt	Sup50	(36,216)	Supr50	(2, 26)
e23cbfu1.txt	Sup51	(36,196)	Supr51	(2, 15)
e23cbfu2.txt	Sup52	(36,196)	Supr52	(2, 15)
e23cbvu1.txt	Sup53	(36,196)	Supr53	(2, 15)
e23cbvu2.txt	Sup54	(36,196)	Supr54	(2, 15)
e23csu1.txt	Sup55	(72,552)	Supr55	(2, 27)
e23csu2.txt	Sup56	(72,552)	Supr56	(2, 27)
e2ac.txt	Sup57	(36,278)	Supr57	(2, 27)
e2d2.txt	Sup58	(36,262)	Supr58	(2, 33)
e2d3.txt	Sup59	(36,214)	Supr59	(2, 29)
e2racp1u1.txt	Sup60	(72,528)	Supr60	(2, 30)
e2racp1u2.txt	Sup61	(72,528)	Supr61	(2, 30)
e2racp2u1.txt	Sup62	(72,528)	Supr62	(2, 30)
e2racp2u2.txt	Sup63	(72,528)	Supr63	(2, 30)
e2rbfu1.txt	Sup64	(36,156)	Supr64	(2, 22)
e2rbfu2.txt	Sup65	(36,156)	Supr65	(2, 22)
e2rbvu1.txt	Sup66	(36,156)	Supr66	(2, 22)
e2rbvu2.txt	Sup67	(36,156)	Supr67	(2, 22)
e2rebu1.txt	Sup68	(324,2664)	Supr68	(3, 51)
e2rebu2.txt	Sup69	(324,2664)	Supr69	(3, 51)

e2rsac.txt	Sup70	(10,33)	Supr70	(2, 11)
e2rsu1.txt	Sup71	(432,3888)	Supr71	(3, 50)
e2rsu2.txt	Sup72	(432,3888)	Supr72	(3, 50)
e3ac.txt	Sup73	(5,13)	Supr73	(2, 7)
e3d2.txt	Sup74	(15,69)	Supr74	(2, 20)
e3d3.txt	Sup75	(15,61)	Supr75	(2, 16)
e3rebu1.txt	Sup76	(324,2664)	Supr76	(3, 51)
e3rebu2.txt	Sup77	(324,2664)	Supr77	(3, 51)
e3rsu1.txt	Sup78	(12,54)	Supr78	(2, 17)
e3rsu2.txt	Sup79	(12,54)	Supr79	(2, 17)
e4ac.txt	Sup80	(5,13)	Supr80	(2, 7)
e4d3.txt	Sup81	(108,810)	Supr81	(2, 35)
e4rebu1.txt	Sup82	(648,7164)	Supr82	(3, 63)
e4rebu2.txt	Sup83	(648,7164)	Supr83	(3, 63)
e4rsu1.txt	Sup84	(324,3132)	Supr84	(3, 78)
e4rsu2.txt	Sup85	(324,3132)	Supr85	(3, 78)
e5d2.txt	Sup86	(36,254)	Supr86	(2, 31)
e5d3.txt	Sup87	(108,810)	Supr87	(2, 35)
e6d2.txt	Sup88	(36,262)	Supr88	(2, 33)
ecsacu1.txt	Sup89	(108,864)	Supr89	(2, 37)
ecsacu2.txt	Sup90	(108,864)	Supr90	(2, 37)
edbf1.txt	Sup91	(18,81)	Supr91	(2, 19)
edbf2.txt	Sup92	(36,262)	Supr92	(2, 33)
edbf3.txt	Sup93	(15,75)	Supr93	(2, 22)
edbf5.txt	Sup94	(18,81)	Supr94	(2, 19)
edbf6.txt	Sup95	(36,262)	Supr95	(2, 33)
edbf7.txt	Sup96	(18,93)	Supr96	(2, 31)
edbf8.txt	Sup97	(18,93)	Supr97	(2, 31)
edbv1.txt	Sup98	(18, 81)	Supr98	(2, 19)
edbv2.txt	Sup99	(36, 262)	Supr99	(2, 33)
edbv3.txt	Sup100	(15, 75)	Supr100	(2, 22)
edbv5.txt	Sup101	(18, 81)	Supr101	(2, 19)
edbv6.txt	Sup102	(36, 262)	Supr102	(2, 33)
egeral2u1.txt	Sup103	(162, 1539)	Supr103	(2, 40)
egeral2u2.txt	Sup104	(162, 1539)	Supr104	(2, 40)
egeralu1.txt	Sup105	(54, 432)	Supr105	(2, 38)
egeralu2.txt	Sup106	(54, 432)	Supr106	(2, 38)
eplanta1.txt	Sup107	(27, 177)	Supr107	(2, 24)
eplanta2.txt	Sup108	(27, 165)	Supr108	(2, 26)
ercb.txt	Sup109	(4,6)	Supr109	(2, 4)
erfiltro.txt	Sup110	(4,6)	Supr110	(2, 6)
errac.txt	Sup111	(6,21)	Supr111	(2, 18)
errbf.txt	Sup112	(12,38)	Supr112	(2, 18)
errbv.txt	Sup113	(12,38)	Supr113	(2, 12)

8.6 Implementação em CLP

Como no item 7.7 , os resultados do projeto do controle foram implementados no CLP escolhido para o controle da UPCH e que está instalado na bancada de simulação.

Mais uma vez, utilizou-se o software CGLI para traduzir os arquivos Grail gerados no processo de cálculo dos supervisores modulares (os arquivos referentes ao sistema produto e aos supervisores modulares reduzidos), sendo as seqüências operacionais e interfaces de comunicação implementadas manualmente.

Como citado no item 6.9 , foi necessária a implementação manual de variáveis de controle da evolução de alguns supervisores modulares, já que esta funcionalidade ainda não está implementada na versão atual do CGLI. No entanto esta tarefa não representou grande problema visto que se trata de apenas dez supervisores dos 113 que compõem o controlador.

8.7 Resultados e conclusões

8.7.1 Resultados

O sistema implementado na bancada de simulação foi confrontado com as especificações de projeto e respondeu de maneira satisfatória, realizando todas as tarefas corretamente e proporcionando uma comunicação eficiente através do painel simulado.

Embora o sistema seja composto por 33 módulos do sistema produto e 113 supervisores modulares locais, a depuração, avaliação e correção de não conformidades durante a implementação e os teste foi realizada sem problemas, visto que a pesquisa feita sobre as linhas de código é fácil e o encadeamento dos elementos do programa é também de fácil compreensão. Importante ressaltar que, mesmo com o grande número de supervisores modulares, a utilização de um algoritmo de redução permitiu que o programa final fosse bastante pequeno e, conseqüentemente, com um ciclo de varredura bastante rápido.

Notou-se durante o projeto que a modificação das especificações e/ou inclusão e exclusão destas não impõe grande gasto de tempo ou dificuldades de projeto, no entanto causa um gasto razoável de tempo computacional para recalcular os supervisores.

8.7.2 Conclusões

A utilização da bancada de teste e simulação, originalmente concebida para um projeto paralelo que acabou colaborando com o presente trabalho, tornou possível a comparação das metodologias de projeto combinatório em Álgebra Booleana e a Teoria de Controle Supervisório Modular Local (TCSML), que resultou no trabalho "SILVA NETO, F. A. C.; SOUTO, R. B.; DE NEGRI, V. J.. Comparação entre métodos de projeto de controle de sistemas a eventos discretos e sua implementação em CLP. **Anais do XI CREEM** - UERJ, Nova Friburgo, 2004".

Notou-se, então, que as técnicas de projeto de SED tem vantagens e desvantagens marcantes, as quais podemos citar algumas para os casos de Álgebra de Boole e Teoria de controle supervisório modular local (TCSML) como pode ser visto na Tabela 17.

Tabela 17 - Comparação entre métodos de projeto de SED - Álgebra de Boole e TCSML.

Complexidade do sistema	Álgebra Booleana		TCSML	
	Pouco complexo	Muito complexo	Pouco complexo	Muito complexo
Facilidade de utilização do método	Fácil ↑	Difícil ↓	Médio	Médio
Clareza do programa resultante	Média	Baixa ↓	Alta ↑	Média
Facilidade de criação do código em CLP	Fácil ↑	Difícil ↓	Fácil ↑	Fácil ↑
Demanda por memória do CLP	Pequeno ↑	Médio	Médio	Grande ↓
Facilidade de manutenção	Média	Difícil ↓	Fácil ↑	Média
Robustez	Alta ↑	Baixa ↓	Alta ↑	Alta ↑
Tempo de desenvolvimento	Pequeno ↑	Grande ↓	médio	Grande ↓

Obs: As características mais positivas estão marcadas com o símbolo '↑' e as mais negativas com '↓.'

Embora existam métodos numéricos para calcular expressões booleanas sobre tabelas-verdade com grande número de variáveis, estas tabelas são às vezes compostas de milhares de linhas e isso torna o projeto e alteração muito complexo e trabalhoso para o caso de sistemas mais intrincados. Já o projeto utilizando TCSML tem a vantagem de ser de melhor compreensão e de fácil modificação, no entanto demanda grande quantidade de memória, para a implementação, e tempo de projeto.

O grande tempo de projeto é necessário para a fase de modelagem e especificação do sistema que, além de exigir o conhecimento por parte do projetista da teoria de linguagens e autômatos e controle supervisão, ainda não possui uma ferramenta e interface de desenvolvimento agradável às atividades de engenharia de projeto.

Percebeu-se, durante a execução dos testes, que um grande problema da teoria de controle supervisão, que é a demanda por esforço e memória computacional durante o processo de cálculo do supervisor e planta do sistema devido à explosão de estados característica da técnica, é drasticamente atenuado com a utilização da TCSML. Durante os cálculos deste trabalho não foram notados grandes esforços e tempos de computação para calcular os supervisores locais, entretanto, dois problemas ainda persistem e neste caso provocaram um considerável gasto de tempo computacional.

O primeiro problema é a necessidade de realizar a composição de todos os supervisores locais, obtendo o supervisor monolítico, para testar a modularidade entre eles, o que gasta muito tempo na composição e também no teste. O segundo é a redução dos supervisores, cujo processo tem o tempo razoavelmente grande para supervisores modulares também grandes. Como exemplo, um supervisor de 400 estados demorou uma hora para ser reduzido em um computador com 512 megabytes de memória RAM e 2GHz de frequência

de operação e um supervisor de 640 estados consumiu mais de três horas e meia de cálculo (calculado no XPTCT86 - WONHAM, 2004).

As ferramentas utilizadas são acadêmicas (não comerciais) e embora sejam bastante funcionais não têm uma interface de trabalho muito amigável (são executadas em janelas de *prompt* de comando do MS-DOS), e em muitas atividades acabam exigindo bastante trabalho braçal do projetista.

De um modo geral, o modelo para a comunicação com o ambiente externo proporcionou um melhor entendimento e organização do projeto quanto a este aspecto, mostrando também adequado às técnicas e ferramentas existentes. Outro aspecto importante deste trabalho foi a aplicação das técnicas e ferramentas de projeto de sistemas automáticos a um sistema diferente de um SMMA (SANTOS, 2003), mostrando que elas são viáveis a outras classes de SED.

Deve-se ressaltar que os modelos estruturais e funcionais mostrados aqui não são exclusivos dos sistemas a eventos discretos e sim são relacionados aos sistemas automáticos em geral. A exemplo disto, pode-se citar o trabalho de SOUZA (2005), que representa um sistema de controle de hidráulica proporcional, que utiliza um PC, condicionadores de sinal e placas de aquisição de dados como *hardware* do sistema de informação, utilizando o modelo em rede C/A com comunicação com o ambiente externo introduzido neste trabalho.

CAPÍTULO 9

CONCLUSÕES

No decorrer deste documento, viu-se a importância de se utilizar métodos e ferramentas adequadas para o projeto de sistemas automáticos para se obter resultados com maior rapidez e qualidade. Foi realizada uma revisão do conhecimento existente sobre o assunto no sentido de se obter uma metodologia de projeto totalmente adequada aos casos de sistemas automáticos. Pôde ser observado que muitos esforços já foram realizados para amadurecer técnicas e ferramentas de modo a facilitar as atividades de projeto e promover a integração de áreas de conhecimento distintas.

Baseado no panorama exposto, o presente trabalho buscou contribuir para o esforço coletivo de pesquisadores diversos, abordando aplicações reais onde as soluções ainda estavam indefinidas ou inadequadas. Assim, realizou-se a proposta de aprimoramento em aspectos importantes do processo de projeto, na modelagem e controle, além dos meios de implementação destas propostas.

Estas propostas, que são basicamente um incremento nos modelos existentes, são bastante coerentes e sólidas visto que se baseiam integralmente nos modelos estabelecidos, lançando mão de analogias com as estruturas tradicionais para justificar as novas estruturas propostas. Este incremento comporta um conceito introduzido nesta pesquisa para o caso de comunicação do ambiente externo com o sistema automático, a modelagem da comunicação com o usuário. Embora não seja difícil associar os modelos relacionados com os usuários aos aspectos psicológicos do ser humano, a abstração realizada pelos modelos propostos está relacionada às informações trocadas no processo de comunicação entre usuários e máquinas automáticas, necessárias à operação adequada dos sistemas, e não ao estado emocional destes indivíduos.

Como forma de testar a validade da proposta, foi realizado um projeto e implementação em CLP de um exemplo pequeno mas que contempla todos os aspectos importantes de serem testados para verificar a relevância desta proposta. Uma estrutura de implementação conhecida para os modelos comportamentais foi modificada e utilizada para realizar a programação de modelos e controladores em um CLP comercial, instalado em uma bancada de simulação e testes, permitindo avaliar a viabilidade do conjunto proposto. Esta fase de teste e validação permitiu a elaboração de um procedimento de projeto que auxilia na sistematização das atividades.

Os resultados obtidos foram bastante animadores, visto que se apresentaram como previsto, com todas as funcionalidades dos requisitos de projeto sendo atendidas. Quanto aos aspectos de projeto, pôde ser observado um incremento na facilidade de compreensão do problema como um todo, melhorando a capacidade de inferir soluções e modificações.

O projeto do sistema de controle da UPCH, descrito no capítulo anterior, mostrou que o conhecimento atual sobre projeto de sistemas automáticos já proporciona resultados satisfatórios quanto à solução de problemas reais, no sentido que já se pode obter resultados que atendam às necessidades industriais e não somente as acadêmicas. A bancada didática e a unidade de potência ainda estão sendo construídas e o sistema completo ainda terá outros elementos não considerados aqui, como por exemplo um HMI industrial. Desta maneira, o programa gerado por este trabalho é ainda um protótipo que vai ser incrementado e modificado durante as atividades finais de concepção do equipamento em questão, mas de qualquer maneira os resultados obtidos até este ponto permitem que estas modificações sejam rápidas e fáceis.

A atividade de projeto da UPCH permitiu ainda a comparação entre o projeto com Álgebra Booleana e o projeto com TCSML, em que suas vantagens e desvantagens foram salientadas, contribuindo para o incremento no conhecimento quanto à aplicação destas técnicas.

9.1 Perspectivas

Primeiramente, as propostas realizadas neste trabalho devem ser amplamente testadas, com sistemas de características diversas e com complexidade de comunicação com o ambiente externo bastante variada, o que é tarefa que foge às possibilidades desta pesquisa. O exercício desse tipo de crítica comprobatória certamente vai consolidar os aspectos mais úteis da proposta e adequar à realidade os que, por ventura, não estejam totalmente de acordo com as funcionalidades a que se destinam.

Um aspecto percebido durante esta pesquisa foi a representação parcial do modelo comportamental utilizado, em relação ao modelo estrutural e funcional em rede C/A. Notou-se que, por definição, uma agência pode alterar o estado de suas portas de entrada e saída mas não necessariamente o faz quando da execução da função e esta característica não está bem representada nos modelos comportamentais em autômatos utilizados até agora. À primeira vista, a tarefa de representar esta peculiaridade parece não ser nem um pouco trivial e provavelmente vai exigir um bom esforço de raciocínio e pesquisa, mas que promoveria imensamente a facilidade de se criar modelos padrões para agências que no futuro podem estar integrados a uma ferramenta de *software* de projeto.

A avaliação da decomposição das agências em relação ao esforço de especificação e cálculo de controladores pode ser realizada a fim de se propor uma solução de compromisso, encontrando um nível de refinamento adequado a facilitar do controle. Como foi visto anteriormente, muitas vezes a lei de controle que se quer implementar pode ser realizada através da configuração da estrutura do sistema (via *hardware*), dispensando a utilização de mais supervisores para realizar este trabalho (controle via *software*).

Uma outra sugestão de aprimoramento é a avaliação de outras estruturas de implementação em CLP e outros *hardwares* de controle. A utilização de técnicas que aproximem o projeto teórico da execução prática agiliza o processo e facilita a implementação e portanto mais opções devem ser estudadas e comparadas para que o resultado seja realmente efetivo.

Também se comentou que os modelos em rede C/A não são limitados aos sistemas vistos sob a perspectiva a eventos discretos e, desta forma, um estudo sobre o projeto de sistemas automáticos contínuos certamente enriqueceria muito a base de conhecimento que dará suporte a uma metodologia de projeto.

REFERÊNCIAS BIBLIOGRÁFICAS

AMARAL, D. C. e ROZENFELD, H. – **Gerenciamento do Conhecimento Explícito Sobre o Processo de Desenvolvimento de Produto** - 3º Congresso Brasileiro de Gestão de Desenvolvimento de Produto, Setembro de 2001.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS - ABNT. **NBR 5274**: Símbolos gráficos de eletricidade: Contatos, chaves (interruptores), dispositivos de alarme e de sinalização. Rio de Janeiro, 1973. 7 p.

ATTIÉ, S. S. **Automação hidráulica e pneumática empregando a teoria de sistemas a eventos discretos**. Dissertação (Mestrado em Engenharia Mecânica). Universidade Federal de Santa Catarina, Florianópolis. 1998. 191 p.

BACK, N. **Metodologia de projeto de produtos industriais**. Rio de Janeiro: Guanabara Dois, 1983. 389 p.

BOLLMANN, A. **Fundamentos da automação industrial pneumática**: Projetos de comandos binários eletropneumáticos. São Paulo: ABHP, 1996. 278 p.

BOUZON, G. **Sensores em sistemas a eventos discretos**. Dissertação (Mestrado em Engenharia Elétrica) Universidade Federal de Santa Catarina, Florianópolis, 2004. 90 p.

CARDOSO, J., VALETTE, R. **Redes de Petri**. Florianópolis: Ed. da UFSC, 1997, 212 p.

CASSANDRAS, C. G., LAFORTUNE, S. **Introduction to discrete event systems**. Kluwer Academic Publishers. USA, 1999, 822 p.

CHANDRASEKARAN, B., JOSEPHSON, J. R. Function in device representation. **Engineering with Computers**. London: Springer-Verlag, 2000. pp. 162 – 177.

CONDOOR, S. et al. **A Cognitive Framework for the Design Process**, Design Theory and Methodology, vol. 12, pp. 277-281, 1992.

CURY, J. E. R. **Controle supervísório**: Abordagem Ramadge-Wonham. Apostila do curso de sistemas a eventos discretos. PEGEEL – UFSC, 2003. 73 p.

_____. **Teoria de controle supervísório de sistemas a eventos discretos**. Apostila. Simpósio Brasileiro de Automação Inteligente, 5º, Canela – RS, 2001, 82 p.

CURY, J. E. R., BITTENCOURT, G. **Fundamentos da matemática discreta para controle e automação**. Apostila do curso. PEGEEL – UFSC, 2003.

DA CUNHA, A. E. C., GARCIA, T. R. **Grail para controle supervísório de sistemas a eventos discretos**. UFSC, Florianópolis, 2002. Disponível em: <http://www.das.ufsc.br/~cury/cursos/grail_seds.ps>. Acesso em: 24 Janeiro 2005.

DE NEGRI, V. J. **Estruturação da modelagem de sistemas automáticos e sua aplicação a um banco de testes para sistemas hidráulicos**. Tese (Doutorado em Engenharia Mecânica) – Universidade Federal de Santa Catarina, Florianópolis, 1996. 180 p.

_____. **Introdução aos sistemas para automação e controle industrial**. Apostila de curso. PosMec – UFSC, 2004. 52 p.

DE NEGRI, V. J., VIEIRA, A. D. **Integração de tecnologias para a automação industrial com sistemas hidráulicos e pneumáticos**. V Seminário nacional de hidráulica e pneumáti-

co: automação e controle industrial, Florianópolis, 1997, Anais... SENAI/CTAI, 1997. p. 81-101.

DUFOUR, C. A. - **Estudo do Processo e das Ferramentas de Reprojeto de Produtos Industriais, Como Vantagem Competitiva e Estratégia de Melhoria Constante**, Dissertação de Mestrado em Engenharia, UFSC, 1996.

FERREIRA, M. G. G. **Utilização de modelos para a representação de produtos no projeto conceitual**. 140 f. Dissertação (Mestrado em Engenharia Mecânica) – Universidade Federal de Santa Catarina, Florianópolis, 1997.

FONSECA, A. J. H. **Sistematização do processo de obtenção das especificações de projeto de produtos industriais**. 180 f. Tese (Doutorado em Engenharia Mecânica) - Universidade Federal de Santa Catarina, Florianópolis, 2000.

FORCELLINI, F. A. **Projeto Conceitual**. Apostila do Curso. PosMec - UFSC, 2002. 161 p.

FREY, G., LITZ, L. **Formal methods in PLC programming**. Proceedings of the INTERNATIONAL CONFERENCE ON SYSTEMS, MAN & CYBERNETICS - IEEE, Nashville, 2000. pp. 2431-2436.

FREY, G., LITZ, L. **Verification and Validation of Control Algorithms by Coupling of Interpreted Petri Net's**. Proc. of the IEEE SMC'98, San Diego, 1998. Vol. 1, pp. 7-12.

FUTAMI, A. H. e VALENTINA, L. V. O. D. e POSSAMAI, O. – Um Modelo de Gestão do Conhecimento para a Melhoria da Qualidade do Produto. **Revista Brasileira de Gestão de Desenvolvimento de Produto**, Março 2002.

FURST, F. L.; DE NEGRI, V. J. **Projeto de sistemas hidráulicos de controle de posição: Projeto PADCT/REIVAX**, Capacitação industrial para construção de sistemas hidráulicos de controle de turbinas. LASHIP – UFSC, Florianópolis, 2002, 125 p.

JURAN, J. M. **A qualidade desde o projeto: os novos passos para o planejamento da qualidade em produtos e serviços**. São Paulo: Pioneira, 1992.

HAUSER, J. R. e CLAUSING, D. **The House of Quality**, Harvard Business Review, May-Jun., 1988.

HELLGREN, A. **On the Implementation of Discrete Event Supervisory Control**: with focus on flexible manufacturing systems. Thesis for the Degree of Doctor of Philosophy. CHALMERS UNIVERSITY OF TECHNOLOGY, Sweden, 2002. 159 p.

HEUSER, C. A. **Modelagem conceitual de sistemas**. V EBAI, 1990, 150 p.

HOUAISS, A., VILLAR, M. S. e FRANCO, F. M. M. – **Dicionário Houaiss da Língua Portuguesa**, Objetiva – RJ, 2001.

HSU, P. L., LEE, J. S. **A PLC-Based Design for the Sequence Controller in Discrete Event Systems**. Proceedings of the International Conference on Control Applications - IEEE, Anchorage, Alaska, 2000. pp. 929-934.

INTERNATIONAL ELECTROTECHNICAL COMMISSION – IEC. **IEC 848**. Preparation of function charts for control systems. Switzerland, 1988. 100p.

LIU, J., DARABI, H. **Ladder Logic Implementation of Ramadge-Wonham Supervisory Controller**. Proceedings of the Sixth International Workshop on Discrete Event Systems (WODES'02) – IEEE, Zaragoza, Spain, 2002. 7 p.

MADER, A., WUPPER, H. **What is the method in applying formal methods to PLC applications?** 4th Int. Conf. Automation of Mixed Processes: Hybrid Dynamic Systems (ADPM), Shaker Verlag, Aachen, Germany, 2000, pp. 165-171

MADER, A. **A Classification of PLC Models and Applications.** 5th Int. Workshop on Discrete Event Systems (WODES) -- Discrete Event Systems, Analysis and Control, Kluwer Academic Publishers, Massachusetts, held in Ghent, Belgium, 2000, pp. 239-247

MAGALHÃES, A. J. P., ALMEIDA, F. G. **Falando de automação.** Secção de automação, instrumentação e controlo (SAIC) – Faculdade de Engenharia da Universidade do Porto (FEUP) – Porto, Portugal, 1996. Disponível em: <http://paginas.fe.up.pt/saic/Documentos/automa.zip> . Acesso em: 07/01/2005.

MIYAGI, P. E. Controle programável. Fundamentos do controle de sistemas a eventos discretos. São Paulo: Edgard Blücher, 1996. 194 p.

OGATA, K. **Engenharia de controle moderno.** Rio de Janeiro: Prentice Hall do Brasil, 1993, 781 p.

PAES, F. H. S., DE NEGRI, V. J. **Modelagem para Automação de Pequenas Centrais Hidrelétricas.** LASHIP – EMC – UFSC. Florianópolis, 2002. 30 p. Disponível em: <http://www.laship.ufsc.br/PDF/ApostilaPDF/ModAutCHE_Completo.pdf>. Acesso em: 02 de Fevereiro de 2005.

PAHL, G., BEITZ, W. **Engineering design: a systematic approach.** Tradução por Arnold Pomerans e Ken Wallace. London : Design Council, 1984. 397 p.

QUEIROZ, M. Controle supervisorio modular e multitarefa de sistemas compostos. Tese (Doutorado em engenharia elétrica), Universidade Federal de Santa Catarina – UFSC, Florianópolis, 2004. 147 p.

QUEIROZ, M. H., CURY, J. E. R. - **Controle Supervisorio Modular de Sistemas de Manufatura Discretos.** Revista Controle & Automação/ Vol. 13 N° 2, Campinas, Maio/Agosto de 2002. Disponível em: < <http://www.scielo.br/pdf/ca/v13n2/a04v13n2.pdf>>

QUEIROZ, M. H., CURY, J. E. R. - Synthesis and Implementation of Local Modular Supervisory Control for a Manufacturing Cell. Proceedings of the Sixth International Workshop on Discrete Event Systems (WODES'02) – IEEE, Zaragoza, Spain, 2002. 6 p.

QUEIROZ, M. H.; SANTOS, E. A. P.; CURY, J. E. R. Síntese modular do controle supervisorio em diagrama escada para uma célula de manufatura. 5° Simpósio Brasileiro de Automação Inteligente. **Anais do V simpósio brasileiro de automação inteligente.** Canela - RS, 2001.

RAYMOND, D., WOOD, D. **User's guide to Grail.** Version 2.5, 1996. Disponível em: < <http://www.csd.uwo.ca/research/grail/.papers/user.ps>>. Acesso em: 24 Janeiro 2005.

REISIG, W. Introductory examples and basic definitions. In: **Petri Netz: an introduction.** Berlin: Springer-Verlag, 1985. 161 p.

REISIG, W. **A Primer in Petri Net Design,** New York: Springer, 1992.

SANTOS, E. A. P. **Contribuições ao projeto conceitual de sistemas de manipulação e montagem automatizados.** 194 f.. Tese (Doutorado em Engenharia Elétrica) – Universidade Federal de Santa Catarina, Florianópolis, 2003.

SILVA NETO, F. A. C. **Projeto e Implementação de Controles de Sistema a Eventos Discreto em uma Unidade de Potência Hidráulica**. Iniciação científica, UFSC, Florianópolis, 2004. 50 f.

SILVA NETO, F. A. C., SOUTO, R. B., DE NEGRI, V. J. Comparação entre métodos de projeto de controle de sistemas a eventos discretos e sua implementação em CLP. **Anais do XI CREEM - UERJ**, Nova Friburgo, 2004.

SLACK, N., CHAMBERS, S., HARLAND, C., HARRISON, A. e JOHNSTON, R. – **Administração da Produção**, Atlas – SP, 1997.

SOUZA, A.D. C. **Projeto do sistema de controle de uma bancada didática para posicionadores Eletro-Hidráulicos Proporcionais**. Projeto de Fim de Curso (Engenharia de Controle e Automação Industrial), Florianópolis, 2003.

SOUZA, A.D. C. **Desenvolvimento de Sistema de Projeto e Controle de Posicionadores Hidráulicos**. Dissertação (Mestrado em engenharia mecânica) – Universidade Federal de Santa Catarina, Florianópolis, 2005.

SU, R., WONHAM, W.M. Supervisor Reduction for Discrete-Event Systems. **Discrete Event Dynamic Systems**. 2004. Volume 14, Issue 1, Pages 31 – 53.

THE INTERNATIONAL ORGANIZATION FOR STANDARDIZATION – ISO. **ISO 1219-1**. Fluid power systems and components – Graphic symbols. Switzerland, 1991. 40p.

THE INTERNATIONAL ORGANIZATION FOR STANDARDIZATION – ISO. **ISO 1219-2**. Fluid power systems and components – Circuit diagrams. Switzerland, 1995. 22p.

UNIVERSITY OF WESTERN ONTARIO - UWO. **The GRAIL+ project**. A symbolic computation environment for finite-state machines, regular expressions, and finite languages. Ontario, Update 2002. Disponível em: <<http://www.csd.uwo.ca/research/grail/>> e <<http://www.das.ufsc.br/~cury/cursos/grail-dist.zip>>. Acesso em: 24 Janeiro 2005.

WONHAM, W. M. **Design Software: XPTCT**: Version 86. Toronto, 2004. Disponível em: <<http://www.control.toronto.edu/people/profs/wonham/wonham.html>>. Acesso em: 24 Janeiro 2005.

VAZ, A F; WONHAM, W M. On supervisor reduction in discrete-event systems. **International Journal of Control**. 1986. Vol. 44, no. 2, pp. 475-491.

VIEIRA, A. D. **Implementação de estrutura de controle de sistemas a eventos discretos em controlador lógico programável, utilizando a teoria de controle supervisório modular local**. PUC-PR e UFSC, 2003. 30 p.

VIEIRA, A. D. **Contribuições à implementação de sistemas de controle supervisório**. Qualificação de Tese de Doutorado. UFSC, Florianópolis, 2004. 175 p.

YOSHIKAWA, H. Design philosophy: the state of the art. **Annals of the CIRP**, **38 (2)**, 1989. p. 579-586.

APÊNDICES

APÊNDICE 1

MODELAGEM E OPERAÇÕES SIMBÓLICAS DE SED, UTILIZANDO A FERRAMENTA GRAIL.

O Grail é uma ferramenta de modelagem e operação com máquinas de estados finitos (FM na sigla em inglês), autômatos de estados finitos, de maneira simbólica. Os arquivos de modelos devem ser gerados em formato ASCII (texto comum), cuja estrutura esta descrita na Figura 9.1.

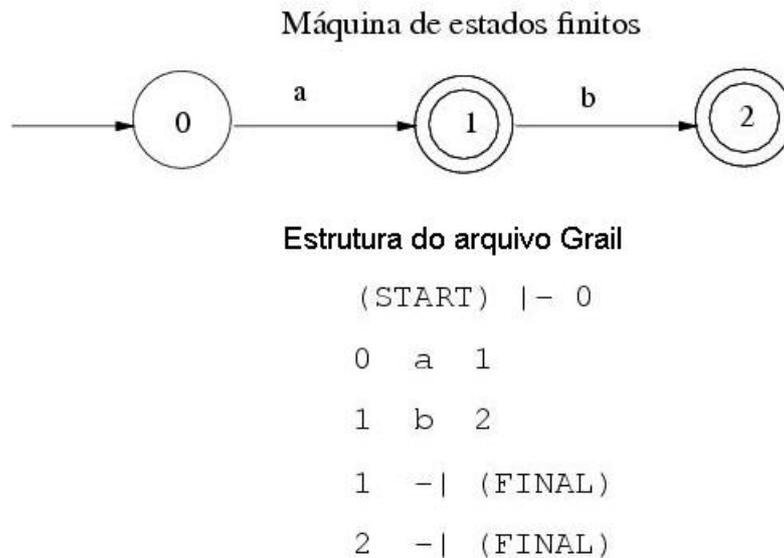


Figura 9.1 - Autômato modelado em arquivo Grail (adaptado de DA CUNHA e GARCIA, 2002).

Nota-se que a estrutura do arquivo é bastante simples sendo representadas três partes distintas: o estado inicial, as transições entre os estados do modelo e os estados marcados ou finais.

O Grail é na verdade um pacote de funções executáveis em ambiente MS-DOS, sobre os modelos em arquivos coma estrutura mostrada, cujos resultados das operações podem ser destinados a novos arquivos ou a serem exibidos na tela. Cada função tem sua própria sintaxe e argumentos de entrada para a execução, sendo que todas as operações são realizadas diretamente ia *prompt* de comando do MS-DOS. O fato de se executar funções com chamadas diretamente na linha de comando do MS-DOS permite que o projetista crie um arquivo de execução em bloco (arquivos .BAT). Este fato facilita muito o trabalho já que uma vez digitada a seqüência de comandos para o cálculo ela pode ser executada através do arquivo BAT sempre que existir a necessidade de recalculiar o controle.

Algumas funções existentes e as mais utilizadas neste trabalho estão descritas na Tabela 18.

Tabela 18 - Principais funções GRAIL utilizadas.

Função	Ação
fmrenum	Renumerar os estados de uma FM
fmstats	Obtêm informações sobre a FM
isomorph	Testa se duas FMs são isomórficas
fmalpha	Obtêm o alfabeto de uma FM
fmsupc	Encontra a máxima linguagem controlável
fmsync	Realiza o produto síncrono de duas FMs
fmtrim	Encontra a componente TRIM de uma FM

As operações utilizando o pacote Grail do projeto da unidade de potência (Capítulo 8) foram realizadas até o nível em que foram geradas as plantas locais e as especificações locais, sendo que os supervisores modulares locais e suas versões reduzidas foram obtidos utilizando o software CTCT. Um exemplo destas operações pode ser visto na Tabela 19.

Tabela 19 - Exemplos de operações com funções Grail.

Exemplo de operação	Descrição
fmsync mu1liga.txt rebu1.txt > xsp8.txt	Produto síncrono entre os modelos mu1liga e rebu1, que possuem eventos em comum, gerando o módulo 8 do sistema produto.
fmsync xsp3.txt xsp11.txt > xpl10.txt	Produto síncrono dos módulos 3 e 11 do sistema produto, por que são referenciados pela especificação edbv5, gerando a planta local 10.
fmsync xpl10.txt edbv5.txt > xel10.txt	Produto síncrono entre a planta local 10 e a especificação que a gerou, formando a especificação local 10.
fmtrim xel10.txt > xeltrim10.txt	Geração da componente TRIM da especificação local 10.

As operações mostradas na tabela acima são realizadas para todas as especificações e de posse das plantas locais e das componentes TRIM das especificações locais pode-se utilizar as funções para encontrar a máxima linguagem controlável, ou seja, os supervisores modulares locais. No caso específico deste projeto, as plantas locais e as componentes TRIM das especificações locais foram convertidos em arquivos para CTCT, que foi utilizado para calcular e reduzir os supervisores modulares.

APÊNDICE 2

EXEMPLOS DE USO DO CTCT E FERRAMENTAS DE CONVERSÃO GRAIL-CTCT

O CTCT (WONHAM, 2004) é uma ferramenta de modelagem e operação de máquinas de estados finitos, autômatos de estados finitos, de maneira numérica diferentemente do pacote Grail que realiza as operações sobre os modelos de maneira simbólica. O CTCT utiliza uma interface de operação em ambiente MS-DOS, em que as operações podem ser escolhidas através de um menu numérico como visto na Figura 9.2.

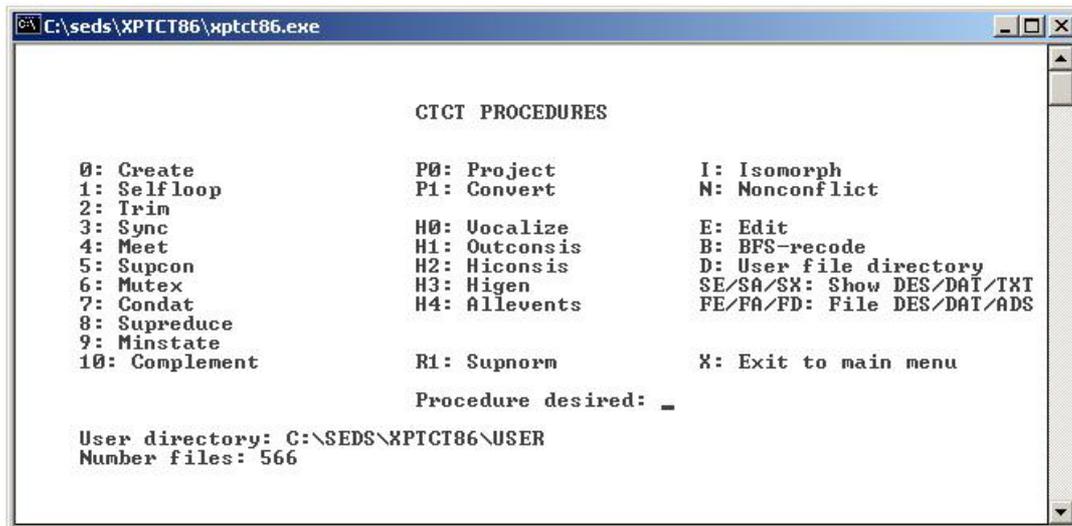


Figura 9.2 - Tela de operação do CTCT.

Os arquivos padrão de modelos do CTCT não são no formato ASCII, embora exista um formato alternativo que permita entradas de texto mas que não são muito amigáveis para a modelagem. As operações realizadas neste software são bastante rápidas mas devem ser realizadas uma a uma todas as vezes que se necessita recalculer todo o sistema, o que não facilita o trabalho.

No caso do projeto deste trabalho, os arquivos de plantas locais e especificações locais aparadas (TRIM) foram convertidos do formato Grail para o formato padrão do CTCT, de modo a permitir o término do cálculo dos supervisores modulares. Para fazer esta conversão, foi utilizado um pacote de conversão com funções executáveis em MS-DOS. Utilizou-se a função `tct2any` que converte arquivos com formato do CTCT para vários formatos inclusive o Grail e vice-versa.

Para diferenciar que tipo de operação a função vai realizar os argumentos do comando de execução, que são os arquivos a serem convertidos, devem estar com a extensão relativa ao seu formato. Por exemplo, para converter um arquivo do formato Grail para o formato CTCT o comando fica:

```
tct2any arquivo1.grail arquivo1.des (sendo que .DES é a extensão padrão do CTCT)
```

No formato do CTCT, os estados e eventos são números, e não símbolos como no Grail, e o programa diferencia os eventos controláveis dos não controláveis através da paridade, sendo que os eventos controláveis são representados por números ímpares e os não controláveis por números pares. Assim, para converter arquivos entre os formatos Grail e CTCT, é necessário um mapa de referência entre os eventos simbólicos do Grail e os eventos numéricos do CTCT e deste modo o pacote de conversão utiliza um terceiro arquivo com a extensão LABEL, com as relações entre símbolos e números dos eventos.

O arquivo LABEL deve ter o mesmo nome do arquivo convertido e caso ele não exista a execução da função de conversão vai criar um arquivo genérico. Este arquivo deve ser editado para se compatibilizar com o padrão de paridade de eventos controláveis e não controláveis, sendo que todos os arquivos que compõem o mesmo sistema devem ter as mesmas relações numéricas entre seus eventos, ou seja, se dois ou mais modelos contém eventos em comum estes eventos devem ser representados pelo mesmo número em todos os respectivos arquivos LABEL.

O arquivo LABEL serve tanto para a conversão Grail-CTCT quanto para CTCT-Grail e qualquer erro nestes arquivos torna o resultado totalmente incoerente. Um exemplo do formato do arquivo LABEL pode ser visto na Figura 9.3, onde na primeira coluna estão as representações numéricas do CTCT e na segunda as simbólicas do Grail.



Figura 9.3 - Formato do arquivo LABEL.

APÊNDICE 3

EXEMPLOS DE UTILIZAÇÃO DO CGLI

O programa CGLI converte arquivos de supervisores modulares e sistema produto em lista de instruções para CLP, levando em consideração a estrutura de implementação mostrada neste trabalho. A versão atual do CGLI permite gerar lista de instruções para dois modelos de CLP, das marcas BOSH e SIEMENS.

O programa tem interface Windows em que o projetista segue uma seqüência de operações que inicia com a escolha do modelo de CLP, Figura 9.4, e continua com a entrada dos arquivos do sistema produto e dos supervisores em formato Grail, Figura 9.5.

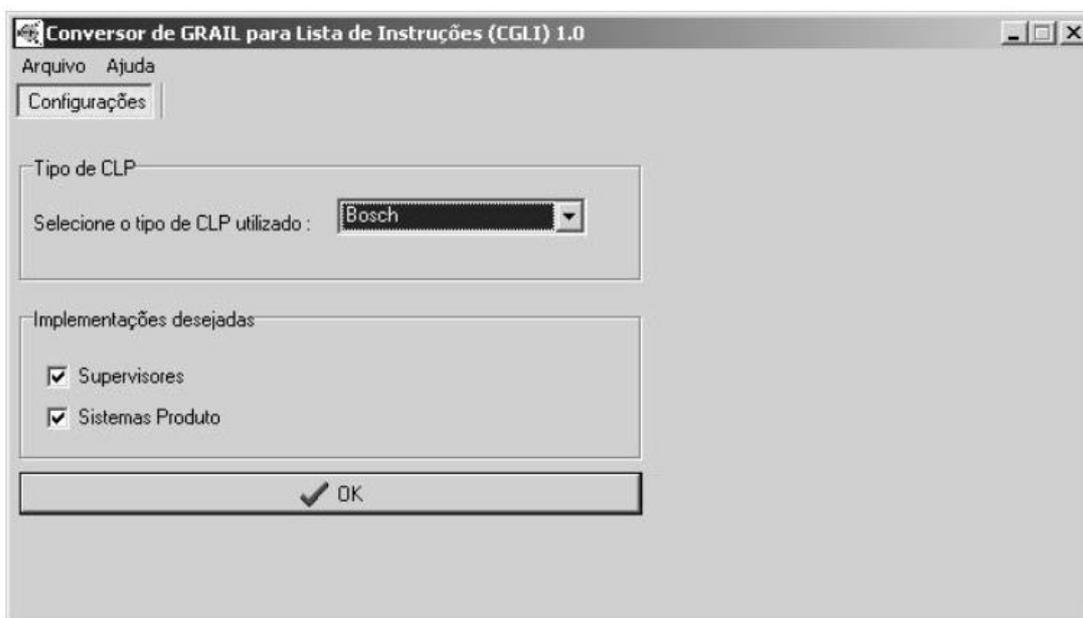


Figura 9.4 - Tela de seleção do modelo de CLP.

Na seqüência, o programa mostra a lista de símbolos relativos aos eventos do sistema, Figura 9.6, e o projetista pode configurar o nome das variáveis no CLP (que já vem pré-configurado) e a controlabilidade do evento. De maneira similar, as variáveis relativas aos estados dos supervisores e módulos do sistema produto também podem ser configurados, Figura 9.7.

Na tela final, o programa pede que o usuário informe o nome da variável de intertravamento, que impede a evolução da planta sem a atualização dos supervisores, e também a numeração de memória para a declaração das variáveis, Figura 9.8. Os arquivos são gerados no diretório onde o projeto foi iniciado sendo que todas as configurações das telas são salvas através de um arquivo de projeto com extensão GLI.

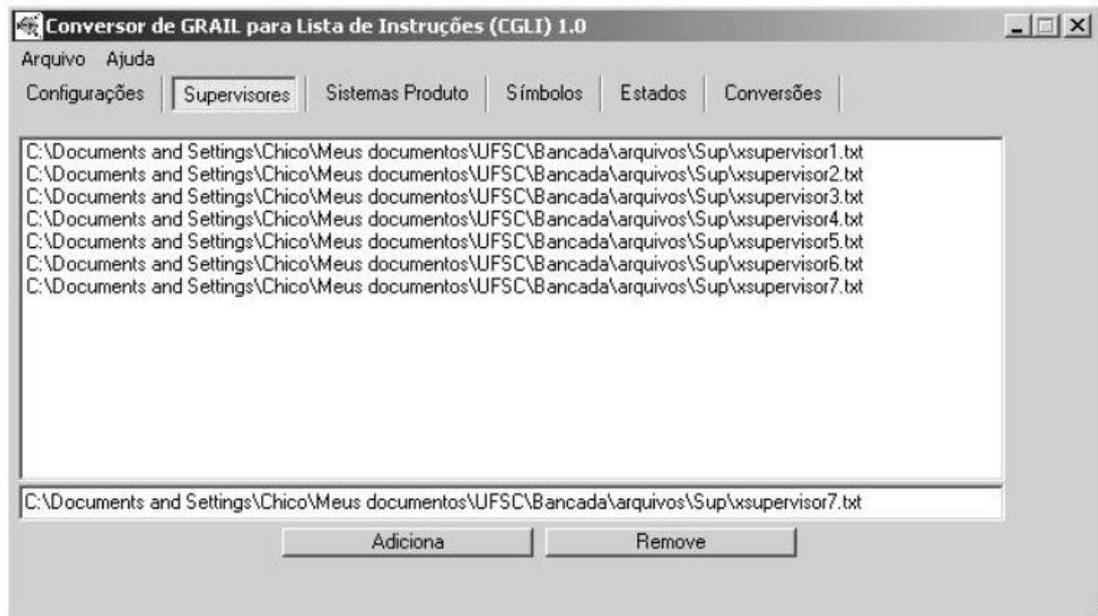


Figura 9.5 - Tela de entrada dos arquivos dos supervisores.

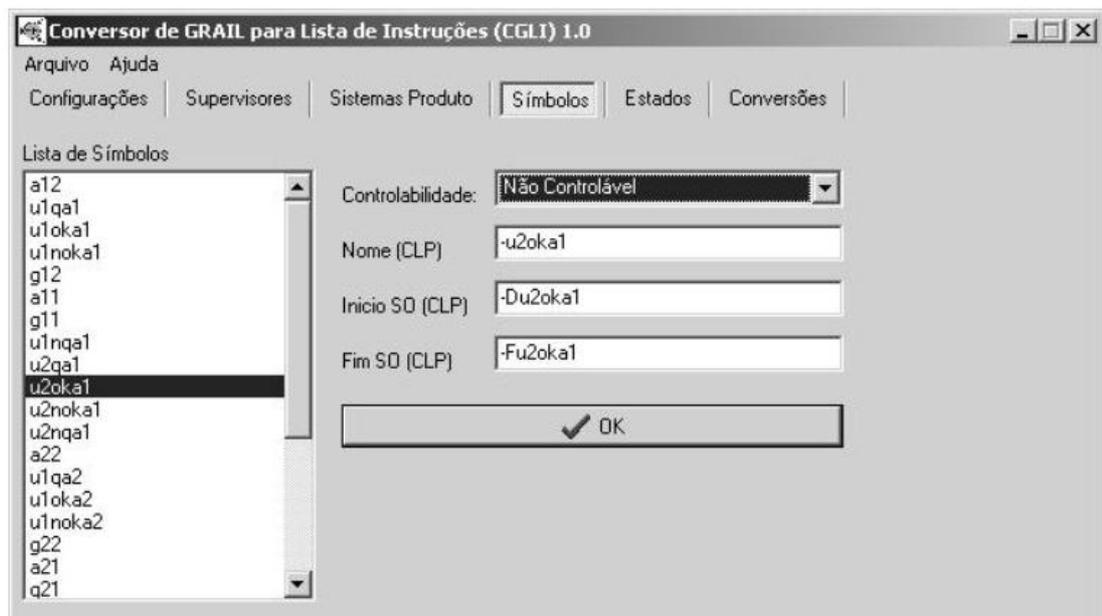


Figura 9.6 - Tela de configuração dos símbolos.

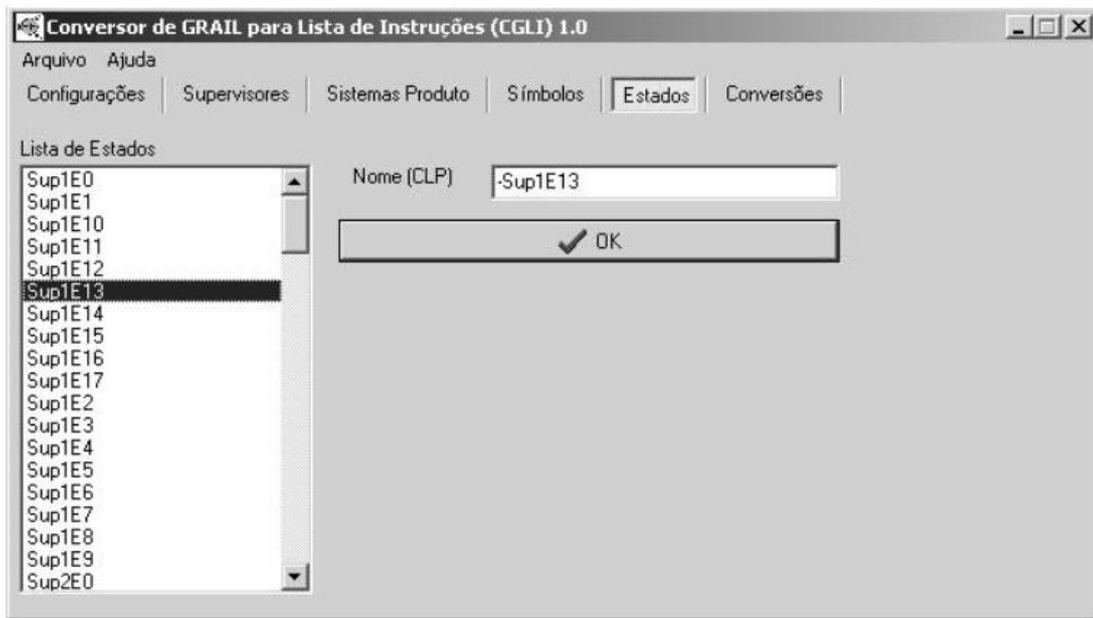


Figura 9.7 - Tela de configuração das variáveis de representação dos estados.

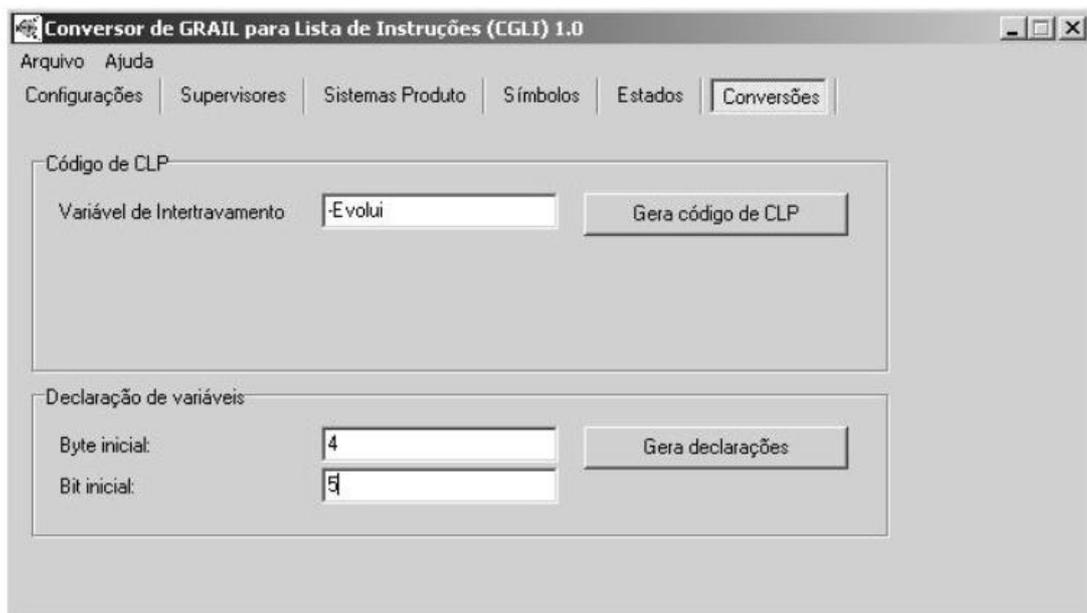


Figura 9.8 - Tela de geração do código e da lista de declaração de variáveis.

**ANEXO - CONTROLE SUPERVISÓRIO MODULAR DE SISTEMAS DE MANUFATURA
DISCRETOS.**