

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica
Departamento de Engenharia da Computação e Automação Industrial

**UM SISTEMA BASEADO EM CONHECIMENTO PARA
DIMENSIONAMENTO E CONFIGURAÇÃO DE
CENTRAIS TELEFÔNICAS CPA-T**

Autor: Laura Fazano

Orientador: Fernando A. C. Gomide †

Antonio Campos

DISSERTAÇÃO APRESENTADA À FACULDADE DE ENGENHARIA ELÉTRICA
DA UNIVERSIDADE ESTADUAL DE CAMPINAS — FEE — UNICAMP —
COMO PARTE DOS REQUISITOS EXIGIDOS PARA OBTENÇÃO DO
TÍTULO DE MESTRE EM ENGENHARIA ELÉTRICA

Agosto 1991

B.2 - 3140090

Este exemplar foi entregue à redação final da
defendida Laura Fazano
provada pela Com.
Julgadora em 22 07 91.
Fernando A. C. Gomide
Orientador

A Carmem e Thiago,
pelo tempo roubado.

“Mais do que de máquinas, precisamos de humanidade.
Mais do que de inteligência, precisamos de afeição e doçura.
Sem essas virtudes a vida será de violência e tudo será perdido.”

O Último Discurso, de Charles Chaplin.

Agradecimentos

Meus sinceros agradecimentos a todos que, de alguma forma, contribuíram para a realização deste trabalho. Em especial :

- ao Prof. Fernando A. C. Gomide, pela orientação e apoio;
- ao Aquino, meu chefe na Elebra Telecom, pelo apoio e incentivo;
- à colega Rosane, importante colaboradora no início deste trabalho;
- aos colegas Madureira e Beto, da Elebra Telecom, pela colaboração indispensável à conclusão deste trabalho;
- a todos meus colegas de CPqD, pelo incentivo e confiança;
- à querida Carmem Sílvia, pela paciência e compreensão.

Sumário

Um dos principais ramos da Inteligência Artificial é o de Sistemas Baseados em Conhecimento. Dentre esses, uma das principais áreas de atuação é a de configuração de equipamentos de grande porte, envolvendo grande quantidade de componentes e de diferentes combinações entre esses componentes. Este trabalho descreve o SSEA, Sistema de Suporte à Engenharia Aplicada, um sistema baseado em conhecimento para dimensionamento e configuração de centrais telefônicas CPA-T da família TRÓPICO RA. É descrito todo o processo de desenvolvimento do sistema, desde a Especificação de Requisitos até a implementação e testes do protótipo.

Conteúdo

Agradecimentos	iv
Sumário	v
1 Introdução	1
1.1 A Inteligência Artificial	1
1.2 Os Sistemas Especialistas	2
1.3 Objetivo do Trabalho	4
1.4 Metodologia de Desenvolvimento do Sistema	4
1.5 Organização do Trabalho	5
1.6 Notação	6
2 Sistemas Especialistas & Telecomunicações	7
2.1 Introdução	7
2.2 Interpretação	8
2.3 Diagnóstico/Reparo	8
2.4 Configuração	10
2.5 Planejamento	11
2.6 Depuração	12
2.7 Conclusão	13
3 Engenharia Aplicada & Sistema TRÓPICO RA	14
3.1 Introdução	14
3.2 O Processo de Engenharia Aplicada	14
3.3 O Sistema TRÓPICO RA	16
3.4 Conclusão	18

4	Especificação de Requisitos e Projeto Lógico	19
4.1	Introdução	19
4.2	Principais Restrições e Limitações	19
4.3	Interfaces Externas	20
4.4	Descrição da Informação	21
4.4.1	Fluxo da Informação	22
4.4.2	Conteúdo da Informação	22
4.5	Descrição Funcional	23
4.5.1	Partição Funcional	23
4.5.2	Descrição Funcional e da Informação	25
4.6	Interfaces Internas	35
4.7	Estrutura Lógica de Arquivos	37
4.8	Testes	39
4.8.1	Critérios de Validação	39
4.8.2	Projeto de Teste Funcional	41
4.9	Conclusão	42
5	Projeto Físico e Prototipagem	43
5.1	Introdução	43
5.2	Projeto Físico	43
5.2.1	Arquitetura do Software	43
5.2.2	Estrutura do Programa	45
5.2.3	Estrutura de Dados	47
5.3	Protótipo Inicial	48
5.3.1	Frames Implementados	48
5.3.2	Rotinas Implementadas em LISP	59
5.3.3	Funcionamento do Protótipo Inicial	60
5.4	Conclusão	63
6	Resultados e Conclusão	64
6.1	Resultados Obtidos	64
6.2	Conclusões	66
6.3	Trabalhos Futuros	67

A Actigramas	69
A.1 Índice	70
A.2 Actigramas	71
B Interfaces Internas	80
Bibliografía	86

Capítulo 1

Introdução

1.1 A Inteligência Artificial

Existem muitas definições sobre o que é Inteligência Artificial. Eis algumas delas:

- é o estudo das faculdades mentais através do uso de modelos computacionais [Char 86];
- é a parte da ciência da computação preocupada em projetar sistemas computacionais inteligentes, ou seja, sistemas que exibam as características que associamos com inteligência no comportamento humano – compreensão de linguagem, aprendizagem, raciocínio, resolução de problemas, etc. [Barr 81];
- é o estudo de como fazer os computadores fazerem coisas nas quais, até o momento, as pessoas são melhores [Rich 83];
- é a descoberta de técnicas que nos permitam projetar e programar máquinas que emulem e estendam nossas capacidades mentais [Jack 86].

Inteligência Artificial se divide em dois ramos principais de pesquisa: máquinas inteligentes e ciência cognitiva [Feig 85]. O primeiro ramo visa a construção de programas computacionais que desempenhem altos níveis de competência em tarefas cognitivas, enquanto o segundo se preocupa em modelar o processamento humano de informações. O ramo de máquinas inteligentes é um enfoque de engenharia e ciência da computação; não é importante se os métodos imitam a estrutura interna do comportamento humano.

O ramo de ciência cognitiva, por outro lado, utiliza o computador como uma linguagem precisa com a qual expressa hipóteses sobre processamento humano de informações, de

maneira similar ao uso da matemática pela física.

Dentro do primeiro ramo da Inteligência Artificial, existem 4 áreas principais de estudo, que são: estratégia de busca, lógica, representação do conhecimento e computação simbólica.

A computação simbólica é a área da Inteligência Artificial que tem produzido maior número de resultados práticos. Ela divide-se em sub-disciplinas relacionadas a suas aplicações:

- recuperação inteligente de informações de banco de dados;
- processamento de linguagem natural;
- aprendizado;
- visão e reconhecimento de imagens;
- robótica e sistemas de sensoriamento;
- solução de problemas combinatoriais e de escalonamento;
- prova de teoremas;
- programação automática;
- sistemas especialistas.

A maior parte das sub-disciplinas acima citadas podem ser agrupadas numa grande classe chamada Sistemas Baseados em Conhecimento, que é uma tentativa de codificar o conhecimento factual e heurístico de áreas especializadas do comportamento humano e acoplar esse conhecimento a mecanismos de inferência – métodos de resolução de problemas que podem ser usados para provar hipóteses e aplicações similares.

Dentre os Sistemas Baseados em Conhecimento, os mais conhecidos são os Sistemas Especialistas, assim chamados por usarem conhecimento especializado e procedimento de inferência na solução de problemas que normalmente só podem ser resolvidos por especialistas humanos.

1.2 Os Sistemas Especialistas

Sistemas Especialistas são normalmente classificados segundo sua área de aplicação, sendo as principais [Haye 83]:

- interpretação:** interpreta e explica uma situação a partir das informações que recebe;
- previsão:** prevê consequências prováveis a partir de uma dada situação;
- diagnóstico:** analisa condições de mal-funcionamento de um sistema, identificando possíveis causas;
- configuração:** desenvolve configurações de objetos que satisfaçam às restrições do problema;
- planejamento:** planeja ações e estratégias a serem seguidas;
- monitoração:** compara o desempenho atual do sistema com o desempenho desejado;
- depuração:** recomenda tratamento para um problema diagnosticado;
- reparo:** desenvolve e executa planos para administrar um tratamento para um problema diagnosticado;
- treinamento:** diagnostica e depura o desempenho de estudantes;
- controle:** controla o funcionamento global de um sistema.

Dentre os sistemas de configuração existentes, o mais conhecido é o XCON (inicialmente conhecido como R1), muito bem documentado [McDe 80], [McDe 81], [McDe 82], [McDe 84], [Brug 86], [Bark 89]. XCON atua na configuração dos sistemas de computadores VAX da Digital Equipment Corporation, tendo sido o primeiro sistema especialista a ser usado na produção diária de uma indústria. Sua tarefa é validar um pedido de cliente, alterando-o se necessário, e determinar a distribuição espacial dos vários componentes que formam um sistema VAX. XCON é um projeto tão bem sucedido que hoje ele é apenas um dos vários sistemas especialistas para configuração *hardware* e *software* em uso ou em desenvolvimento pela Digital. Seguiram-se ao XCON:

- XSEL:** usado interativamente para auxiliar na seleção das partes que compõem um pedido de cliente;
- XFL:** usado para diagramar um *layout* de piso da sala de computadores para a configuração especificada;
- XCLUSTER:** usado para auxiliar na configuração de *clusters*.

Estima-se que o uso desses sistemas de configuração significam para a Digital um retorno de cerca de U\$ 40 milhões/ano [Bark 89].

Existem atualmente muitos outros sistemas de configuração, em áreas tão diversas quanto arquitetura [Akin 88], [Flem 88], [Oxma 88], telecomunicações [Sche 86], circuitos integrados [DeJe 85], [Lena 82], [Vird 87], impressão de jornais [Lan 90], etc.

1.3 Objetivo do Trabalho

O objetivo deste trabalho é o de desenvolver e implementar o SSEA, Sistema de Suporte à Engenharia Aplicada, um sistema especialista para configuração do Sistema TRÓPICO RA, desenvolvido pelo CPqD TELEBRÁS. O Sistema TRÓPICO RA é uma central telefônica de grande capacidade, sendo composta por uma grande variedade de componentes, permitindo dezenas de configurações diferentes, de acordo com a aplicação a que se destina. Este trabalho é motivado por 3 fatores principais:

1. permitir que a configuração do Sistema TRÓPICO RA seja feita a um custo menor, mais rapidamente e com melhor qualidade;
2. permitir que o conhecimento necessário à tarefa de configuração seja memorizado, tendo em vista que o tempo necessário para treinamento de uma pessoa no assunto é grande, e os especialistas humanos são raros;
3. fomentar o desenvolvimento da cultura em aplicações de sistemas especialistas, tendo em vista a grande variedade de potenciais aplicações na área de telecomunicações.

1.4 Metodologia de Desenvolvimento do Sistema

Sendo o SSEA um sistema híbrido, com funções implementadas utilizando técnicas de inteligência artificial e outras utilizando técnicas tradicionais, seu desenvolvimento foi feito integrando-se essas duas técnicas, sendo a parte tradicional baseada principalmente nas técnicas descritas por [Pres 87]. Buscou-se com isso produzir um sistema bem documentado, confiável e consistente, já que seu objetivo é ser usado comercialmente pelas empresas fabricantes do Sistema TRÓPICO RA.

Assim, foram seguidas as seguintes fases para seu desenvolvimento:

1. Definição do Problema: análise e definição da tarefa de dimensionamento e configuração de centrais telefônicas TRÓPICO RA;
2. Especificação de Requisitos do Sistema: definição do domínio da informação, funções, interfaces, restrições do projeto, critérios de validação;
3. Projeto Lógico do Sistema: definição da arquitetura do sistema, determinando sua estrutura lógica, identificando a forma de implementação das funções (*software* convencional ou baseado em conhecimento) e definindo o projeto de teste do sistema;
4. Projeto Físico: definição do mapeamento funcional/físico da macro-função DIMENSIONAR, identificando a estrutura *software* (física) correspondente à estrutura lógica definida;
5. Protótipo: implementação do protótipo inicial da macro-função DIMENSIONAR, aplicando técnicas de inteligência artificial, e verificação e validação de interfaces e funções;
6. Sistema Completo: através de desenvolvimento incremental do protótipo, implementação completa da macro-função DIMENSIONAR e sua validação.

1.5 Organização do Trabalho

Esta dissertação está organizada da seguinte forma:

Capítulo 2 : discute-se alguns sistemas especialistas existentes na área de telecomunicações;

Capítulo 3 : caracteriza-se a tarefa de dimensionar e configurar o Sistema TRÓPICO RA;

Capítulo 4 : apresenta-se de forma resumida a Especificação de Requisitos e o Projeto Lógico do Sistema;

Capítulo 5 : apresenta-se o Projeto Físico da macro-função DIMENSIONAR e discute-se o processo de prototipagem dessa mesma macro-função;

Capítulo 6 : apresenta-se as conclusões obtidas durante o desenvolvimento do trabalho;

Apêndice A : traz o índice dos Actigramas e os Actigramas criados durante o desenvolvimento do sistema;

Apêndice B : traz a representação gráfica das interfaces entre macro-funções e entre funções da macro-função DIMENSIONAR.

1.6 Notação

De forma a facilitar sua leitura e compreensão, este documento utiliza uma notação específica que diferencia, através de tipos diferentes de letras, os vários elementos do sistema.

Informações são escritas em *sans serif*, como itens funcionais.

Funções são escritas em MAIÚSCULAS, como DIMENSIONAR.

Palavras em idioma estrangeiro são escritas em *itálico*, como *software*, exceto quando fizerem parte de uma informação ou função, caso em que seguirão a regra específica, como *hardware* em itens funcionais hardware.

Controles são escritos em tipo máquina, como premissas e vínculos.

Os diagramas do sistemas seguem a metodologia SADT¹, sendo chamados também atigramas. Foi incluído um símbolo especial para representação de arquivos em memória de massa e outro para representação de entidades externas, pois essa notação não existe no SADT e sua ausência prejudicaria a compreensão do sistema.

¹SADT: *Structured Analysis and Design Technique*.

Capítulo 2

Sistemas Especialistas & Telecomunicações

2.1 Introdução

A área de telecomunicações tem se desenvolvido enormemente nos últimos anos. Novas tecnologias são desenvolvidas, novos equipamentos são projetados, novas áreas de atuação são descobertas. Por outro lado, a concorrência é grande, fazendo com que as empresas produtoras de equipamentos procurem sempre produzir o equipamento com melhor desempenho, melhor qualidade e a um custo menor. Assim, muitas dessas empresas estão passando a usar sistemas especialistas de várias maneiras e em várias atividades, buscando com isso fatores que as diferenciem das demais.

Sistemas especialistas são bem aceitos nessa área devido principalmente ao fato de que a quantidade de conhecimento envolvida em cada tarefa é usualmente grande e difícil de ser tratado por *software* convencional. Por outro lado, especialistas humanos são raros e disputados a peso de ouro pelas empresas. Com o uso de sistemas especialistas, visa-se permitir que o conhecimento do especialista humano seja armazenado pela empresa, que a tarefa seja realizada de maneira consistente e eficiente, e que o conhecimento esteja disponível a um maior número de pessoas, de forma a ajudá-las na execução de suas tarefas.

A seguir, examina-se algumas das principais tarefas nas quais os sistemas especialistas já estão sendo aplicados, e algumas tarefas que no futuro provavelmente serão por eles realizadas.

2.2 Interpretação

Interpretação de sinais é o processo de utilizar os dados de um sinal para obter uma descrição em alto nível de um ambiente, incluindo os objetos presentes, suas classificações e suas localizações. Interpretação de sinais tem um papel fundamental em sistemas de supervisão cuja tarefa seja detectar, classificar e localizar elementos específicos (tais como aviões e navios) com base nos sinais que eles emitem ou refletem.

[Kuzm 88] descreve o esforço da AT&T Bell Laboratories no sentido de desenvolver novos sistemas de interpretação de sinais que satisfaçam às necessidades atuais dos sistemas de supervisão. Os sistemas em desenvolvimento integram a tecnologia de inteligência artificial com conhecimento e algoritmos desenvolvidos através de programas já existentes. São utilizadas várias formas de representação do conhecimento e vários procedimentos de utilização desse conhecimento, de modo a incorporar vários métodos de resolução de problemas apropriados aos diferentes aspectos do sistema.

Interpretação de sinais é uma tarefa complexa, que exige a integração de várias técnicas. O uso de sistemas especialistas significa um avanço real na resolução de tais problemas.

2.3 Diagnóstico/Reparo

Na produção de uma central telefônica, todos seus componentes devem ser testados. No entanto, a evolução tecnológica constante provoca uma demanda crescente pelos especialistas na tarefa, principalmente na área de diagnóstico para PBAs¹. Além do conhecimento básico de sistemas e circuitos elétricos, esta atividade requer habilidade em aplicar passos de diagnóstico baseando-se freqüentemente em dados incompletos. A demanda é alta, e para as PBAs mais complexas a falta de pessoal especializado limita a produção.

[Gunh 86] considera a tecnologia de sistemas especialistas a solução para esses problemas, pois ela possui a grande vantagem de armazenar todo o conhecimento relevante para o diagnóstico numa base de conhecimento central, permitindo que engenheiros menos experientes utilizem o conhecimento dos especialistas na tarefa. [Gunh 86] descreve ainda o processo de seleção entre um sistema especialista e um sistema convencional para a tarefa, o desenvolvimento do sistema, as dificuldades enfrentadas, principalmente com os especialistas humanos, e finalmente a aplicação bem sucedida do sistema em várias unidades da ITT Corporation na produção das centrais digitais System 12.

¹PBAs: *Printed Board Assemblies*.

Como evolução do trabalho de [Gunh 86], [Fran 89] descreve DIAESS², uma ferramenta desenvolvida pela Alcatel para geração de sistemas especialistas na área de diagnóstico de falhas em PBAs. A base de conhecimento pode ser povoada de duas formas: através de exemplos ou definindo-se explicitamente as regras. DIAESS tem sido usado para gerar sistemas especialistas para diagnóstico das PBAs da central System 12.

Um dos maiores problemas encontrados pelos operadores de centrais telefônicas grandes e complexas é assegurar a operação sem falhas. Segundo [Than 86], a central telefônica TXE4A da ITT incorpora um sistema de diagnósticos que produz relatórios de falhas e alarmes. O operador é cercado de informações para auxiliar a identificação e localização de falhas. Essa informação está disponível em muitas formas diferentes, incluindo anotações de cursos de treinamento, manuais, diagramas e listagens de falhas. Conseqüentemente, o operador deve gastar algum tempo cruzando informações e atualizando seu próprio conhecimento. Pela dificuldade em manusear essa grande quantidade de informações, ele pode não conseguir detectar a falha que originou o problema. A solução normalmente é demorada e possui prioridades conflitantes. Ao mesmo tempo, quanto maior o tempo de duração da falha menor a qualidade de serviço da central. É necessário um treinamento demorado e uma seleção cuidadosa de técnicos. Pelas características do problema e pela dificuldade na obtenção de pessoal treinado, optou-se pelo desenvolvimento de um sistema especialista que permitisse um diagnóstico *inteligente* de falhas. O sistema, conhecido como AMF³, executa num microcomputador de 16-bits acoplado à central, podendo ser usado nos modos automático e manual, e é usado com sucesso em centrais TXE4A. AMF é capaz de localizar e reparar cerca de 85% das falhas ocorridas, reduziu em 33% o tempo necessário para retirar a falha e em 75% a quantidade de documentação necessária, reduziu a quantidade de treinamento necessária para o operador, reduziu custos operacionais e aumentou o tempo entre falhas.

[Call 88] descreve TOPAS-ES⁴, um sistema especialista multi-tarefa distribuído e em tempo real para manutenção de redes de circuitos de comutação que usa o conhecimento de diagnóstico de especialistas em manutenção de rede para analisar os problemas da rede e localizar falhas. Consiste de dois sub-sistemas especialistas que trabalham juntos, cooperando

²DIAESS: *DIAGNOSIS Expert System Shell*.

³AMF: *Advanced Maintenance Facility*.

⁴TOPAS-ES: *Testing Operations Provisioning Administration System - Expert System*.

entre si. O primeiro, ESTA⁵, é responsável pela análise do problema, e o segundo, ESTS⁶, é responsável pela seccionalização do problema. Juntos formam um sistema responsável pela manutenção da rede formada por centrais 4ESS e 5ESS da AT&T, realizando sua tarefa sem auxílio de especialistas humanos.

Sistemas especialistas para diagnóstico e reparo de falhas já provaram ser eficientes e economicamente viáveis. As aplicações atuais usam abordagens nas quais o sistema especialista captura o relacionamento entre o sintoma observado e o componente em falha. O objetivo agora é obter um aumento do desempenho através do uso, pelo sistema especialista, de um modelo da arquitetura e funcionamento do sistema sob diagnose [Atwo 86].

2.4 Configuração

A configuração de uma central telefônica (também chamada Engenharia Aplicada) é o processo que converte os requisitos de uma operadora em informações detalhadas sobre a quantidade de equipamento necessária, *layout* e interconexão desse equipamento, sua conexão à rede telefônica já existente, e todas as demais informações necessárias à instalação da central. A Engenharia Aplicada é realizada por engenheiros de telecomunicações experientes que sejam especialistas na sua área, e portanto são escassos. O conhecimento usado para a tarefa normalmente está coletado em manuais, porém o engenheiro deve modificá-lo e extendê-lo para cada nova central. Esse processo de atualização traz dificuldades à medida que o ciclo de atualização dos manuais é ultrapassado pela taxa de mudanças no próprio conhecimento. Os sistemas convencionais existentes para automatizar a tarefa são geralmente difíceis de serem atualizados com a velocidade necessária para corresponder às mudanças existentes na base de conhecimento.

Visando solucionar esses problemas, [Sche 86] descreve um sistema especialista para a Engenharia Aplicada do System 12 da ITT. A opção por um sistema especialista foi feita levando-se em conta, ainda, o fato da central ser dividida em sub-sistemas, permitindo desenvolvimento incremental e prototipagem rápida do sistema. O sistema encontra-se integrado ao conjunto de ferramentas *software* do System 12, tornando o conhecimento, antes retido nas mãos de poucos, disponível em todas as unidades envolvidas na fabricação da central.

⁵ESTA: *Expert System Trouble Analyzer*.

⁶ESTS: *Expert System Trouble Sectionalizer*.

Para a área de Engenharia Aplicada, a abordagem de sistemas especialistas tem várias vantagens sobre a abordagem convencional, pois o *software* convencional exige codificação e recodificação freqüente do programa para testar e aperfeiçoar a manipulação e o uso do conhecimento. Codificar um programa desse porte é uma tarefa demorada, e mesmo pequenas revisões no conhecimento podem exigir que o programa seja em grande parte reescrito ou mesmo inteiramente descartado. A especificação do conhecimento é por demais complexa para ser corretamente apreendida por uma especificação inicial: existem muitos casos e interações, assim como alterações do próprio conhecimento. Por outro lado, a abordagem de sistemas especialistas trata a própria especificação inicial como um programa, tornando necessária pouca ou nenhuma codificação extra.

SSEA, descrito nesta dissertação, também é um sistema configurador. Neste caso, a opção por um sistema especialista levou em consideração o fato de o próprio sistema a ser configurado ainda estar em desenvolvimento. Pelo fato de o Sistema TRÓPICO RA ser altamente modular, sua versão final, prevista para entrar em operação dentro de alguns anos, conterà várias funções que não estão presentes em sua versão atual (por exemplo, sinalização por canal comum). Um sistema configurador baseado em *software* convencional sofreria grandes mudanças a cada nova função implementada na central. No caso do SSEA, apenas parte do conhecimento deverá sofrer pequenas alterações e deverá ser acrescentado o conhecimento referente à nova função implementada.

2.5 Planejamento

Planejamento de processo de componentes mecânicos visa gerar um plano detalhado de como fabricá-los, baseando-se em dados de projeto e nas tecnologias disponíveis. Inclui o planejamento de todo o processo de fabricação: seleção das operações e da seqüência na qual serão realizadas, das máquinas, aparelhos e ferramentas a serem usados, geração de dados para programação de controle numérico e cálculos de tempo de execução. [Fran 89] descreve um sistema em desenvolvimento pela Alcatel SEL, em parceria com indústrias e universidades, para planejamento de processo para fabricação de componentes. O conceito básico do sistema é ser de fácil uso e capaz de ser modificado ou ampliado para se adaptar a novas ferramentas e tecnologias. Foi implementado um protótipo inicial, em LISP, que cobre uma quantidade limitada de componentes, com resultados encorajadores.

[Fran 89] descreve ainda SONIA, um sistema para escalonamento de tarefas industriais.

Sua função principal é atribuir prioridades às atividades de acordo com critérios de escalonamento e selecionar operações de forma que seja fabricado apenas o que é realmente necessário. É responsável ainda pela alocação de recursos às tarefas selecionadas, especificando horários de início e fim de cada operação; pelo replanejamento e controle de distúrbios em tempo real, através da comparação de restrições estimadas com as restrições realmente encontradas; e pelos testes de falhas de plano, testando através de simulações as reações de um plano a situações inesperadas (tais como atrasos e quebras de máquinas).

Devido à sua complexidade, planejamento de processos ainda é um dos pontos mais fracos na automação do processo de fabricação. O uso de sistemas especialistas no planejamento de processos é um passo decisivo para a automação completa do processo de fabricação em geral, e nas empresas de telecomunicações em particular.

2.6 Depuração

Em sistemas de comutação, o *software* é usualmente grande e complexo. Além dos sistemas de centrais telefônicas, a próxima geração de sistemas de comutação possui requisitos ainda maiores que os atuais. Esses sistemas devem executar em tempo real, pois exige-se tempo de resposta imediato; devem ser multi-processo, permitindo grande concorrência para execução de várias tarefas rapidamente; devem oferecer continuidade de serviço, pois os serviços de comutação não devem ser interrompidos mesmo na presença de falhas. Esses fatores levam a uma complexidade do *software* cada vez maior, tornando a fase de testes complexa e exaustiva. Como consequência, a fase de testes normalmente tem metade do custo total do projeto e a tarefa de depuração do *software* torna-se extremamente difícil para engenheiros inexperientes.

[Akao 89] descreve DBES⁷, um sistema especialista para depuração e diagnose de *software* de sistemas de comutação desenvolvido pela NEC Corporation. DBES foi desenvolvido em conjunto com especialistas e usuários que depuram *software* de sistemas de comutação, com o objetivo de satisfazer aos requisitos de ambos e de construir um sistema que se comporte da mesma forma que os humanos, ou seja, aproximar o sistema da forma natural de pensamento humano. Um protótipo está sendo usado experimentalmente por especialistas e não especialistas. Com base nos resultados obtidos pelo protótipo, estima-se que o sistema completo será capaz de detectar entre 60% e 70% dos erros de *software*. Os erros não

⁷DBES: *DeBug Expert System*.

cobertos pelo sistema são aqueles que nunca ocorreram e aqueles que são tão complexos que mesmo os especialistas não podem prever. O tempo gasto em depuração será reduzido em 40% a 50%, e engenheiros novatos serão capazes de realizar a depuração.

A tarefa de depuração tem várias abordagens possíveis, porém a que melhor se adapta a *softwares* grandes e complexos, como é o caso de sistemas de comutação, é a abordagem de diagnóstico. Dessa forma, usa-se a heurística para permitir que engenheiros inexperientes possam depurar o *software* durante a fase de teste de integração, e não apenas na fase de teste isolado.

2.7 Conclusão

Neste capítulo analisou-se algumas das tarefas da área de telecomunicações já resolvidas ou prestes a serem resolvidas através de sistemas especialistas.

A estrutura dos sistemas especialistas é bastante diferente daquela dos sistemas convencionais. O uso de bases de conhecimento e mecanismos de inferência permite que o conhecimento normalmente embutido em qualquer *software* seja completamente separado do mecanismo de controle que determina como o programa opera. Graças a essa separação, novo conhecimento pode ser adicionado, ou o conhecimento já existente pode ser alterado, sem modificar o modo de execução do programa. Num sistema convencional, qualquer mudança normalmente requer novo teste completo do sistema para detectar possíveis efeitos colaterais. Essa característica dos sistemas baseados em conhecimento tem grande impacto sobre a produtividade e manutenibilidade do *software*. Outra característica importante é que os sistemas especialistas são capazes de justificar suas conclusões, ao contrário dos sistemas convencionais incapazes de explicá-las ao usuário.

Não há dúvida que a tecnologia de sistemas especialistas abrirá as portas da automação computacional a um grande número de aplicações que a tecnologia tradicional não viabiliza. A indústria de telecomunicações, como pudemos perceber pelos vários sistemas apresentados, está atenta a essa nova tecnologia, não apenas aplicando formas já consagradas de seu uso mas também investindo, e muito, na pesquisa de novas aplicações.

No próximo capítulo será caracterizado o processo de Engenharia Aplicada em geral e para o Sistema TRÓPICO RA em particular, procurando apresentar a tarefa a ser realizada pelo SSEA.

Capítulo 3

Engenharia Aplicada & Sistema TRÓPICO RA

3.1 Introdução

Neste capítulo, caracteriza-se o processo de Engenharia Aplicada, dividindo-o em suas atividades principais. Caracteriza-se também a Engenharia Aplicada específica do Sistema TRÓPICO RA e suas diferenças e similaridades com outros sistemas configuradores.

3.2 O Processo de Engenharia Aplicada

Engenharia Aplicada para centrais telefônicas é o processo de conversão dos requisitos de uma Companhia Operadora (ex.: TELESP) em informações detalhadas quanto a: quantidade de equipamento necessária, *layout* e interconexão desse equipamento, conexão da central à rede já existente, e todas as demais informações necessárias à instalação da central. Deve ser determinada a configuração *hardware* e *software* mais econômica possível, de forma a permitir que o fabricante vença uma concorrência, porém sem permitir que o serviço prestado pela central se deteriore devido a um sub-dimensionamento.

O processo inicia-se com o fornecimento, pela Companhia Operadora, do documento Especificação da Central, que deve conter todas as informações que permitam o correto dimensionamento de todos os componentes da central. Essas informações são, basicamente: identificação, localização e utilização do equipamento, dados de tráfego, capacidade, entroncamento, dados de encaminhamento, serviços disponíveis e evolução prevista da central.

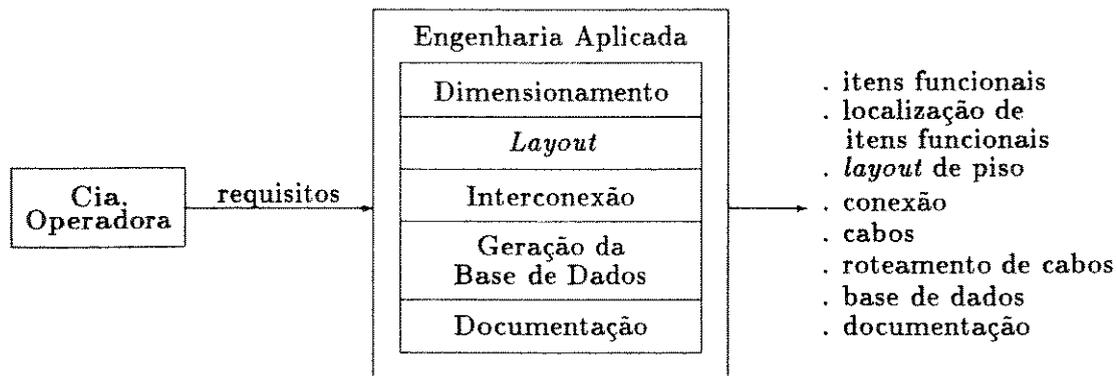


Figura 3.1: Processo de Engenharia Aplicada

A partir dessas informações, o processo é iniciado. Esse processo pode ser dividido em 5 atividades principais (fig. 3.1):

- **dimensionamento:** a partir da Especificação da Central, determina os itens funcionais, ou seja, equipamentos necessários e pacote *software* genérico da central, e o entroncamento da central à rede já existente;
- **layout:** determina a localização dos itens funcionais, ou seja, localização de cada equipamento na central e localização do *software*, e *layout* de piso da central no edifício;
- **interconexão:** determina as interconexões entre os componentes da central, identifica e rotula os cabos necessários e determina seu roteamento;
- **geração da base de dados:** realiza o povoamento das tabelas em memória de massa da central;
- **documentação:** a partir das informações produzidas pelas atividades anteriores, produz a documentação necessária para instalação, operação, manutenção, etc., da central.

A tarefa de Engenharia Aplicada é realizada por um engenheiro de telecomunicações experiente, que conheça toda a estrutura da central, seus diversos componentes, as combinações possíveis entre eles, as várias premissas e vínculos existentes para dimensionamento e configuração. Além disso, seu conhecimento deve acompanhar a evolução do sistema.

Seguindo a classificação apresentada por [Haye 83], é uma tarefa de configuração, sendo seu objetivo construir uma descrição de objetos relacionados entre si (placas, gavetas, equipamentos auxiliares) satisfazendo às premissas e vínculos específicos de cada equipamento.

3.3 O Sistema TRÓPICO RA

O Sistema TRÓPICO RA foi especificado pelo CPqD TELEBRÁS com a participação das Companhias Operadoras, visando atender às necessidades específicas do Sistema Nacional de Telecomunicações. Seu desenvolvimento conta com a participação das indústrias nacionais ELEBRA, SESA, PHT e STC, que são as fabricantes das centrais, e é progressivo, em etapas. Utiliza tecnologia digital CPA-T¹, sendo um sistema avançado tanto em termos de concepção estrutural como em componentes e materiais. Sua estrutura é altamente modular, tanto em termos de *hardware* quanto *software*, permitindo que cada central seja configurada da maneira que melhor se adequa à sua aplicação, e tendo-se em mente futuras expansões, tanto de capacidade quanto de serviços disponíveis. É uma central telefônica de grande porte, tendo capacidade para mais de 100.000 assinantes e capacidade de tráfego comutado prevista de 12.600 Erl (Erlang).

No Sistema TRÓPICO RA existem cerca de 50 PECI² diferentes; essas PECIs podem ser agrupadas de maneiras diferentes, formando sub-bastidores (gavetas); por sua vez, sub-bastidores podem ser agrupados de várias maneiras, formando bastidores (armários); finalmente, os bastidores são reunidos formando filas. Além disso, existem equipamentos auxiliares, como conversores de corrente, distribuidores de energia, dispositivos de memória de massa, painéis de alarme e periféricos (impressoras, terminais de vídeo, discos rígidos, etc.). Seu *software* é escrito quase que inteiramente em CHILL³, linguagem recomendada pelo CCITT⁴ para *software* de telecomunicações.

Devido à grande quantidade de variáveis e restrições existentes, o processo de Engenharia Aplicada do Sistema TRÓPICO RA chega a durar semanas até estar concluído, e muitas vezes não se tem certeza de que a configuração encontrada é a melhor.

SSEA é um sistema especialista cujo objetivo é automatizar o processo de Engenharia Aplicada para o Sistema TRÓPICO RA, de maneira a reduzir os custos do processo, reduzir

¹CPA-T: central com Controle por Programa Armazenado de Comutação Temporal.

²PECI: Placas Equipadas de Circuito Impresso.

³CHILL: *CCITT High Level Language*.

⁴CCITT: *Comité Consultatif International pour la Téléphonie et Télégraphie*.

o tempo de resposta entre o pedido e a produção da central, reduzir os tempos de instalação e teste e obter maior uniformidade das configurações geradas.

A tarefa de Engenharia Aplicada possui várias características que indicam para sua resolução um sistema especialista :

- o processo de Engenharia Aplicada é feito em estágios bem definidos, facilitando o desenvolvimento incremental de um sistema especialista;
- a quantidade de informação envolvida no processo é grande, dificultando o desenvolvimento de um sistema convencional, que teria que ser muito extenso e complexo para manusear tais informações;
- grande parte do conhecimento necessário à fase de dimensionamento já está compilada no Manual de Dimensionamento ([Dime90]) e no Livro Ouro do Sistema TRÓPICO RA ([Ouro 88]), bastando convertê-lo ao formato apropriado;
- um sistema especialista se adapta mais facilmente às alterações que se farão necessárias quando da introdução de novas facilidades e/ou novas tecnologias (como RDSI⁵, canal comum). Num sistema especialista essas atualizações se traduzirão em modificações ou acréscimo de novas regras, enquanto que num sistema convencional implicaria em recodificação de grande parte do código ou até mesmo em seu descarte total.

SSEA possui algumas características em comum com o sistema apresentado por [Sche 86], descrito na seção 2.4, entre elas:

- ambos têm a mesma tarefa, dimensionamento e configuração de uma central telefônica;
- ambos utilizam regras para representação do conhecimento;
- em ambos a fonte principal de conhecimento são os manuais de Engenharia Aplicada do equipamento.

Existem porém vários pontos que os diferenciam:

- o conhecimento envolvido na execução da tarefa é outro, pois enquanto o sistema descrito por [Sche 86] dimensiona a central System 12 o SSEA dimensiona a central TRÓPICO RA, de tecnologia nacional;

⁵RDSI: Rede Digital de Serviços Integrados.

- SSEA executa em micro-computador de 16-bits, enquanto que o sistema descrito por [Sche 86] executa em ambiente de grande porte; assim, a portabilidade do SSEA é maior;
- SSEA deverá, em sua versão completa, cobrir todas as fases da Engenharia Aplicada, enquanto que o sistema descrito por [Sche 86] cobre apenas a fase de dimensionamento *hardware*;
- o sistema descrito por [Sche 86] é *forward chaining* puro, fazendo com que o usuário forneça as informações conforme elas são necessárias; a desvantagem dessa abordagem é que a ordem em que as informações são solicitadas ao usuário não é necessariamente a ordem que ele, usuário, acha mais lógica, e a interação com o usuário é frequente; SSEA procura reduzir esse problema utilizando uma combinação de *forward chaining* (para instanciação de *frames*) e *backward chaining* (para busca dentro do *frame*), solicitando ao usuário grupos de informações em alguns pontos da execução, e procurando solicitar essas informações na sequência em que o usuário está habituado.

3.4 Conclusão

Neste capítulo caracterizou-se o processo de Engenharia Aplicada em geral e para o Sistema TRÓPICO RA em particular, de maneira a apresentar a tarefa realizada pelo SSEA.

Mostrou-se também os pontos de similaridade e diferenciação entre o SSEA e o sistema descrito por [Sche 86] para dimensionamento e configuração da central System 12.

No próximo capítulo serão apresentados de forma resumida a Especificação de Requisitos e o Projeto Lógico do Sistema.

Capítulo 4

Especificação de Requisitos e Projeto Lógico

4.1 Introdução

Neste capítulo apresenta-se aspectos da Especificação de Requisitos e do Projeto Lógico do Sistema. Para uma descrição completa do Sistema, o leitor pode consultar [Faza 90], capítulos 4 e 5.

Os actigramas reunidos no Apêndice A são parte importante do Projeto Lógico do SSEA, permitindo a compreensão rápida e em linhas gerais do sistema através de representação gráfica.

4.2 Principais Restrições e Limitações

As restrições principais estão contidas nas premissas e vínculos, que constituem o conhecimento especializado para desempenhar cada uma das tarefas que compõem o sistema.

SSEA divide-se em tarefas que exigem grande quantidade de conhecimento, constituindo a parte especialista propriamente dita, e em tarefas executadas através de *software* convencional, implementadas em C. Opera em microcomputador de 16 bits padrão IBM-PC, com sistema operacional DOS (versão 2.1 ou maior).

Para uma fase posterior do projeto, é possível a definição de uma interface entre o SSEA e o ADS¹, residente em sistema VAX, de forma a reduzir o número de informações a serem

¹ADS: Ambiente de Desenvolvimento de Sistemas do Sistema TRÓPICO RA.

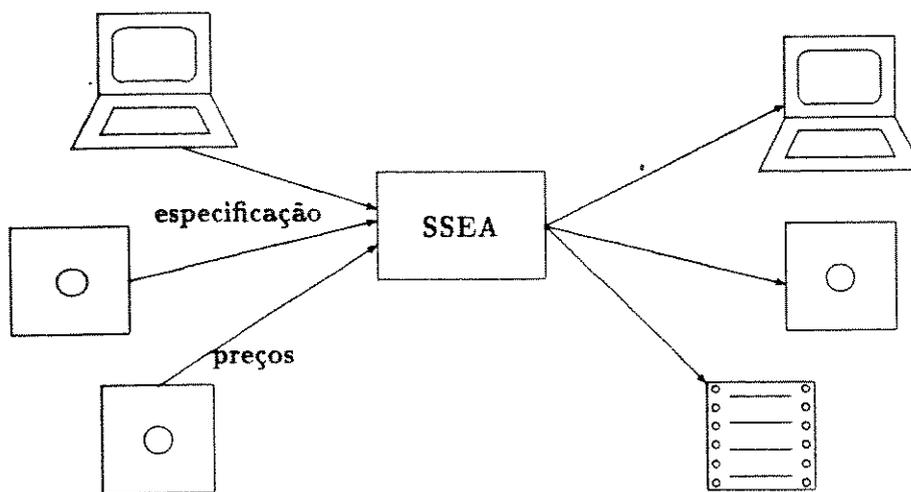


Figura 4.1: Interface do SSEA com o ambiente externo

fornecidas pelo usuário.

4.3 Interfaces Externas

A fig. 4.1 mostra como se dá a interface do sistema com o ambiente externo.

O sistema pode ser executado interativamente ou em *batch*. Em ambos os casos, busca informações sobre preços dos itens funcionais em arquivo em memória de massa.

Ao executar no modo *batch*, a interface de entrada comunica-se com unidades de memória de massa, buscando as informações necessárias em arquivos previamente gravados através de execuções anteriores do próprio sistema. Ainda no modo *batch*, a interface de saída também é feita através da unidade de memória de massa, criando arquivos texto com os resultados obtidos. Esses arquivos podem posteriormente ser listados em terminal de vídeo ou impressora.

A execução no modo interativo apresenta duas interfaces com o usuário; a primeira se faz através de menus e perguntas dirigidas, visando a obtenção da especificação da central; a segunda é uma interface gráfica através da qual o usuário fornece os diagramas representativos das plantas do edifício. Ainda no modo interativo, a interface de saída é múltipla: a interface com a memória de massa continua presente, como no modo *batch*, acompanhada

agora de outra interface com terminal de vídeo ou impressora, conforme escolha do usuário. A exceção para esse caso é com respeito às informações geradas pela macro-função GERAR BASE DE DADOS, que devido ao volume e características da informação produzida (são tabelas CHILL lidas pelo *software* da central) são apenas armazenadas em arquivos em memória de massa, e também com respeito à documentação gerada pela macro-função GERAR DOCUMENTAÇÃO, que devido ao volume são armazenadas em arquivos em memória de massa permitindo sua impressão posterior.

4.4 Descrição da Informação

O sistema, em seu nível mais abstrato (Actigrama A-0, Apêndice A), recebe como entrada a especificação da central e as plantas do edifício onde será instalada a central. Ambas as informações são fornecidas pela Companhia Operadora que deseja adquirir a central. Recebe ainda informações sobre preços dos itens funcionais hardware, residentes em arquivo em memória de massa.

A partir dessas informações, o sistema produz, como saída:

- relação de itens funcionais que constituirão a central (*hardware* e *software*), contendo preços totalizados por item e preço total da configuração fornecida;
- localização de itens funcionais selecionados, tanto internos quanto externos à central;
- layout de piso da central, ou seja, a distribuição física da central no edifício;
- conexão entre itens funcionais da central;
- relação de cabos necessários para a conexão, identificados e rotulados;
- roteamento de cabos, ou seja, seu caminho entre um item funcional *hardware* e outro;
- base de dados da central, que consiste das informações a serem utilizadas pelo *software* durante a carga e o funcionamento da central;
- documentação referente à instalação, operação, manutenção, etc., da central.

4.4.1 Fluxo da Informação

O detalhamento inicial (Actigrama A0, Apêndice A) nos mostra o sistema dividido em suas 5 macro-funções: DIMENSIONAR, ESQUEMATIZAR, INTERCONECTAR, GERAR BASE DE DADOS, GERAR DOCUMENTAÇÃO.

A especificação da central é entrada das macro-funções DIMENSIONAR, GERAR BASE DE DADOS e GERAR DOCUMENTAÇÃO.

Preços é entrada da macro-função DIMENSIONAR.

Em DIMENSIONAR, a especificação da central é transformada em itens funcionais.

Itens funcionais e plantas do edifício são transformadas, pela macro-função ESQUEMATIZAR, em esquematização, composta de layout de piso e localização de itens funcionais.

Localização de itens funcionais hardware e layout de piso são transformadas, pela macro-função INTERCONECTAR, em interconexão, composta de cabos, roteamento de cabos e conexão.

Localização de itens funcionais, itens funcionais e especificação da central se transformam, através da macro-função GERAR BASE DE DADOS, na base de dados da central.

Especificação da central, itens funcionais, esquematização, interconexão e base de dados são entradas da macro-função GERAR DOCUMENTAÇÃO, produzindo assim toda a documentação referente à central que foi configurada.

4.4.2 Conteúdo da Informação

Especificação da central é composta de: identificação/localização da central; configuração do sistema; numeração; dados de tráfego; terminais; entroncamento; serviços; encaminhamento; tarifação; sincronismo; operação, manutenção e supervisão; evolução prevista da central; e sobressalentes.

Plantas do edifício são diagramas gráficos do edifício onde será instalada a central.

Itens funcionais: relação de itens *hardware* que comporão a central, com respectivas quantidades, preços totalizados por item e preço total; e relação de itens *software* que comporão a central.

Esquematização é composto de: layout de piso (diagrama gráfico mostrando a distribuição física de bastidores e equipamentos auxiliares no edifício) e localização de itens funcionais (que se divide em localização de itens funcionais hardware e localização de itens funcionais software).

Interconexão é composto de: cabos (para cada conexão identificada, identificação do tipo de cabo que a realizará e rotulação desse cabo), conexão (para cada item funcional *hardware*, itens funcionais *hardware* a que ele se conecta) e roteamento de cabos (para cada conexão identificada, caminho pelo qual o cabo seguirá).

Base de dados é o conjunto de tabelas CHILL, definidas a nível de BSs², que contém todas as informações necessárias para a carga e o funcionamento do *software* e estarão residentes em memória de massa (disco rígido) da central.

Documentação é o conjunto de todos os documentos referentes à central a ser produzida como Manual de Operação e Manutenção, Manual de Falhas, Manual de Instalação, etc.

4.5 Descrição Funcional

4.5.1 Partição Funcional

A seguir é descrita a partição do sistema em suas macro-funções lógicas, com comentários quanto às tarefas desempenhadas, informações recebidas e fornecidas e interfaces internas e externas.

1. DIMENSIONAR

A macro-função DIMENSIONAR recebe como entrada especificação da central. A partir dessa informação, consultando o arquivo de preços e respeitando as premissas e vínculos existentes para a tarefa (que constituem o conhecimento necessário para realizar a tarefa), dimensiona um a um os módulos que compõem a central e determina seu custo. Define os itens funcionais necessários (*hardware* e *software*).

A informação produzida é tanto saída do sistema como um todo quanto entrada de outra macro-função: itens funcionais é uma das entradas de ESQUEMATIZAR, GERAR BASE DE DADOS e GERAR DOCUMENTAÇÃO.

A macro-função DIMENSIONAR possui interface, tanto de entrada quanto de saída, com o ambiente externo. Através de sua interface de entrada pode buscar as informações com um diálogo interativo com o usuário ou diretamente em arquivos em memória de massa, respeitando o modo de ativação escolhido pelo usuário. Através de sua interface de saída fornece suas respostas tanto para arquivos em memória de massa quanto para impressora ou terminal de vídeo.

²BS: Bloco de Serviço, unidade funcional do Sistema TRÓPICO RA.

2. ESQUEMATIZAR

A macro-função ESQUEMATIZAR recebe como entrada plantas do edifício e itens funcionais. A partir dessas informações, e respeitando as premissas e vínculos existentes esquematiza um a um os módulos que compõem a central. Produz a esquematização da central, ou seja, prepara um layout de piso, mostrando a localização de bastidores e equipamentos auxiliares no edifício, e define a localização de itens funcionais tanto *hardware* (placa em sub-bastidor, sub-bastidor em bastidor, etc.) quanto *software* (BS em processador).

As informações produzidas são tanto saídas do sistema como um todo quanto entradas de outras macro-funções: esquematização é uma das entradas de GERAR DOCUMENTAÇÃO, layout de piso é uma das entradas de INTERCONECTAR e localização de itens funcionais *hardware* é uma das entradas de INTERCONECTAR e GERAR BASE DE DADOS.

A macro-função ESQUEMATIZAR possui interface, tanto de entrada quanto de saída, com o ambiente externo. Através de sua interface de entrada pode buscar a informação plantas do edifício interativamente através de uma interface gráfica ou diretamente em arquivos em memória de massa, respeitando o modo de ativação escolhido pelo usuário. Através de sua interface de saída fornece suas respostas tanto para arquivos em memória de massa quanto para impressora ou terminal de vídeo.

3. INTERCONECTAR

A macro-função INTERCONECTAR recebe como entrada layout de piso e localização de itens funcionais *hardware*. A partir dessas informações, consultando o arquivo de preços e respeitando as premissas e vínculos existentes e os cabos disponíveis (que constitui a relação de cabos existentes e respectivas características, como tamanho), determina a interconexão, ou seja, a conexão física entre os itens funcionais *hardware* da central. Define a relação de cabos necessários, a conexão entre itens funcionais (*hardware*) e o roteamento de cabos.

A informação produzida é tanto saída do sistema como um todo quanto entrada de outra macro-função: interconexão é uma das entradas de GERAR DOCUMENTAÇÃO.

A macro-função INTERCONECTAR não possui interface de entrada com o ambiente externo. Através de sua interface de saída fornece suas respostas tanto para arquivos em memória de massa quanto para impressora ou terminal de vídeo.

4. GERAR BASE DE DADOS

A macro-função GERAR BASE DE DADOS recebe como entrada localização de itens funcionais, itens funcionais e especificação da central. A partir dessas informações, e respeitando a formação das tabelas (que constitui a definição CHILL das tabelas a serem geradas) gera a base de dados da central, isto é, realiza o povoamento das tabelas em memória de massa da central.

A informação produzida é tanto saída do sistema como um todo quanto entrada de outra macro-função: base de dados é uma das entradas de GERAR DOCUMENTAÇÃO.

A macro-função GERAR BASE DE DADOS possui a mesma interface de entrada com o ambiente externo que a macro-função DIMENSIONAR. Através dela pode buscar a especificação da central através de um diálogo interativo com o usuário ou diretamente em arquivos em memória de massa. Através de sua interface de saída fornece suas respostas para arquivos em memória de massa.

5. GERAR DOCUMENTAÇÃO

A macro-função GERAR DOCUMENTAÇÃO recebe como entrada todas as informações geradas pelas outras macro-funções (itens funcionais, esquematização, interconexão e base de dados) e mais especificação da central. A partir dessas informações, e respeitando os padrões de documentos (que constitui as regras de padronização dos documentos da central), gera toda a documentação referente à central.

A informação produzida é saída do sistema como um todo.

A macro-função GERAR DOCUMENTAÇÃO possui a mesma interface de entrada com o ambiente externo que a macro-função DIMENSIONAR. Através dela pode buscar a especificação da central através de um diálogo interativo com o usuário ou diretamente em arquivos em memória de massa. Através de sua interface de saída fornece suas respostas tanto para arquivos em memória de massa quanto para impressora ou terminal de vídeo.

4.5.2 Descrição Funcional e da Informação

Nesta seção faz-se uma descrição textual do funcionamento da macro-função DIMENSIONAR explicitado pelos actigramas reunidos no Apêndice A. Faz-se também uma breve descrição lógica das informações à medida em que elas aparecem nos actigramas. Para uma descrição das demais macro-funções do sistema, o leitor pode consultar [Faza 90], seção 5.3.1.

Através do Actigrama A0, Apêndice A, vê-se que DIMENSIONAR recebe como entradas a especificação da central, fornecida pelo usuário ou por arquivo em memória de massa, e os preços dos itens, fornecidos por arquivo em memória de massa, e produz como saída a relação de itens funcionais. Além disso, tem como controle o modo de ativação, ou seja, se o sistema está executando em *batch* ou interativamente, e as premissas e vínculos que representam o conhecimento usado pelo especialista no desempenho da tarefa.

Detalhando a macro-função DIMENSIONAR (Actigrama A1, Apêndice A), obtem-se sua divisão em outras 5 funções: OBTER ESPECIFICAÇÃO DA CENTRAL, REALIZAR CÁLCULOS INICIAIS, DIMENSIONAR HARDWARE, DIMENSIONAR SOFTWARE e FORNECER ITENS FUNCIONAIS.

1. OBTER ESPECIFICAÇÃO DA CENTRAL

A função OBTER ESPECIFICAÇÃO DA CENTRAL tem como controle o modo de ativação, especificando se a entrada especificação da central deve ser obtida interativamente, através de interface com o usuário, ou lida de um arquivo em memória de massa.

Após obter a especificação da central, divide-a em três grandes fluxos de informação (tráfego, informações hardware e informações software) que serão utilizados por outras funções de DIMENSIONAR.

Esta função não possui como controle premissas e vínculos, ou seja, não existe conhecimento especializado envolvido na sua realização. Assim, sem necessitar de um detalhamento maior, pode-se determinar que a função OBTER ESPECIFICAÇÃO DA CENTRAL é realizada através de *software* convencional.

Detalhando mais uma vez, obtem-se as funções em que se divide a função OBTER ESPECIFICAÇÃO DA CENTRAL (Actigrama A11, Apêndice A): DETERMINAR INTERFACE, LER ESPECIFICAÇÃO DA CENTRAL, ARMAZENAR BASE INICIAL, FATORAR ESPECIFICAÇÃO DA CENTRAL.

(a) DETERMINAR INTERFACE

A função DETERMINAR INTERFACE tem como controle o modo de ativação, determinando através dele a interface a ser utilizada pelas demais funções de OBTER ESPECIFICAÇÃO DA CENTRAL.

(b) LER ESPECIFICAÇÃO DA CENTRAL

A função LER ESPECIFICAÇÃO DA CENTRAL tem como controle a interface determinada pela função DETERMINAR INTERFACE, sabendo assim se deve obter a especificação da central interativamente, através de interface com usuário, ou do arquivo CENTRAL.DAT (onde CENTRAL é o mnemônico identificador da central que está sendo dimensionada) em memória de massa.

Após obter a especificação da central, converte-a para um formato utilizável pelas demais funções de OBTER ESPECIFICAÇÃO DA CENTRAL, a base inicial.

(c) ARMAZENAR BASE INICIAL

A função ARMAZENAR BASE INICIAL tem como controle a interface determinada pela função DETERMINAR INTERFACE. Se a interface é com usuário, armazena a base inicial fornecida pela função LER ESPECIFICAÇÃO DA CENTRAL no arquivo CENTRAL.DAT, utilizado por outras macro-funções e pela própria macro-função DIMENSIONAR quando executada em modo *batch* para a mesma central. Se a interface é com memória de massa, não precisa armazenar a base inicial, pois o arquivo já deve necessariamente existir.

(d) FATORAR ESPECIFICAÇÃO DA CENTRAL

A função FATORAR ESPECIFICAÇÃO DA CENTRAL recebe como entrada a base inicial fornecida pela função LER ESPECIFICAÇÃO DA CENTRAL e divide-a em três grandes fluxos de informação (tráfego, informações software e informações hardware) de maneira a otimizar seu uso pelas demais funções de OBTER ESPECIFICAÇÃO DA CENTRAL.

2. REALIZAR CÁLCULOS INICIAIS

A função REALIZAR CÁLCULOS INICIAIS recebe como entrada informações sobre tráfego, produzidas pela função OBTER ESPECIFICAÇÃO DA CENTRAL.

A partir dessas informações determina os resultados iniciais (composto de número de registradores, órgãos auxiliares e número de juntores por tipo de rota), utilizados por outras funções de DIMENSIONAR.

Esta função possui como controle premissas e vínculos, ou seja, envolve conhecimento especializado na sua realização. Parte da tarefa envolve cálculos específicos sobre os dados de tráfego, sendo realizada por *software* convencional, porém o controle da tarefa como um todo é feito por *software* baseado em conhecimento.

Detalhando mais uma vez, obtem-se as funções em que se divide a função REALIZAR CÁLCULOS INICIAIS (Actigrama A12, Apêndice A) : CALCULAR ÓRGÃOS AUXILIARES, CALCULAR JUNTORES, CALCULAR REGISTRADORES.

(a) CALCULAR ÓRGÃOS AUXILIARES

A função CALCULAR ÓRGÃOS AUXILIARES recebe como entrada a matriz de tráfego e o grau de serviço exigido, ambos componentes da informação tráfego fornecida pela função OBTER ESPECIFICAÇÃO DA CENTRAL. Respeitando as premissas e vínculos existentes, determina os órgãos auxiliares³ necessários.

(b) CALCULAR JUNTORES

A função CALCULAR JUNTORES recebe como entrada a matriz de tráfego, o grau de serviço exigido e o tempo de retenção, que juntos compõem a informação tráfego fornecida pela função OBTER ESPECIFICAÇÃO DA CENTRAL. Respeitando as premissas e vínculos existentes, determina o número necessário de jutores por tipo de rota⁴.

(c) CALCULAR REGISTRADORES

A função CALCULAR REGISTRADORES recebe como entradas a matriz de tráfego, o grau de serviço exigido e o tempo de retenção, que juntos compõem a informação tráfego fornecida pela função OBTER ESPECIFICAÇÃO DA CENTRAL. Respeitando as premissas e vínculos existentes, determina o número necessário de registradores. A informação registradores somada à informação jutores produzida pela função CALCULAR JUNTORES e à informação órgãos auxiliares produzida pela função CALCULAR ÓRGÃOS AUXILIARES forma a informação resultados iniciais, resultado da função REALIZAR CÁLCULOS INICIAIS.

3. DIMENSIONAR HARDWARE

A função DIMENSIONAR HARDWARE recebe como entradas informações sobre tráfego e informações hardware, produzidas pela função OBTER ESPECIFICAÇÃO DA CENTRAL, e os resultados iniciais produzidos pela função REALIZAR CÁLCULOS INICIAIS.

A partir dessas informações determina o dimensionamento hardware da central, definindo a relação de componentes hardware necessários e respectivas quantidades, utilizado por outras funções de DIMENSIONAR.

³Órgãos auxiliares: emissores MFC, receptores MFC, receptores MF.

⁴Tipo de rota: analógica ou digital.

Esta função possui como controle **premissas e vínculos**, ou seja, envolve conhecimento especializado na sua realização. Parte da tarefa envolve cálculos, principalmente sobre os dados de tráfego, sendo realizada por *software* convencional, porém o controle da tarefa e a parte que não envolve cálculos é realizada pela componente baseada em conhecimento.

Detalhando mais uma vez, obtem-se as funções em que se divide a função DIMENSIONAR HARDWARE (Actigrama A13, Apêndice A) : DIMENSIONAR MÓDULOS, DIMENSIONAR EPAL, DIMENSIONAR ITENS, DIMENSIONAR EDEG.

(a) DIMENSIONAR MÓDULOS

A função DIMENSIONAR MÓDULOS⁵ recebe como entradas as informações de tráfego e as informações hardware, produzidas pela função OBTER ESPECIFICAÇÃO DA CENTRAL, e os resultados iniciais, produzidos pela função REALIZAR CÁLCULOS INICIAIS. Respeitando as **premissas e vínculos** existentes para a tarefa, determina o dimensionamento de todos os módulos da central.

Detalhando mais uma vez, obtem-se as funções em que se divide a função DIMENSIONAR MÓDULOS (Actigrama A131, Apêndice A) : DIMENSIONAR MT E ULR, DIMENSIONAR MX, DIMENSIONAR MS, DIMENSIONAR MC, DIMENSIONAR MA, DIMENSIONAR MO.

i. DIMENSIONAR MT E ULR

A função DIMENSIONAR MT⁶ E ULR⁷ recebe como entradas a matriz de tráfego e o grau de serviço exigido, componentes da informação tráfego produzida pela função OBTER ESPECIFICAÇÃO DA CENTRAL, o número de juntores por tipo de rota, componente da informação resultados iniciais produzida pela função REALIZAR CÁLCULOS INICIAIS, e as informações MT, subconjunto de informações hardware produzidas pela função OBTER ESPECIFICAÇÃO DA CENTRAL, composta de terminais, OMS e entroncamento. Respeitando as **premissas e vínculos** existentes para a tarefa, determina o dimensionamento dos MT da central. A informação MT é composta de placas ULR (tipos e quantidades das placas presentes na ULR), placas MT

⁵Módulos de: terminais, comutação, sinalização, canal comum, auxiliar e operação, manutenção e supervisão.

⁶MT: Módulo de Terminais.

⁷ULR: Unidade de Linha Remota.

(tipos e quantidades das placas presentes no MT), tráfego MT (tráfego por MT) e n.MT (número de módulos de terminais de cada tipo⁸ necessários).

ii. DIMENSIONAR MX

A função DIMENSIONAR MX⁹ recebe como entradas n.MT e tráfego MT, ambas informações produzidas pela função DIMENSIONAR MT E ULR. Respeitando as premissas e vínculos existentes para a tarefa, determina o dimensionamento dos MX da central. A informação MX é composta de n.planos MX (número de planos de comutação), n.enlaces MX (número de enlaces intermodulares), n.MX (número de módulos de comutação) e placas MX (tipos e quantidades das placas presentes no MX).

iii. DIMENSIONAR MS

A função DIMENSIONAR MS¹⁰ recebe como entradas n.MX e n.planos MX, ambos produzidos pela função DIMENSIONAR MX, e matriz de tráfego e tempo de retenção, ambos componentes da informação tráfego produzida pela função OBTER ESPECIFICAÇÃO DA CENTRAL. Respeitando as premissas e vínculos existentes para a tarefa, determina o dimensionamento dos MS da central. A informação MS é composta de n.planos MS (número de planos de sinalização), n.enlaces MS (número de enlaces de sinalização), n.MS (número de módulos de sinalização) e placas MS (tipos e quantidades das placas presentes no MS).

iv. DIMENSIONAR MC

A função DIMENSIONAR MC¹¹ recebe como entradas n.planos MX, produzido pela função DIMENSIONAR MX, e entroncamento, componente de informações hardware produzida pela função OBTER ESPECIFICAÇÃO DA CENTRAL. Respeitando as premissas e vínculos existentes para a tarefa, determina o dimensionamento dos MC da central. A informação MC é composta de n.enlaces MC (número de enlaces de canal comum), n.MT (número de módulos de canal comum) e placas MC (tipos e quantidades das placas presentes no MC),

⁸Tipos de módulos de terminais: assinantes, troncos digitais, troncos analógicos, equipamentos auxiliares, gerais.

⁹MX: Módulo de Comutação.

¹⁰MS: Módulo de Sinalização.

¹¹MC: Módulo de Canal Comum.

v. DIMENSIONAR MA

A função DIMENSIONAR MA¹² recebe como entradas n.planos MX, produzido pela função DIMENSIONAR MX, serviços suplementares, componente de informações hardware produzida pela função OBTER ESPECIFICAÇÃO DA CENTRAL, e registradores, componente de resultados iniciais produzidos pela função REALIZAR CÁLCULOS INICIAIS. Respeitando as premissas e vínculos existentes para a tarefa, determina o dimensionamento dos MA da central. A informação MA é composta de n.MA (número de módulos auxiliares), n.duplos (número de sub-bastidores duplos necessários) e placas MA (tipos e quantidades das placas presentes no MA),

vi. DIMENSIONAR MO

A função DIMENSIONAR MO¹³ recebe como entradas n.planos MX, produzido pela função DIMENSIONAR MX, tarifação e periféricos CHM, ambos componentes de informações hardware produzidas pela função OBTER ESPECIFICAÇÃO DA CENTRAL. Respeitando as premissas e vínculos existentes para a tarefa, determina o dimensionamento dos MO da central. A informação MO é composta de n.MO (número de módulos de OMS), placas MO (tipos e quantidades das placas presentes no MO), n.MM (número de módulos de memória de massa), n.fontes MM (número de fontes alimentadoras de memória de massa), n.DR (número de unidades de disco rígido), n.DF (número de unidades de disco flexível) e n.fita (número de unidades de fita cartucho). A informação MO somada à informação MA produzida pela função DIMENSIONAR MA, à informação MC produzida pela função DIMENSIONAR MC, à informação MS produzida pela função DIMENSIONAR MS, à informação MX produzida pela função DIMENSIONAR MX e à informação MT produzida pela função DIMENSIONAR MT, forma a informação módulos, que é o resultado da função DIMENSIONAR MÓDULOS.

(b) DIMENSIONAR EPAL

A função DIMENSIONAR EPAL (Elemento Painel de Alarme) recebe como entrada painéis de alarme, componente de informações hardware produzidas pela função OBTER ESPECIFICAÇÃO DA CENTRAL. Respeitando as premissas e

¹²MA: Módulo Auxiliar.

¹³MO: Módulo de Operação, Manutenção e Supervisão.

vínculos existentes para a tarefa, determina o dimensionamento do EPAL. A informação EPAL é composta de n.EPAL (número de elementos painel de alarme), n.bastidores EPAL (número de bastidores de painel de alarme), placas EPAL (tipos e quantidades das placas presentes no EPAL).

(c) DIMENSIONAR ITENS

A função DIMENSIONAR ITENS recebe como entrada módulos produzida pela função DIMENSIONAR MÓDULOS, EPAL produzida pela função DIMENSIONAR EPAL e informações hardware produzida pela função OBTER ESPECIFICAÇÃO DA CENTRAL. Respeitando as premissas e vínculos existentes para a tarefa, determina o dimensionamento de todos os itens ainda ausentes, com exceção do EDEG (Elemento de Distribuição de Energia). A informação itens é composta de muitas informações diferentes, tais como: totalização de placas por tipo de módulo, quantidade de sub-bastidores por tipo de módulo, quantidade de bastidores.

(d) DIMENSIONAR EDEG

A função DIMENSIONAR EDEG (Elemento de Distribuição de Energia) recebe como entrada itens EDEG produzida pela função DIMENSIONAR ITENS, composto de n.MTA (número de módulos de terminais de assinantes) e n.fontes (número de fontes de alimentação da central). Respeitando as premissas e vínculos existentes para a tarefa, determina o dimensionamento do EDEG. A informação EDEG é composta de placas EDEG (tipos e quantidades das placas presentes no EDEG). A informação EDEG somada à informação itens produzida pela função DIMENSIONAR ITENS, à informação EPAL produzida pela função DIMENSIONAR EPAL e à informação módulos produzida pela função DIMENSIONAR MÓDULOS forma a informação hardware, que é o resultado da função DIMENSIONAR HARDWARE.

4. DIMENSIONAR SOFTWARE

A função DIMENSIONAR SOFTWARE recebe como entradas informações software produzida pela função OBTER ESPECIFICAÇÃO DA CENTRAL, e hardware produzida pela função DIMENSIONAR HARDWARE.

A partir dessas informações determina o dimensionamento *software* da central, defi-

nindo a configuração básica a ser usada e o pacote software (BSs, BIs¹⁴, EPISs¹⁵) a ser carregado na central.

Esta função possui como controle **premissas e vínculos**, ou seja, envolve conhecimento especializado na sua realização. Assim, sabe-se que a função DIMENSIONAR SOFTWARE é ao menos parcialmente realizada através de *software* baseado em conhecimento.

Detalhando mais uma vez, obtem-se as funções em que se divide a função DIMENSIONAR SOFTWARE (Actigrama A14, Apêndice A) : DETERMINAR CONFIGURAÇÃO BÁSICA, DETERMINAR CONFIGURAÇÃO DE BSSs, DETERMINAR BIs, DETERMINAR CONFIGURAÇÃO DE EPISS.

(a) DETERMINAR CONFIGURAÇÃO BÁSICA

A função DETERMINAR CONFIGURAÇÃO BÁSICA recebe como entradas as informações software produzidas pela função OBTER ESPECIFICAÇÃO DA CENTRAL e o hardware produzido pela função DIMENSIONAR HARDWARE. Respeitando as **premissas e vínculos** existentes para a tarefa, determina a configuração básica da central, ou seja, o segmento de mercado atingido pela central (10.000 assinantes, 100.000 assinantes, local, trânsito, etc.).

(b) DETERMINAR CONFIGURAÇÃO DE BS

A função DETERMINAR CONFIGURAÇÃO DE BS recebe como entradas as informações software produzidas pela função OBTER ESPECIFICAÇÃO DA CENTRAL, o hardware produzido pela função DIMENSIONAR HARDWARE e a configuração básica selecionada, produzida pela função DETERMINAR CONFIGURAÇÃO BÁSICA. Respeitando as **premissas e vínculos** existentes para a tarefa, determina a configuração de BSs que formarão o *software* da central.

(c) DETERMINAR BIs

A função DETERMINAR BIs recebe como entradas as informações software produzidas pela função OBTER ESPECIFICAÇÃO DA CENTRAL, o hardware produzido pela função DIMENSIONAR HARDWARE, a configuração básica selecionada produzida pela função DETERMINAR CONFIGURAÇÃO BÁSICA, e os BSs selecionados

¹⁴BI: Bloco de Implementação, unidade de implementação *software* ou *hardware* do Sistema TRÓPICO RA.

¹⁵EPIS: Elemento Padrão de Implementação *Software*, elemento *software* reutilizável do Sistema TRÓPICO RA.

produzidos pela função DETERMINAR CONFIGURAÇÃO DE BSs. Respeitando as premissas e vínculos existentes para a tarefa, determina os BIs que formarão o *software* da central.

(d) DETERMINAR CONFIGURAÇÃO DE EPISs

A função DETERMINAR CONFIGURAÇÃO DE EPISs recebe como entradas as informações *software* produzidas pela função OBTER ESPECIFICAÇÃO DA CENTRAL, o *hardware* produzido pela função DIMENSIONAR HARDWARE, a configuração básica selecionada produzida pela função DETERMINAR CONFIGURAÇÃO BÁSICA, os BSs selecionados produzidos pela função DETERMINAR CONFIGURAÇÃO DE BSs e os BIs selecionados produzidos pela função DETERMINAR BIs. Respeitando as premissas e vínculos existentes para a tarefa, determina a configuração de EPISs que formarão o *software* da central. A informação EPISs somada à informação BIs produzida pela função DETERMINAR BIs, à informação BSs produzida pela função DETERMINAR CONFIGURAÇÃO DE BSs e à informação configuração básica produzida pela função DETERMINAR CONFIGURAÇÃO BÁSICA forma a informação *software*, que é o resultado da função DIMENSIONAR SOFTWARE.

5. FORNECER ITENS FUNCIONAIS

A função FORNECER ITENS FUNCIONAIS recebe como entradas as informações sobre preços de itens através de arquivo em memória de massa, o *hardware* produzido pela função DIMENSIONAR HARDWARE e o *software* produzido pela função DIMENSIONAR SOFTWARE.

A partir dessas informações determina os itens funcionais, composto da relação de itens funcionais *software* e itens funcionais *hardware* com respectivos preços. A informação itens funcionais é o resultado da macro-função DIMENSIONAR.

Esta função tem como controle o modo de ativação, que determina se o resultado é fornecido via impressora (modo *batch*) ou via terminal de vídeo (modo interativo). Não possui como controle premissas e vínculos, ou seja, não existe conhecimento especializado envolvido na sua realização. Assim, sem necessitar de um detalhamento maior, pode-se determinar que a função FORNECER ITENS FUNCIONAIS é realizada através de *software* convencional.

Detalhando mais uma vez, obtem-se as funções em que se divide a função FORNECER ITENS FUNCIONAIS (Actigrama A15, Apêndice A) : DETERMINAR PREÇO DIMENSIONAMENTO, ARMAZENAR RESULTADO DIMENSIONAMENTO, LISTAR RESULTADO DIMENSIONAMENTO.

(a) DETERMINAR PREÇO DIMENSIONAMENTO

A função DETERMINAR PREÇO DIMENSIONAMENTO recebe como entradas o hardware produzido pela função DIMENSIONAR HARDWARE, o software produzido pela função DIMENSIONAR SOFTWARE e os preços lidos do arquivo PREÇOS.DAT. Produz o resultado do dimensionamento, composto da relação de itens funcionais hardware e itens funcionais software com respectivos preços totalizados por item.

(b) ARMAZENAR RESULTADO DIMENSIONAMENTO

A função ARMAZENAR RESULTADO DIMENSIONAMENTO recebe como entrada o resultado do dimensionamento produzido pela função DETERMINAR PREÇO DIMENSIONAMENTO e armazena-o em dois arquivos em disco: o arquivo CENTRAL.DIM recebe as informações em forma compactada, para uso posterior pelas demais macro-funções do sistema, e o arquivo CENTRAL_DIM.TXT recebe informações em forma texto, permitindo que seja posteriormente consultado ou listado pelo usuário.

(c) LISTAR RESULTADO DIMENSIONAMENTO

A função LISTAR RESULTADO DIMENSIONAMENTO recebe como entrada o resultado do dimensionamento produzido pela função DETERMINAR PREÇO DIMENSIONAMENTO. Através do controle modo de ativação determina a interface de saída usada (impressora se o modo de ativação é *batch* ou terminal de vídeo se modo de ativação é interativo) e lista a relação de itens funcionais produzida. A informação itens funcionais é o resultado produzido pela função FORNECER ITENS FUNCIONAIS e também o resultado produzido pela macro-função DIMENSIONAR.

4.6 Interfaces Internas

Nesta seção descreve-se primeiramente as interfaces entre as macro-funções do sistema, e a seguir as interfaces entre funções da macro-função DIMENSIONAR. Para uma descrição das

interfaces internas das demais macro-funções, o leitor pode consultar [Faza 90], seção 5.3.2. O Apêndice B traz a representação gráfica das interfaces descritas.

1. Interfaces Entre Macro-Funções (Actigrama A0, Apêndice A)

As macro-funções do sistema comunicam-se sempre através de arquivos em memória de massa, de maneira a permitir o desenvolvimento incremental do sistema e a execução isolada de cada macro-função, bastando para isso que os arquivos necessários tenham sido gerados anteriormente.

- (a) DIMENSIONAR comunica-se com as demais macro-funções através do arquivo CENTRAL.DIM (onde CENTRAL é o mnemônico da central que está sendo dimensionada), transmitindo a informação itens funcionais (figuras B.1, B.2, B.3, Apêndice B).
- (b) ESQUEMATIZAR comunica-se com as demais macro-funções através do arquivo CENTRAL.ESQ. Para INTERCONECTAR é transmitida a informação localização de itens funcionais hardware, e para GERAR BASE DE DADOS e GERAR DOCUMENTAÇÃO é transmitida a localização de itens funcionais. No caso de INTERCONECTAR, a comunicação é feita também através do arquivo CENTRAL.LAY, transmitindo o layout de piso (figuras B.4, B.5, B.6, Apêndice B).
- (c) INTERCONECTAR comunica-se com GERAR DOCUMENTAÇÃO através do arquivo CENTRAL.INT, transmitindo a informação interconexão (figura B.7, Apêndice B).
- (d) GERAR BASE DE DADOS comunica-se com GERAR DOCUMENTAÇÃO através do arquivo CENTRAL.TAB, transmitindo a informação base de dados (figura B.8, Apêndice B).

2. Interfaces Entre Funções Da Macro-Função Dimensionar

As funções de uma mesma macro-função comunicam-se através de variáveis globais à macro-função ou através da passagem de parâmetros na chamada da rotina. A seguir, examina-se as interfaces entre funções da macro-função DIMENSIONAR (Actigrama A1, Apêndice A).

- (a) OBTER ESPECIFICAÇÃO DA CENTRAL comunica-se com REALIZAR CÁLCULOS INICIAIS através do tráfego, com DIMENSIONAR HARDWARE através do tráfego e

das informações hardware e com DIMENSIONAR SOFTWARE através de informações software (figuras B.9, B.10, B.11, Apêndice B).

- (b) REALIZAR CÁLCULOS INICIAIS comunica-se com DIMENSIONAR HARDWARE através dos resultados iniciais (figura B.12, Apêndice B).
- (c) DIMENSIONAR HARDWARE comunica-se com DIMENSIONAR SOFTWARE e com FORNECER ITENS FUNCIONAIS através da informação hardware (figuras B.13, B.14, Apêndice B).
- (d) DIMENSIONAR SOFTWARE comunica-se com FORNECER ITENS FUNCIONAIS através da informação software (figura B.15, Apêndice B).

4.7 Estrutura Lógica de Arquivos

Descreve-se a seguir as características lógicas dos arquivos em memória de massa criados e/ou consultados pelo sistema. No caso dos arquivos CENTRAL.*, CENTRAL é o nome genérico usado para representar o mnemônico da central.

Todos os arquivos têm acesso seqüencial, com exceção de CABOS.DAT e PRECOS.DAT, de acesso indexado, e DEF_TAB.GND, com acesso definido de acordo com o tipo CHILL representado.

1. CABOS.DAT é criado externamente ao SSEA. Contém informações sobre os cabos disponíveis: identificação, características e preço do cabo. É usado para obtenção, pela macro-função INTERCONNECTAR, da relação dos cabos disponíveis e suas características.
2. CENTRAL.DAT é criado pela macro-função DIMENSIONAR. Contém a base inicial, que é a especificação da central em formato resumido, sendo o registro uma linha com 80 caracteres. Se o sistema foi ativado interativamente, serão armazenadas neste arquivo as informações fornecidas pelo usuário. Se o sistema foi ativado via *batch*, este arquivo já deve necessariamente existir pois é nele que será buscada a especificação da central. É utilizado pelas macro-funções DIMENSIONAR, GERAR BASE DE DADOS e GERAR DOCUMENTAÇÃO.
3. CENTRAL.DIM é criado pela macro-função DIMENSIONAR. Contém os itens funcionais em formato compactado, numa estrutura com as seguintes informações: tipo do

item (*hardware* ou *software*), identificação, quantidade necessária, preço unitário e preço total. É usado para obtenção, por outras macro-funções, dos itens funcionais gerados por DIMENSIONAR. É utilizado pelas macro-funções ESQUEMATIZAR, GERAR BASE DE DADOS e GERAR DOCUMENTAÇÃO.

4. CENTRAL.DIM.TXT é criado pela macro-função DIMENSIONAR. Contém os itens funcionais na forma texto, em linhas de 80 caracteres. É usado pelo usuário para impressão de relatório e/ou consulta.
5. CENTRAL.DOC é criado pela macro-função GERAR DOCUMENTAÇÃO. São na verdade vários arquivos, aqui representados pelo mesmo nome, contendo toda a documentação referente à central. É usado pelo usuário para geração em papel dos documentos referentes à central.
6. CENTRAL.ESQ é criado pela macro-função ESQUEMATIZAR. Contém a localização de itens funcionais em formato compactado, numa estrutura com a identificação, o código e a localização do item. É usado para obtenção, por outras macro-funções, da localização de itens funcionais gerada pela macro-função ESQUEMATIZAR. É utilizado pelas macro-funções INTERCONNECTAR, GERAR BASE DE DADOS e GERAR DOCUMENTAÇÃO.
7. CENTRAL_ESQ.TXT é criado pela macro-função ESQUEMATIZAR. Contém a localização de itens funcionais na forma texto. É utilizado pelo usuário para impressão de relatório e/ou consulta.
8. CENTRAL.GRF é criado pela macro-função ESQUEMATIZAR. Contém as plantas do edifício. Se o sistema foi ativado interativamente, serão armazenadas neste arquivo as informações fornecidas pelo usuário. Se o sistema foi ativado via *batch*, este arquivo já deve necessariamente existir pois é nele que serão buscadas as plantas do edifício. É utilizado pela macro-função ESQUEMATIZAR.
9. CENTRAL.INT é criado pela macro-função INTERCONNECTAR. Contém a interconexão em formato compactado, numa estrutura com a identificação e o código do cabo, o código dos itens origem e destino, a rota do cabo e seu preço unitário, e totalização por cabo. É usado para obtenção, por outras macro-funções, da interconexão gerada pela macro-função INTERCONNECTAR. É utilizado pela macro-função GERAR DOCUMENTAÇÃO.

10. **CENTRAL.INT.TXT** é criado pela macro-função **INTERCONECTAR**. Contém a interconexão na forma texto. É utilizado pelo usuário para impressão de relatório e/ou consulta.
11. **CENTRAL.LAY** é criado pela macro-função **ESQUEMATIZAR**. Contém o layout de piso em formato compactado. É usado para obtenção, por outras macro-funções, do layout de piso gerado por **ESQUEMATIZAR**. É utilizado pelas macro-funções **INTERCONECTAR** e **GERAR DOCUMENTAÇÃO**.
12. **CENTRAL.LAY.TXT** é criado pela macro-função **ESQUEMATIZAR**. Contém o layout de piso na forma expandida. É usado pelo usuário para impressão e/ou consulta ao layout de piso.
13. **CENTRAL.TAB** é criado pela macro-função **GERAR BASE DE DADOS**. Contém a base de dados em formato compactado. É usado para obtenção, por outras macro-funções, da base de dados gerada por **GERAR BASE DE DADOS**. É utilizado pela macro-função **GERAR DOCUMENTAÇÃO**.
14. **CENTRAL.TAB.TXT** é criado pela macro-função **GERAR BASE DE DADOS**, contendo a base de dados na forma texto. É utilizado pelo usuário para impressão de relatório e/ou consulta.
15. **DEF.TAB.GND** é criado externamente, contendo as definições CHILL das tabelas em memória de massa. É usado para obtenção, pela macro-função **GERAR BASE DE DADOS**, das estruturas das tabelas a serem povoadas.
16. **PREÇOS.DAT** também é criado externamente. Contém os preços dos itens funcionais disponíveis, numa estrutura com a identificação e o preço unitário de cada item. É usado para obtenção, pela macro-função **DIMENSIONAR**, dos preços de itens funcionais.

4.8 Testes

4.8.1 Critérios de Validação

SSEA é um sistema baseado em conhecimento, e como tal a forma de validação mais difundida é a execução de casos de teste com a verificação dos resultados pelos especialistas humanos ou pelas fontes de conhecimento. Essa é a forma de validação adotada para o

SSEA, porém com o cuidado de formalizar quais pontos devem ser validados no sistema completo e quais critérios adotar nessa validação.

O processo de validação final se dará quando da conclusão de cada macro-função e quando da integração entre macro-funções. Serão feitas porém validações parciais durante o desenvolvimento de cada macro-função, sendo o protótipo inicial utilizado para validação de interfaces com usuário, do método de representação do conhecimento utilizado e da correção de cada área de conhecimento implementada.

Para a validação final, serão avaliados os seguintes aspectos, sugeridos por [Haye 83]:

1. Decisões, Recomendações e Desempenho

É essencial que as decisões tomadas e recomendações fornecidas pelo sistema sejam confiáveis e pertinentes, de maneira a convencer seus usuários de sua utilidade. Deve-se também avaliar seu desempenho, se possível com o auxílio de especialistas que não tenham participado do desenvolvimento do sistema.

2. Raciocínio Correto

Além de avaliar a resposta final fornecida pelo sistema, deve-se avaliar também se o raciocínio usado para atingir a resposta está correto. Se esse raciocínio é correto ou não deve ser decidido em conjunto pelo engenheiro do conhecimento e pelo especialista humano.

3. Interface Com Usuário

A interação sistema/usuário deve ser avaliada com atenção, pois dela pode depender o sucesso do sistema. Pontos principais a serem observados:

- os termos técnicos utilizados nas perguntas e respostas geradas pelo sistema devem ser apropriados e familiares ao usuário;
- o sistema deve ser capaz de explicar o raciocínio utilizado para chegar às suas conclusões e de fazê-lo no nível de conhecimento apropriado;
- o sistema deve ser capaz de auxiliar o usuário se ele estiver confuso sobre o que deve fazer ou se precisar de auxílio por qualquer razão enquanto usar o sistema;
- as respostas do sistema devem ser fornecidas de maneira a não inibir ou constranger o usuário, visando assim minimizar a barreira muitas vezes existente quanto ao uso de computadores.

4. Eficiência

A eficiência deve ser avaliada quanto a dois aspectos principais: o sistema não deve requerer que o usuário dispenda um tempo muito longo para utilizá-lo e deve ser tecnicamente eficiente, como por exemplo quanto a uso de CPU ou acesso a disco.

Deve ser desenvolvida uma biblioteca de casos de teste, de maneira a tornar possível o teste consistente do sistema a qualquer alteração feita. Esses casos de teste devem cobrir todas as áreas de conhecimento envolvidas e devem tentar induzir o sistema a erros. Quanto maior o número de casos de teste, mais confiáveis os resultados do teste.

4.8.2 Projeto de Teste Funcional

Orientação do Teste

Após a realização do teste isolado de cada função, será realizado o Teste de Integração dessas funções.

Cada módulo já testado (função ou macro-função) será integrado aos demais. Serão testadas as interfaces com usuário, interfaces entre funções e entre macro-funções, e verificado o conteúdo dos arquivos gerados pelo sistema.

Estratégia de Integração

Cada uma das macro-funções deve ser totalmente testada antes de ser integrada às demais. A integração entre macro-funções deve ser incremental, de maneira a permitir os ajustes eventualmente necessários entre as interfaces. A ordem a ser seguida para integração de macro-funções é a ordem lógica de funcionamento do sistema: DIMENSIONAR, ESQUEMATIZAR, INTERCONECTAR, GERAR BASE DE DADOS, GERAR DOCUMENTAÇÃO. Essa ordem de integração possui dois motivos principais: o sistema é desenvolvido incrementalmente, uma macro-função de cada vez, na ordem acima descrita; e cada macro-função fornece resultados necessários ao funcionamento da próxima, também na ordem acima descrita.

Dentro de cada macro-função, a integração também deve ser incremental, neste caso *top-down*, permitindo o teste de funções inteiras a cada vez, o que facilita o teste do ponto de vista do sistema baseado em conhecimento, pois assim pode-se testar uma área do conhecimento de cada vez. São testadas interfaces com usuário, interfaces internas, interfaces entre *software* convencional e *software* baseado em conhecimento, comunicação com arqui-

vos. São também fornecidos casos de teste que exijam que a macro-função exercite cada uma das funções, e o resultado será avaliado por um especialista humano.

Considerações Especiais

As macro-funções GERAR BASE DE DADOS e GERAR DOCUMENTAÇÃO, realizadas através de *software* convencional, deverão ser testadas por métodos tradicionais de teste funcional.

As demais macro-funções, realizadas através de *software* baseado em conhecimento com algumas rotinas procedurais, devem ser testadas de acordo com o método mais utilizado para sistemas baseados em conhecimento, que é a resolução de tantos casos de teste quantos forem possíveis.

4.9 Conclusão

Neste capítulo apresentou-se, de forma resumida, a Especificação de Requisitos e o Projeto Lógico do Sistema, com ênfase à macro-função DIMENSIONAR, escolhida para prototipagem inicial.

Durante todo o processo de definição da Especificação de Requisitos e do Projeto Lógico, procurou-se usar técnicas tradicionais de Engenharia de *Software* buscando a integridade e confiabilidade do sistema. Nesse sentido a técnica utilizada (SADT) mostrou-se de grande utilidade, permitindo que vários aspectos do sistema fossem conferidos e refinados.

No próximo capítulo será apresentado o Projeto Detalhado da macro-função DIMENSIONAR e discutido o processo de prototipagem dessa mesma macro-função.

Capítulo 5

Projeto Físico e Prototipagem

5.1 Introdução

Neste capítulo, discute-se a implementação da macro-função DIMENSIONAR e seu primeiro Protótipo.

5.2 Projeto Físico

5.2.1 Arquitetura do Software

A partir da análise do tipo de conhecimento envolvido no problema e das características da ferramenta a ser usada, optou-se pelo uso de regras como modo de representação do conhecimento da macro-função DIMENSIONAR.

Com o objetivo de facilitar o desenvolvimento incremental e a manutenção do sistema, essas regras estão agrupadas de acordo com o tipo de conhecimento que representam. Esses grupos de regras surgiram através do mapeamento funcional/físico, ou seja, a partir das unidades lógicas especificadas no Projeto Funcional (cap.4) surgiram as unidades físicas representadas pelos grupos de regras. As funções implementadas em C e LISP formam bibliotecas separadas, sendo acessíveis a todos os grupos de regras.

O mapeamento funcional/físico (fig. 5.1) é quase 1 para 1, ou seja, para cada unidade lógica existe uma unidade física. Isso porque regras são uma forma mais detalhada de representação lógica, e torna-se fácil agrupá-las fisicamente seguindo-se as unidades lógicas especificadas.

Funcional	Físico
DIMENSIONAR	DIMENSIONAMENTO
OBTER ESPECIFICAÇÃO DA CENTRAL	OBTER-ESPEC-CENTRAL
DETERMINAR INTERFACE	LER-ESPEC-CENTRAL
LER ESPECIFICAÇÃO DA CENTRAL	LER-ESPEC-CENTRAL
ARMAZENAR BASE INICIAL	ARMAZENAR-BASE-INICIAL
FATORAR ESPECIFICAÇÃO DA CENTRAL	LER-ESPEC-CENTRAL
REALIZAR CÁLCULOS INICIAIS	REALIZAR-CALC-INICIAIS
CALCULAR ÓRGÃOS AUXILIARES	CALC-ORGAOS
CALCULAR JUNTORES	CALC-JUNTORES
CALCULAR REGISTRADORES	CALC-REGISTRADORES
DIMENSIONAR HARDWARE	DIMENSIONAR-HW
DIMENSIONAR MÓDULOS	DIM-MODULOS
DIMENSIONAR MT e ULR	DIM-MT-ULR
DIMENSIONAR MX	DIM-MX
DIMENSIONAR MS	DIM-MS
DIMENSIONAR MC	DIM-MC
DIMENSIONAR MA	DIM-MA
DIMENSIONAR MO	DIM-MO
DIMENSIONAR EPAL	DIM-EPAL
DIMENSIONAR ITENS	DIM-EPAL
DIMENSIONAR EDEG	DIM-EDEG
DIMENSIONAR SOFTWARE	DIMENSIONAR-SW
DETERMINAR CONFIGURAÇÃO BÁSICA	DET-CONFIG-BASICA
DETERMINAR CONFIGURAÇÃO DE BS	DET-CONFIG-BS
DETERMINAR CONFIGURAÇÃO DE EPIS	DET-CONFIG-EPIS
DETERMINAR BI's	DET-BI
FORNECER ITENS FUNCIONAIS	FORNECER-ITENS-FUNC
DETERMINAR PREÇO DIMENSIONAMENTO	DET-PRECO-DIM
ARMAZENAR RESULTADO DIMENSIONAMENTO	ARMAZENAR-DIM
LISTAR RESULTADO DIMENSIONAMENTO	LISTAR-DIM
ESQUEMATIZAR	ESQUEMATIZACAO
INTERCONECTAR	INTERCONEXAO
GERAR BASE DE DADOS	GERAR-BASE
GERAR DOCUMENTAÇÃO	GERAR-DOC

Figura 5.1: Mapeamento Funcional/Físico

A exceção a esse mapeamento 1 para 1 são as funções DETERMINAR INTERFACE, LER ESPECIFICAÇÃO DA CENTRAL e FATORAR ESPECIFICAÇÃO DA CENTRAL, todas mapeadas em um único grupo, LER-ESPEC-CENTRAL. Essas funções não possuem conhecimento especializado, e através da análise de suas tarefas optou-se por reuni-las num único grupo de regras.

5.2.2 Estrutura do Programa

A ferramenta utilizada possui um tipo de estrutura básica chamada *frame*. Tipicamente, um *frame* descreve uma classe de objetos à qual são associados vários tipos de informações, incluindo como usar o próprio *frame*. Na ferramenta, em particular, o *frame* é uma estrutura mais simples, consistindo basicamente de grupos de regras e parâmetros relacionados a sub-áreas do problema, e de algumas estruturas de controle, tais como a meta de cada *frame* e dados iniciais necessários.

Frames são utilizados para implementar fisicamente a arquitetura definida e para definir hierarquia entre os grupos de regras, conforme representado na fig. 5.2.

O *frame* DIMENSIONAMENTO agrupa todas as regras relativas à função lógica DIMENSIONAR, dividindo-se em 5 *sub-frames*: OBTER-ESP-CENTRAL, REALIZAR-CALC-INICIAIS, DIMENSIONAR-HW, DIMENSIONAR-SW e FORNECER-ITENS-FUNCIONAIS. Esses *frames* dividem-se em outros, e assim sucessivamente. Dessa forma define-se uma hierarquia entre *frames*, obedecendo aos seguintes princípios:

- um *frame* herda os parâmetros definidos em seus *frames* ancestrais, podendo modificar seu conteúdo;
- um *frame* tem acesso a regras definidas em seus *frames* descendentes;
- dois *frames* irmãos não têm acesso aos parâmetros um do outro.

Essas características permitem que sejam definidos escopos para variáveis e regras, facilitando o controle sobre o fluxo do conhecimento entre as funções, como mostrado no exemplo a seguir.

Exemplo 1: O *frame* DIM-MODULOS tem como ancestrais os *frames* DIMENSIONAR-HW, DIMENSIONAMENTO e SISTEMA, e como um de seus descendentes o *frame* DIM-MA.

Conseqüentemente, DIM-MODULOS tem acesso aos parâmetros definidos em DIMENSIONAR-HW, DIMENSIONAMENTO e SISTEMA, e às regras definidas em DIM-MA.

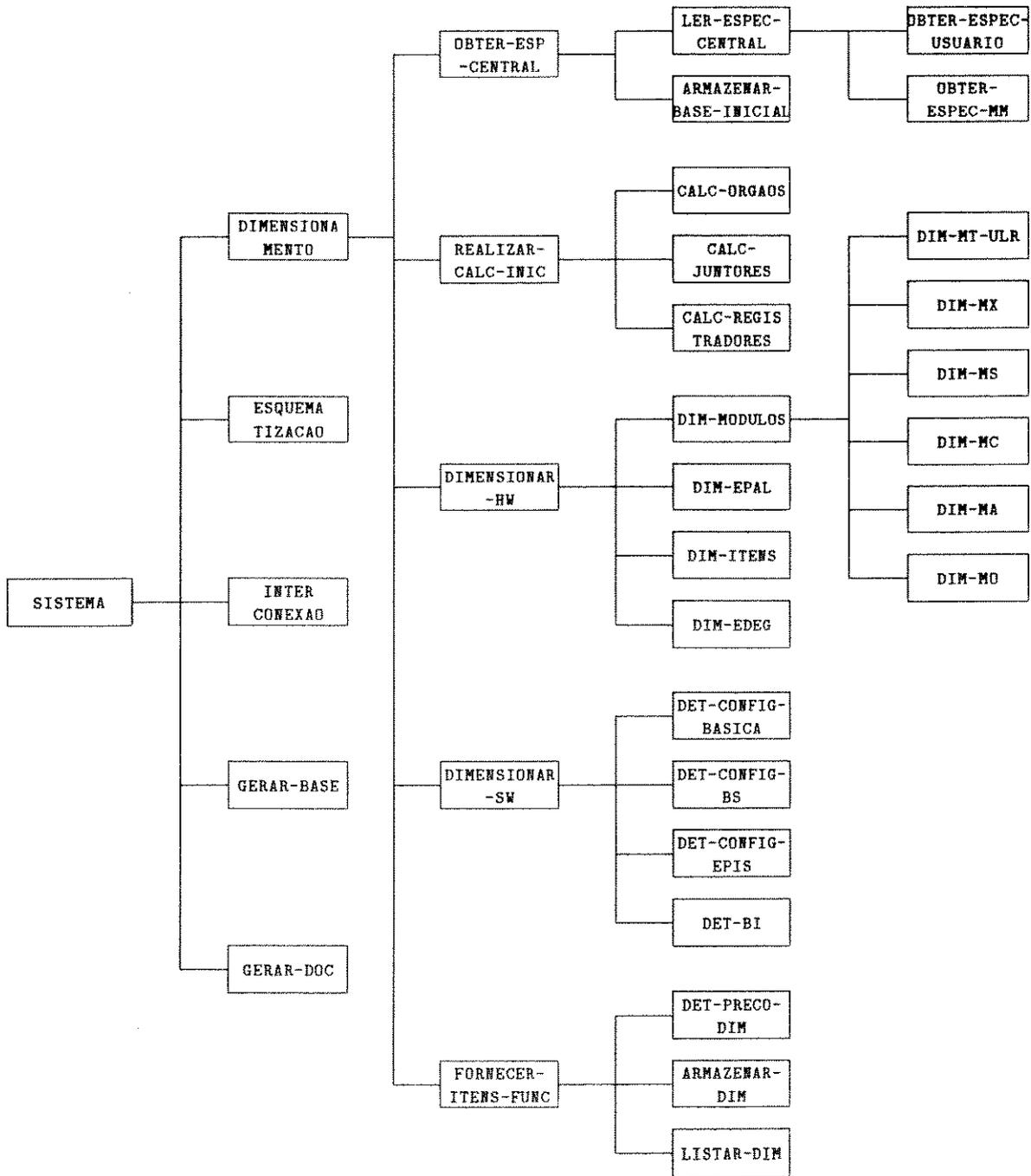


Figura 5.2: Estrutura do Programa

5.2.3 Estrutura de Dados

A estrutura básica de todos os dados da função DIMENSIONAR é a lista no padrão LISP. A ferramenta utilizada acrescenta um elemento a mais em todas as listas, que é o *fator de certeza*, porém esse elemento só é utilizado pelo SSEA para determinar se um parâmetro possui valor definido ou não. Abaixo, tem-se exemplos de parâmetros.

Exemplo 2: MO é o parâmetro que representa o Módulo de Operação, Manutenção e Supervisão, sendo uma lista cujos elementos são pares que consistem do nome da placa e da sua quantidade para aquele módulo. Um valor possível para MO é:

```
((FMC 1)(SPP 1)(UPN 1)(MEM 1)(DIS 1)(RTE 1)(SER 2)(FMG 1))
```

Exemplo 3: NOME-OP é o parâmetro que representa o nome da empresa operadora, sendo uma lista de um único elemento. Um valor possível para NOME-OP é:

```
( TELESP )
```

Com o objetivo de restringir e controlar o fluxo de informações entre *frames*, todos os dados têm seu escopo restrito ao mínimo necessário. Assim, parâmetros necessários apenas à busca de solução de um único *frame* são declarados localmente ao *frame*. Os parâmetros globais, declarados no *frame* SISTEMA, são aqueles necessários a mais de um *frame*. A seguir, tem-se exemplos de parâmetros globais e locais.

Exemplo 4: MO é um parâmetro global, definido no *frame* SISTEMA, pois é acessado por dois *frames* : DIM-MO, cuja meta é determinar seu valor, e FORNECER-ITENS-FUNC, que tem como uma meta listar seu valor.

Exemplo 5: BHCA é um parâmetro local ao *frame* DIM-MS, pois seu valor é necessário somente para o dimensionamento do Módulo de Sinalização e Sincronismo.

5.3 Protótipo Inicial

O desenvolvimento do protótipo inicial é um passo extremamente importante no processo de construção de um sistema especialista. Alguma parte do código pode ser salva para versões posteriores do sistema, porém o mais importante é testar a adequação do modo de representação do conhecimento e das ferramentas escolhidas e a viabilidade prática do projeto.

No caso do SSEA, o protótipo inicial implementa parte do *frame* DIMENSIONAMENTO, deixando de lado o *frame* DIMENSIONAR-SW, já que nem mesmo os especialistas sabem ao certo como realizar o dimensionamento do *software* do Sistema TRÓPICO RA.

Duas fontes principais de conhecimento foram utilizadas para o desenvolvimento do protótipo inicial: o Manual de Dimensionamento do Sistema TRÓPICO RA ([Dime90]) e a versão inicial de um programa de dimensionamento *hardware*, implementado por um especialista em dimensionamento da Elebra Telecon. Esse programa não abrange todo o conhecimento contido no Manual de Dimensionamento. Isso, aliado ao fato dele apresentar o conhecimento de uma forma mais lógica e estruturada que o Manual, tornou o programa a fonte ideal de conhecimento para o protótipo inicial, permitindo que o Engenheiro do Conhecimento se ambientasse mais rapidamente à área de domínio do problema.

5.3.1 Frames Implementados

A maior parte do conhecimento existente na função de Dimensionamento reside no *frame* DIM-MODULOS e seus descendentes. Os demais *frames* têm por objetivo fornecer ou receber informações de DIM-MODULOS. Tendo por objetivo a construção de um protótipo que fornecesse resultados concretos, decidiu-se iniciar a implementação por esse *frame* e implementar os demais conforme fosse necessário. Assim, o *frame* OBTER-ESP-CENTRAL foi implementado à medida que DIM-MODULOS necessitava de suas informações, o mesmo ocorrendo com FORNECER-ITENS-FUNC.

Dentre os *frames* descendentes de DIM-MODULOS, a implementação iniciou-se pelos mais simples, passando depois aos mais complexos, de maneira a permitir que o Engenheiro do Conhecimento se ambientasse gradativamente ao domínio do problema e à ferramenta utilizada. Foram implementados, pela ordem, os *frames* DIM-MO, DIM-MA, DIM-MC e DIM-MS.

Também com o objetivo de facilitar a construção do protótipo inicial, não foi implementado o modo de ativação *batch*. Todo o funcionamento é interativo, não sendo feitos acessos

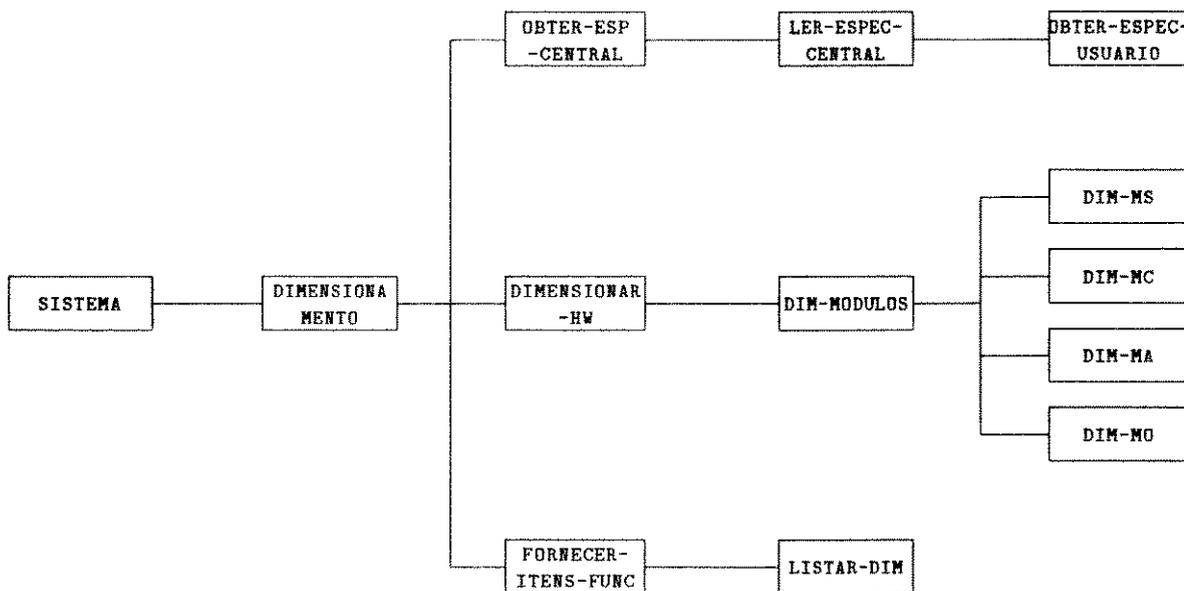


Figura 5.3: *Frames* implementados no Protótipo Inicial

a arquivos. No entanto, todos os *frames* foram implementados de maneira a facilitar futuras expansões. A estrutura de *frames* implementada no protótipo inicial está representada na fig. 5.3.

A seguir, descreve-se o conteúdo dos *frames* implementados, com alguns exemplos de regras e parâmetros.

SISTEMA

O *frame* SISTEMA controla o acionamento dos demais *frames* do sistema. Através dele, o usuário seleciona a tarefa a ser realizada e o modo de ativação do sistema. Nele estão definidos todos os parâmetros globais e alguns parâmetros de controle.

Exemplo 6: O parâmetro DIMENSIONAMENTO-OK define se a tarefa de dimensionamento foi concluída corretamente. Pode assumir os valores YES/NO, é atualizado pela regra 17 e participa da premissa da regra 16. É um parâmetro de controle.

DIMENSIONAMENTO-OK

TRANSLATION :: (o dimensionamento esta concluido)

```
TYPE :: YES/NO
UPDATED-BY :: (RULE017)
USED-BY :: (RULE016)
```

Exemplo 7: O parâmetro MA representa o Módulo Auxiliar. Pode assumir uma lista de valores, é atualizado pelas regras 58 e 59 e participa da premissa da regra 50. Seu valor deve ser determinado pelo sistema.

MA

```
TRANSLATION :: (Modulo Auxiliar)
TYPE :: SINGLEVALUED
UPDATED-BY :: (RULE058 RULE059)
USED-BY :: (RULE050)
```

Exemplo 8: O parâmetro TEMPO-CHAM representa o tempo de retenção de chamadas. Seu valor é um número positivo, é atualizado pela regra 25 e participa da conclusão da regra 80. Seu valor é fornecido pelo usuário.

TEMPO-CHAM

```
TRANSLATION :: (o tempo de retencao de chamadas)
PROMPT :: (Qual o tempo de retencao de chamadas (em segundos) ?)
TYPE :: SINGLEVALUED
EXPECT :: POSITIVE-NUMBER
UPDATED-BY :: (RULE025)
CONTAINED-IN :: (RULE080)
```

Exemplo 9: A regra 1 determina que se a tarefa selecionada é dimensionar então deve ativar o *frame* DIMENSIONAR. É uma regra antecedente, ou seja, é testada quando algum parâmetro de sua premissa recebe valor, permitindo que se implemente busca para frente (*forward chaining*).

RULE001

```
ANTECEDENT :: YES
IF :: (TAREFA = DIMENSIONAR)
THEN :: (CONSIDERFRAME DIMENSIONAMENTO)
```

DIMENSIONAMENTO

O *frame* DIMENSIONAMENTO controla a tarefa de dimensionar a central, sendo responsável pelo acionamento dos demais *frames* implementados no protótipo. Nele estão definidos parâmetros de controle.

Exemplo 10: O parâmetro DIM-HW-OK define se a tarefa de dimensionamento *hardware* foi concluída. Pode assumir os valores YES/NO, é atualizado pela regra 20 e participa da premissa da regra 17.

DIM-HW-OK

```
TRANSLATION :: (dimensionamento hardware foi concluido
                corretamente)
TYPE :: YES/NO
UPDATED-BY :: (RULE020)
USED-BY :: (RULE017)
```

Exemplo 11: O parâmetro ESP-CENTRAL-OK define se a especificação da central foi obtida. Pode receber os valores YES/NO, é atualizado pela regra 18 e participa da premissa da regra 17.

ESP-CENTRAL-OK

```
TRANSLATION :: (Especificacao da Central obtida corretamente)
TYPE :: YES/NO
UPDATED-BY :: (RULE018)
USED-BY :: (RULE017)
```

Exemplo 12: A regra 17 determina que se especificação da central foi obtida e dimensionamento *hardware* foi concluído e itens funcionais foram fornecidos então dimensionamento está concluído.

RULE017

```
IF :: (ESP-CENTRAL-OK AND DIM-HW-OK AND ITENS-FORNECIDOS-OK)
THEN :: (DIMENSIONAMENTO-OK)
```

OBTER-ESP-CENTRAL

O *frame* OBTER-ESPEC-CENTRAL controla a tarefa de obter a especificação da central, sendo responsável pelo acionamento do *frame* LER-ESPEC-CENTRAL. Nele estão definidos parâmetros de controle.

Exemplo 13: O parâmetro ESPEC-LIDA-OK define se a especificação da central foi lida. Pode receber os valores YES/NO, é atualizado pela regra 28 e participa da premissa da regra 18.

ESPEC-LIDA-OK

```
TRANSLATION :: (Especificacao da Central lida corretamente)
TYPE :: YES/NO
UPDATED-BY :: (RULE028)
USED-BY :: (RULE018)
```

Exemplo 14: A regra 18 determina que se especificação da central foi lida então especificação da central foi obtida.

RULE018

```
IF :: (ESP-LIDA-OK)
THEN :: (ESP-CENTRAL-OK)
```

LER-ESPEC-CENTRAL

O *frame* LER-ESPEC-CENTRAL controla a tarefa de ler a especificação da central, sendo responsável pelo acionamento do *frame* OBTER-ESPEC-USUARIO. Nele estão definidos parâmetros de controle.

Exemplo 15: O parâmetro ESPEC-USUARIO-OK define se a especificação da central foi fornecida pelo usuário, no caso do modo de ativação ser interativo. Pode receber os valores YES/NO, é atualizado pela regra 25 e participa da premissa da regra 28.

ESPEC-USUARIO-OK

```
TRANSLATION :: (especificacao fornecida corretamente pelo usuario)
TYPE :: YES/NO
UPDATED-BY :: (RULE025)
USED-BY :: (RULE028)
```

Exemplo 16: A regra 27 determina que se o modo de ativação é interativo então deve ativar o *frame* OBTER-ESPEC-USUARIO. É uma regra antecedente.

RULE027

```
ANTECEDENT :: YES
IF :: (MODO-ATIVACAO = INTERATIVAMENTE)
THEN :: (CONSIDERFRAME OBTER-ESPEC-USUARIO)
```

OBTER-ESPEC-USUARIO

O *frame* OBTER-ESPEC-USUARIO executa a tarefa de obter do usuário a especificação da central, sendo ativado quando o modo de ativação é interativo. Nele estão definidos parâmetros locais cujos valores devem ser fornecidos pelo usuário, e que serão copiados para os parâmetros globais.

Exemplo 17: O parâmetro L-TEMPO-CHAM representa o tempo de retenção de chamadas. Seu valor é solicitado ao usuário, que deve fornecer um número positivo. Participa da premissa e da conclusão da regra 25. É um parâmetro local ao *frame*.

L-TEMPO-CHAM

TRANSLATION :: (o tempo de retencao de chamadas)
PROMPT :: (Qual o tempo de retencao de chamadas (em segundos) ?)
TYPE :: SINGLEVALUED
EXPECT :: POSITIVE-NUMBER
USED-BY :: (RULE025)
CONTAINED-BY :: (RULE025)

Exemplo 18: A regra 25 determina que se os valores necessários foram obtidos então devem ser copiados para os parâmetros globais e a especificação foi fornecida corretamente pelo usuário.

RULE025

IF :: (L-NOME-CENTRAL IS KNOWN AND L-MNE-CENTRAL IS KNOWN AND
L-NOME-OP IS KNOWN AND L-N-REG IS KNOWN AND
L-SERV-SUPL IS KNOWN AND L-TRAF-ENT IS KNOWN AND
L-TEMPO-CHAM IS KNOWN)
THEN :: (NOME-CENTRAL = (VALUE L-NOME-CENTRAL) AND
MNE-CENTRAL = (VALUE L-MNE-CENTRAL) AND
NOME-OP = (VALUE L-NOME-OP) AND
N-REG = (VALUE L-N-REG) AND
SERV-SUPL = (VALUE L-SERV-SUPL) AND
TRAF-ENT = (VALUE L-TRAF-ENT) AND
TEMPO-CHAM = (VALUE L-TEMPO-CHAM) AND
ESPEC-USUARIO-OK)

DIMENSIONAR-HW

O *frame* DIMENSIONAR-HW controla a tarefa de dimensionamento *hardware* da central, sendo responsável pelo acionamento do *frame* DIM-MODULOS. Nele estão definidos parâmetros de controle.

Exemplo 19: O parâmetro MODULOS-DIM-OK define se os módulos foram dimensionados. Pode receber os valores YES/NO, é atualizado pela regra 32 e participa da premissa da regra 20.

MODULOS-DIM-OK

```
TRANSLATION :: (Modulos dimensionados)
TYPE :: YES/NO
UPDATED-BY :: (RULE032)
USED-BY :: (RULE020)
```

Exemplo 20: A regra 20 determina que se os módulos foram dimensionados então o dimensionamento *hardware* está concluído.

RULE020

```
IF :: (MODULOS-DIM-OK)
THEN :: (DIM-WH-OK)
```

DIM-MODULOS

O *frame* DIM-MODULOS controla o dimensionamento dos módulos da central, sendo responsável pelo acionamento dos *frames* DIM-MS, DIM-MC, DIM-MA e DIM-MD. Nele estão definidos parâmetros de controle.

Exemplo 21: O parâmetro MA-OK define se o Módulo Auxiliar foi dimensionado. Pode receber os valores YES/NO, é atualizado pela regra 50 e participa da premissa da regra 32.

MA-OK

```
TRANSLATION :: (MA dimensionado)
TYPE :: YES/NO
UPDATED-BY :: (RULE050)
USED-BY :: (RULE032)
```

Exemplo 22: A regra 20 determina que se os módulos MA, MC, MO e MS foram dimensionados então todos os módulos foram dimensionados.

RULE032

```
IF :: (MS-OK AND MC-OK AND MA-OK AND MO-OK)
THEN :: (MODULOS-DIM-OK)
```

DIM-MS

O *frame* DIM-MS executa o dimensionamento do Módulo de Sinalização da central. Determina o número de módulos necessários e a configuração de placas de cada um deles. Nele estão definidos parâmetros locais e de controle.

Exemplo 23: O parâmetro TRAF-COMUT representa o tráfego comutado. É atualizado pela regra 80 e participa da conclusão da mesma regra.

TRAF-COMUT

```
TRANSLATION :: (o trafego comutado)
TYPE :: SINGLEVALUED
UPDATED-BY :: (RULE080)
USED-BY :: (RULE080)
```

Exemplo 24: A regra 73 determina que se o número de Módulos de Sinalização é conhecido e se o Módulo de Sinalização é conhecido então o Módulo de Sinalização está dimensionado.

RULE073

```
IF :: (N-MS IS KNOWN AND MS IS KNOWN)
THEN :: (MS-OK)
```

Exemplo 25: A regra 76 determina que se o número de MX por plano é 1 então o número de placas ECV é 1.

RULE076

```
IF :: (NUM-MX-PLANO = 1)
THEN :: (ECV = 1)
```

DIM-MC

O *frame* DIM-MC executa o dimensionamento do Módulo de Canal Comum da central. Determina o número de módulos necessários e a configuração de placas de cada um deles. Nele estão definidos parâmetros locais e de controle.

Exemplo 26: A regra 64 determina que se o número de Módulos de Canal Comum é maior que 0 então MC deve receber a configuração de placas determinada. A conclusão desta regra é escrita diretamente em LISP, para permitir a montagem de uma lista de pares.

RULE064

```
IF :: (N-MC > 0)
THEN :: (MC = (LIST (CONS (QUOTE TTS) 1) (CONS (QUOTE CTE) 1)
(CONS (QUOTE FMT) 1) (CONS (QUOTE CCO) (VAL1 FRAME
CCO-AUX R)) (CONS (QUOTE DAP) (VAL1 FRAME DAP-AUX R)) ))
```

DIM-MA

O *frame* DIM-MA executa o dimensionamento do Módulo Auxiliar da central. Determina o número de módulos necessários e a configuração de placas de cada um deles. Nele estão definidos parâmetros locais e de controle.

Exemplo 27: A regra 51 determina que se existem serviços suplementares então o número de Módulos Auxiliares deve ser acrescido de 2.

RULE051

```
IF :: (SERV-SUPL)
THEN :: (N-MA = (N-MA-AUX2 + 2))
```

DIM-MO

O *frame* DIM-MO executa o dimensionamento do Módulo de Operação, Manutenção e Supervisão e do Módulo de Memória de Massa da central. Determina o número de módulos necessários e a configuração de placas de cada um deles. Nele estão definidos parâmetros de controle.

Exemplo 28: A regra 41 determina que a configuração do Módulos de Memória de Massa é fixa. Essa é a representação usada para um fato, ou uma regra cuja premissa é sempre verdadeira.

RULE041

```
IF :: (VERDADE)
THEN :: (MM = (LIST (CONS (QUOTE FMM) 2) (CONS (QUOTE EUDR) 2))
          (CONS (QUOTE EUDF) 1) (CONS (QUOTE EUMC) 1) ))
```

FORNECER-ITENS-FUNC

O *frame* FORNECER-ITENS-FUNC controla o fornecimento dos resultados obtidos, ou seja, controla a interface de saída da tarefa dimensionamento. É responsável pelo acionamento do *frame* LISTAR-DIM. Nele estão definidos parâmetros de controle.

LISTAR-DIM

O *frame* LISTAR-DIM executa a interface de saída dos resultados do dimensionamento. Fornece em vídeo a configuração de todos os módulos dimensionados. Nele estão definidos parâmetros de controle. Este *frame* faz acesso a várias funções escritas diretamente em LISP e que encontram-se numa biblioteca de rotinas.

Exemplo 28: A regra 70 determina que se o cabeçalho do Módulo de Canal Comum foi impresso então imprime os resultados obtidos e MC foi listado. Faz acesso à rotina FORNECE-PLACAS, escrita em LISP.

RULE070

```
IF :: (CABEC-MC-OK)
```

```

THEN :: (PRINT :LEFT 4 "Numero de Modulos de Canal Comum :" N-MC
        :LINE AND (E (FORNECE-PLACAS (VAL1 FRAME MC))) AND PRINT
        :HOLD/CLEAR AND DIM-MC-LIST-OK)

```

5.3.2 Rotinas Implementadas em LISP

A ferramenta utilizada fornece várias funções pré-definidas para manipulação dos parâmetros, tais como funções aritméticas e de manipulação de telas. Porém algumas rotinas necessárias tiveram que ser construídas diretamente em LISP, seja por serem procedimentos genéricos que seriam repetidos em várias regras seja por serem procedimentos recursivos cuja implementação na ferramenta seria muito trabalhosa.

No caso de procedimentos genéricos, encontram-se principalmente os procedimentos de formatação de telas para impressão de resultados. Foram construídas rotinas para impressão dos cabeçalhos, de maneira a permitir sua reutilização por vários módulos.

Exemplo 29: A rotina `cabec-dim-hw` imprime na tela o cabeçalho principal do resultado do dimensionamento. Essa rotina é utilizada sempre que se imprime uma nova tela.

```

(define cabec-dim-hw
  (lambda ()
    (mprintt :tab 3 :attr (quote (CYAN))
             "Central:" (val1 frame 'mne-central)
             :tab 40
             "Operadora:" (val1 frame 'nome-op)
             :line 1 :tab 26
             "Dimensionamento Hardware"
             :line :attr (quote (WHITE)) )))

```

No caso de procedimentos recursivos, encontram-se principalmente os procedimentos para construção ou impressão de listas, cuja manipulação de outra maneira seria muito trabalhosa.

Exemplo 30: A rotina `calc-cgs-ms` calcula o número de placas CGS para cada Módulo de Sinalização e Sincronismo. Seu resultado é uma lista onde cada elemento é o número de placas de um MS.

```
(define calc-cgs-ms
  (lambda (l-num-cgs l-n-ms)
    (if (> l-n-ms 1)
        (cons (ceiling (/ l-num-cgs l-n-ms))
              (calc-cgs-ms (- l-num-cgs (/ l-num-cgs l-n-ms))
                           (- l-n-ms 1) ))
        (cons (ceiling (/ l-num-cgs l-n-ms))
              '() ) )))
```

Exemplo 31: A rotina `fornece-placas` é responsável pela impressão de todas as duplas placa/quantidade. É utilizada sempre que se imprime o resultado do dimensionamento de um módulo.

```
(define fornece-placas
  (lambda (lista-pl)
    (if (null? lista-pl)
        '()
        (begin (mprintt :tab 4
                       "Placa: " (caar lista-pl)
                       :tab
                       "Quantidade: " (cdar lista-pl))
               (fornece-placas (cdr lista-pl) ))))
```

Todas essas rotinas encontram-se numa biblioteca separada que é carregada quando da execução do sistema.

5.3.3 Funcionamento do Protótipo Inicial

A execução do protótipo inicia-se pelo *frame* raiz, `SISTEMA`. É solicitado ao usuário que selecione uma das tarefas disponíveis e o modo de ativação desejado. Se a tarefa selecionada

é Dimensionar, DIMENSIONAMENTO é ativado através de uma regra antecedente que tem como conclusão o comando CONSIDERFRAME DIMENSIONAMENTO. Dessa maneira é implementada uma busca para frente, mais eficiente nesse caso já que se sabe com certeza que as regras necessárias encontram-se no *frame* DIMENSIONAMENTO.

O uso de *frames*, assim, passa a ter duas grandes vantagens: 1) dividir a base de conhecimento em sub-conjuntos de regras de maneira a reduzir o número de regras pesquisadas a cada busca, e 2) permitir a implementação de busca para frente nos casos em que ela é apropriada.

A partir do momento em que o *frame* DIMENSIONAMENTO é ativado, ele assume o controle da execução, que só retorna ao *frame* SISTEMA quando a meta do DIMENSIONAMENTO é atingida ou quando se esgotam as possibilidades de atingi-la. Neste caso, o dimensionamento não é concluído corretamente.

O *frame* DIMENSIONAMENTO é responsável pela ativação de outros 3 *frames*, pela ordem: OBTER-ESP-CENTRAL, DIMENSIONAR-HW e FORNECER-ITENS-FUNC. OBTER-ESP-CENTRAL é responsável, no protótipo inicial, por obter do usuário a especificação da central, interativamente. O usuário fornece a especificação através de respostas às perguntas feitas pelo sistema, e essas respostas são conferidas *on-line* para verificar se encontram-se entre os valores esperados.

Exemplo 32: O parâmetro TRAF-ENT representa o tráfego de entrada. O usuário deve fornecer seu valor, para o qual o sistema espera um número positivo. Caso o valor fornecido esteja errado, o sistema avisa ao usuário o tipo de valor esperado e solicita que ele forneça nova resposta.

Tendo obtido todos os valores necessários, a meta do *frame* OBTER-ESP-CENTRAL é atingida e o controle da execução retorna ao *frame* DIMENSIONAMENTO, que ativa a seguir o *frame* DIMENSIONAR-HW.

Em DIMENSIONAR-HW e seus descendentes encontra-se todo o conhecimento do protótipo inicial no que se refere ao domínio do problema. DIM-MODULOS aciona, pela ordem, os *frames* DIM-MS, DIM-MC, DIM-MA, DIM-MO.

Como não estão implementados no protótipo inicial os *frames* DIM-MT-ULR e DIM-MX nem o *frame* REALIZAR-CALC-INICIAIS, alguns valores necessários ao dimensionamento dos módulos e que seriam fornecidos por esses *frames* não se encontram disponíveis. O sistema solicita então que o usuário forneça esse valores, à medida que são necessários.

Exemplo 33: O parâmetro NUM-MX-PLANO representa o número de MX por plano, cujo valor é definido no dimensionamento do Módulo de Comutação. Esse dimensionamento não é realizado pelo protótipo, portanto NUM-NX-PLANO não possui valor definido. Como ele é necessário ao dimensionamento do Módulo de Sinalização e Sincronismo, o *frame* DIM-MS solicita que o usuário forneça seu valor.

DIM-MS tem como meta dimensionar o Módulo de Sinalização e Sincronismo, devendo para isso determinar o número de módulos necessários e a configuração de placas de cada módulo. Para determinar esses valores, a busca está restrita apenas às regras do *frame* DIM-MS, já que um *frame* só tem acesso às suas regras e às de seus descendentes. Esse mecanismo reduz enormemente o número de regras que devem ser pesquisadas para o dimensionamento do módulo, tornando a execução do sistema mais rápida.

Dentro do *frame* é implementada busca para trás, que é a maneira mais natural de implementação na ferramenta utilizada.

Ao final de sua execução, o *frame* DIM-MS terá determinado os valores dos parâmetros N-MS (número de Módulos de Sinalização) e MS (lista cujos elementos são outras listas, uma para cada módulo; essas sub-listas por sua vez têm como elementos pares relacionando nome da placa e quantidade).

Os *frames* DIM-MC, DIM-MA e DIM-MO são executados a seguir, de maneira similar ao DIM-MS, restringindo a busca às regras do *frame* e fornecendo ao final o módulo dimensionado.

Estando concluído o dimensionamento *hardware* da central, os resultados devem ser fornecidos ao usuário. Para isso, o *frame* DIMENSIONAMENTO ativa o *frame* FORNECER-INTENS-FUNC, que por sua vez ativa o *frame* LISTAR-DIM.

A meta de LISTAR-DIM é listar em vídeo o dimensionamento da central. Para isso ele deve listar o dimensionamento de cada módulo. Para cada módulo é impresso um cabeçalho com o nome da central, o nome da empresa operadora e a tarefa que está sendo realizada. A seguir, é impresso um sub-cabeçalho com o nome do módulo e é iniciada a impressão dos resultados. Para cada módulo, é impresso o número de módulos e a configuração de placas.

Tendo fornecido os resultados, o *frame* DIMENSIONAMENTO atingiu suas metas, e consequentemente assim também o *frame* SISTEMA. Nesse momento, o sistema encontra-se pronto para nova execução.

Uma facilidade oferecida pela ferramenta e ainda não citada são os comandos WHY e HOW. Através do WHY, o usuário pode questionar o sistema a respeito da necessidade de determinada informação. O sistema responde mostrando ao usuário qual o contexto em que a informação é necessária e a regra que necessita daquele valor. O usuário pode perguntar novamente por que a regra citada é necessária, e assim seguir a lógica do sistema até seu início.

Já o comando HOW permite que o usuário questione o sistema sobre como determinado resultado foi obtido. O sistema responderá mostrando a regra que concluiu aquele resultado.

5.4 Conclusão

Neste capítulo discutiu-se o Projeto Físico da macro-função DIMENSIONAR e a implementação de seu protótipo inicial.

Com o objetivo de apresentar resultados práticos já na fase de prototipagem, optou-se por implementar alguns dos *frames* principais no que se refere à quantidade de conhecimento envolvida. Dessa forma, foram implementados os *frames* que dimensionam os módulos de Sinalização e Sincronismo, de Canal Comum, Auxiliar, de Operação, Manutenção e Supervisão e de Memória de Massa.

No próximo capítulo, serão discutidos os resultados obtidos e os trabalhos futuros para expansão do sistema.

Capítulo 6

Resultados e Conclusão

Neste capítulo apresenta-se os resultados decorrentes da implementação do protótipo do SSEA e as conclusões obtidas durante seu desenvolvimento. Testes e os trabalhos futuros necessários à expansão do sistema são também considerados.

6.1 Resultados Obtidos

O protótipo do SSEA foi implementado utilizando-se basicamente o conhecimento existente no Manual de Dimensionamento ([Dime90]) e no programa de dimensionamento *hardware* já existente. Devido às limitações de memória impostas pelo equipamento utilizado, o protótipo cobre apenas um sub-conjunto das funções previstas.

Esse protótipo conta atualmente com 17 *frames*. Desses, 12 referem-se especificamente à tarefa de dimensionamento e os demais realizam entrada e saída de dados. Possui ainda 83 regras.

Foi adotado o seguinte procedimento de teste para validação do protótipo: primeiramente foram selecionados casos de configuração, porém sem ter conhecimento prévio dos resultados. A seguir, os mesmos casos de teste foram executados tanto pelo protótipo do SSEA quanto pelo sistema convencional já existente e os resultados foram comparados. Como o SSEA realiza, hoje, apenas parte do dimensionamento *hardware*, também realizado pelo sistema convencional, e como esse sistema foi implementado por especialista em dimensionamento, a comparação dos resultados permite que se tenha uma idéia da correção dos mesmos.

Seguindo esse procedimento, foram executados 5 casos de teste, cujos resultados encontram-

<i>Caso de Teste</i>	<i>Coincidência</i>
caso 1	100%
caso 2	100%
caso 3	100%
caso 4	80%
caso 5	100%

Tabela 6.1: Coincidência de Resultados – SSEA/Sistema Convencional

se na tabela 6.1.

No caso de teste 4, o valor de 80% encontrado significa que, dos 5 módulos (MS, MC, MA, MO, MM) dimensionados pelo SSEA, 1 apresentou resultados diferentes daqueles apresentados pelo sistema convencional.

Exemplo 34: A seguir, têm-se um exemplo de caso de teste e da resposta fornecida pelo SSEA.

Caso de Teste:

Qual o nome da central ? *Exemplo*

Qual a sigla (mnemônico) da central ? *EXEM*

Qual o nome da empresa operadora ? *CPqD*

Qual a quantidade de registradores ? *100*

Existem serviços suplementares ? *SIM*

Qual o tráfego de entrada ? *1200.00*

Qual o tempo de retenção de chamadas (em segundos) ? *180.00*

Qual o tráfego originado ? *1619.70*

Qual o número de planos de comutação ? *4*

Qual o número de MX por plano ? *4*

Qual o número de rotas de canal comum ? *4*

Resposta fornecida:

MS: 2 Módulos de Sinalização. MS 1: 2 placas DAP, 1 placa CTE, 1 placa FMC, 1 placa RIS, 1 placa CSI, 2 placas CGS, 2 placas ECV. MS 2: 2 placas DAP, 1 placa CTE, 1 placa FMC, 1 placa RIS, 1 placa CSI, 1 placa CGS, 2 placas ECV.

MC: 2 Módulos de Canal Comum, ambos com 1 placa TTS, 1 placa CTE, 1 placa FMT, 4 placas CCO, 2 placas DAP.

MA: 4 Módulos Auxiliares, cada um com 1 placa FMC, 2 placas SPP, 1 placa UPN, 2 placas MEM, 1 placa RTE.

MO: 1 Módulo de Operação, Manutenção e Supervisão, com 1 placa FMC, 2 placas SPP, 1 placa UPN, 1 placa MEM, 1 placa DIS, 1 placa RTE, 2 placas SER, 1 placa FMG.

MM: 1 Módulo de Memória de Massa, com 2 Fontes de Módulo de Memória de Massa, 2 Elementos Unidade de Disco Rígido, 1 Elemento Unidade de Disco Flexível, 1 Elemento Unidade de Fita Magnética-Cartucho.

As respostas fornecidas pelo protótipo, comparativamente às fornecidas pelo sistema convencional, apresentam-se praticamente 100% corretas. A avaliação, no entanto, necessita de refinamento por especialista humano.

Quanto ao tempo de resposta é muito bom. A resposta é obtida praticamente no mesmo instante em que os dados são fornecidos, mesmo levando-se em consideração que o protótipo é apenas interpretado, e não compilado.

6.2 Conclusões

O uso de técnicas de Engenharia de *Software* durante o desenvolvimento do SSEA mostrou-se apropriado e de grande utilidade. O uso de SADT proporcionou uma maior compreensão do problema, necessária para a construção da representação gráfica (os Actigramas). Por outro lado, a técnica de decomposição estruturada dos módulos de cada actigrama permitiu um refinamento gradual das funções desempenhadas e das informações necessárias.

A partir do projeto de sistema, o mapeamento funcional/físico fez-se naturalmente, produzindo um *software* bem estruturado. Essa característica nem sempre está presente em sistemas baseados em conhecimento, onde muitas vezes as regras não têm nenhum tipo de controle. No caso do SSEA, a divisão das regras em *frames* de acordo com o tipo de conhecimento envolvido facilita a manutenção do sistema e melhora sua eficiência, reduzindo o número de regras pesquisadas a cada busca.

A implementação do protótipo teve como objetivos principais demonstrar a viabilidade

técnica do projeto e a adequação do modo de representação do conhecimento e das ferramentas escolhidas.

A viabilidade técnica foi demonstrada, já que o protótipo está implementado e fornecendo resultados satisfatórios, tanto em termos de respostas fornecidas quanto de tempo de execução.

O modo de representação escolhido mostrou-se apropriado, permitindo que o conhecimento seja facilmente codificado em regras.

Quando ao esquema de implementação adotado, existem vários pontos favoráveis, principalmente quanto às facilidades para entrada e saída de dados e à estruturação da base de conhecimento, possível através do uso de *frames*. Os principais pontos negativos encontrados foram as dificuldades para implementação de busca para frente, para manipulação de listas que não sigam um padrão pré-definido e para implementação de recursão. Estes últimos pontos puderam ser solucionados através do uso de LISP, seja diretamente nas regras seja em rotinas.

O equipamento usado foi um micro-computador padrão PC 286 com 640 Kbytes de memória. A implementação do sistema completo e mesmo da macro-função DIMENSIONAR completa é inviável nesse equipamento. Prevê-se que nem mesmo o uso de expansão de memória deve solucionar o problema, já que os *frames* DIM-MX e DIM-MT-ULR, ainda não implementados, são os maiores em termos de conhecimento envolvido. É necessário migrar para equipamento de maior capacidade.

O SSEA confirma a viabilidade do uso de sistemas baseados em conhecimento para a tarefa de configuração, principalmente no caso de equipamentos complexos com grande número de componentes e inúmeras combinações possíveis, como é a situação de centrais telefônicas.

6.3 Trabalhos Futuros

O SSEA, como se encontra atualmente, é apenas um protótipo. Apresenta resultados parciais, e portanto insuficientes para que o sistema possa ser usado pela indústria. Para que isso seja possível, várias etapas adicionais precisam ser cumpridas.

Deverá ser feita a migração para equipamento de maior porte, provavelmente para ambiente UNIX/estações de trabalho. Como consequência, serão necessários novos estudos quanto às ferramentas e *softwares* disponíveis (*shells*, linguagens, compiladores).

Uma vez definido o novo ambiente de desenvolvimento, prevê-se a reiniciação da implementação do sistema. Nessa fase deverá ser definido se o protótipo pode ser reutilizado ou se será descartado totalmente.

Durante a fase de desenvolvimento do protótipo, o conhecimento envolvido na tarefa de dimensionamento também evoluiu. Algumas placas foram substituídas por outras, com conseqüente mudança nas regras de configuração. Será necessário alterar-se a base de conhecimento do SSEA para adaptar-se às novas placas.

A seguir, deverá ser concluída a implementação da macro-função DIMENSIONAR. Para isso o primeiro passo será a implementação dos demais *frames* de DIMENSIONAMENTO. Uma vez implementados todos os *frames*, deverá ser refinado o conhecimento contido no sistema. Esse refinamento exigirá uma participação mais intensa do especialista em dimensionamento, de forma a que possa ser também incluído o conhecimento não contido em manuais. Esse é o trabalho mais elaborado de ser realizado na tarefa de dimensionamento, e também o mais importante, pois só quando esse conhecimento estiver codificado é que o SSEA apresentará respostas realmente *especialistas*.

Finalizada a implementação da macro-função DIMENSIONAR, deverão ser feitos os projetos detalhados das demais macro-funções e suas implementações.

Além disso, poderá ser estudada uma possível ligação do SSEA com o ADS, visando automatizar todo o processo de desenvolvimento do TRÓPICO RA, desde o Projeto de Sistemas até o dimensionamento e configuração das centrais.

Apêndice A

Actigramas

A.1 Índice

A-0 Sistema de Suporte a Engenharia Aplicada (contexto)

A0 Sistema de Suporte a Engenharia Aplicada

A1 Dimensionar

A11 Obter Informações

A12 Realizar Calculos Iniciais

A13 Dimensionar Hardware

A131 Dimensionar Modulos

A14 Dimensionar Software

A15 Fornecer Itens Funcionais

A2 Esquematizar

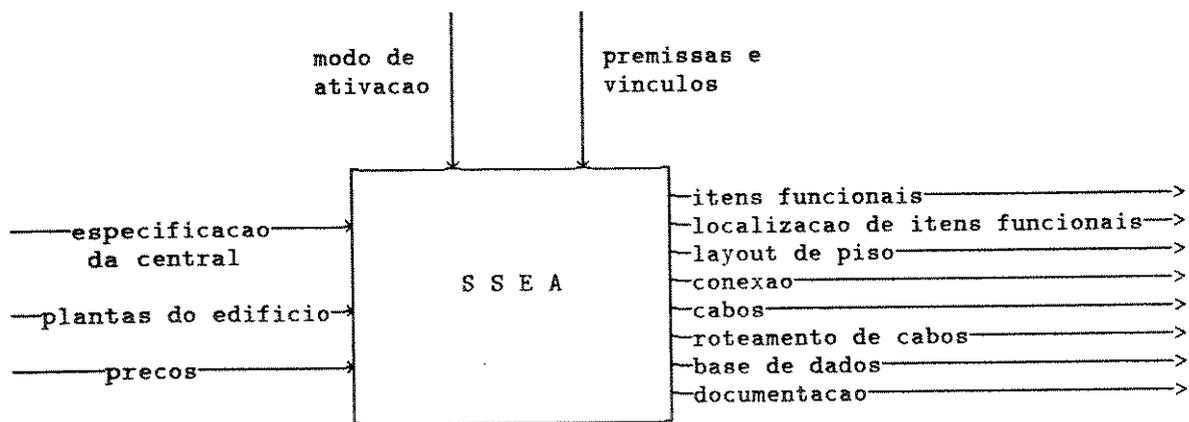
A3 Interconectar

A4 Gerar Base de Dados

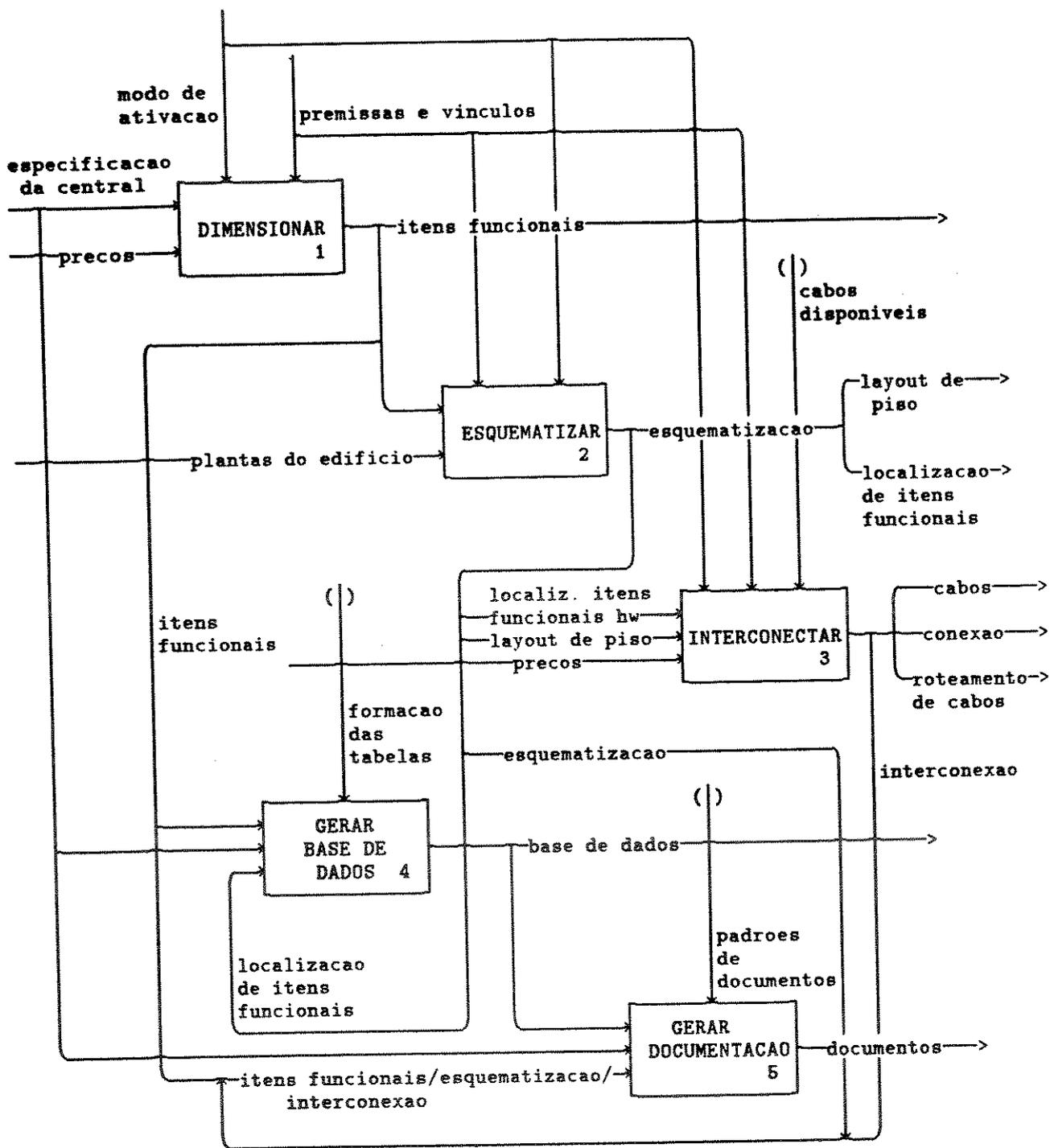
A5 Gerar Documentação

A.2 Actigramas

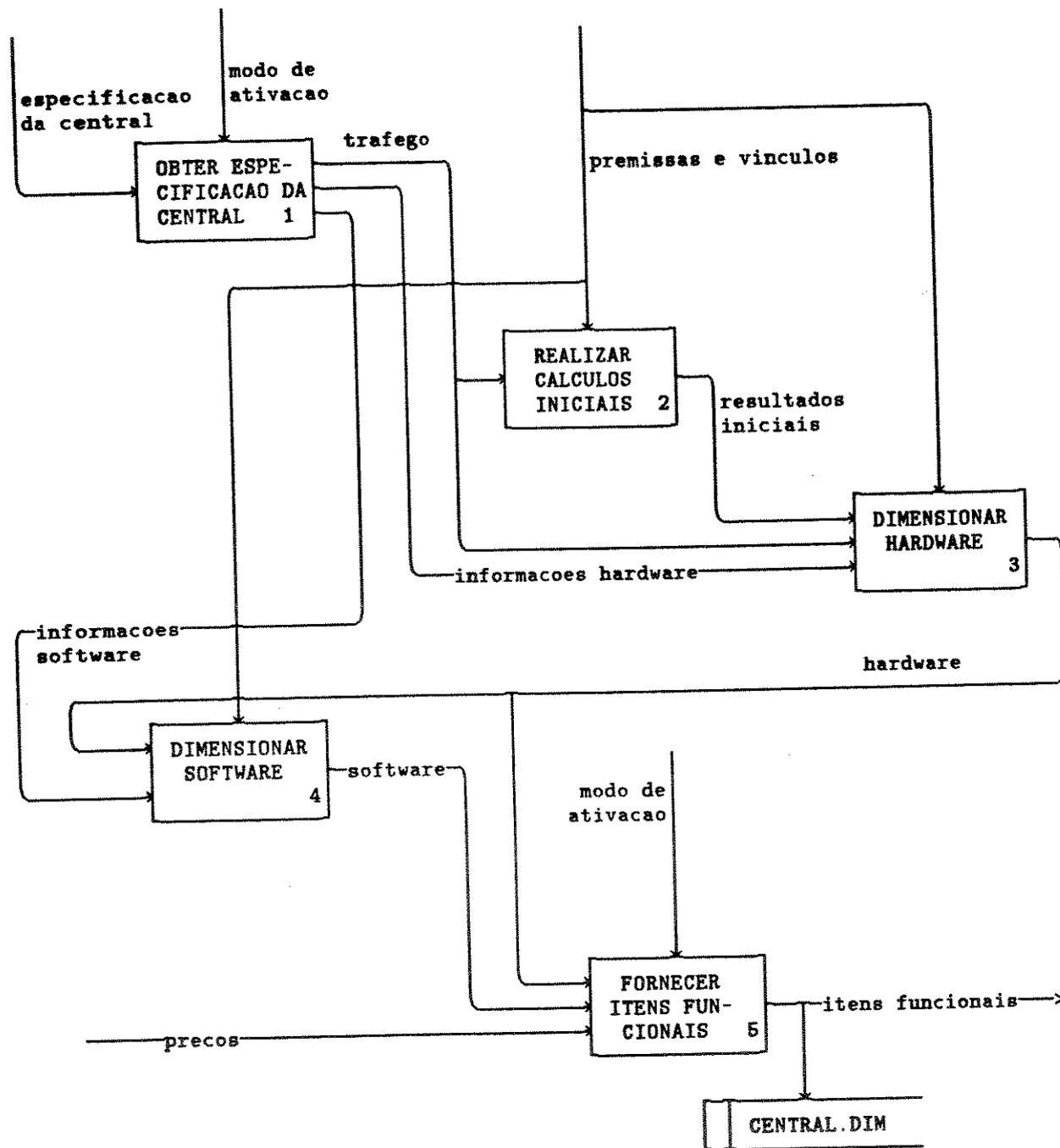
A seguir apresenta-se os actigramas referentes ao contexto do sistema e à macro-função DIMENSIONAR. Para o leitor interessado em conhecer a descrição do sistema completo, recomenda-se a consulta a [Faza 90].



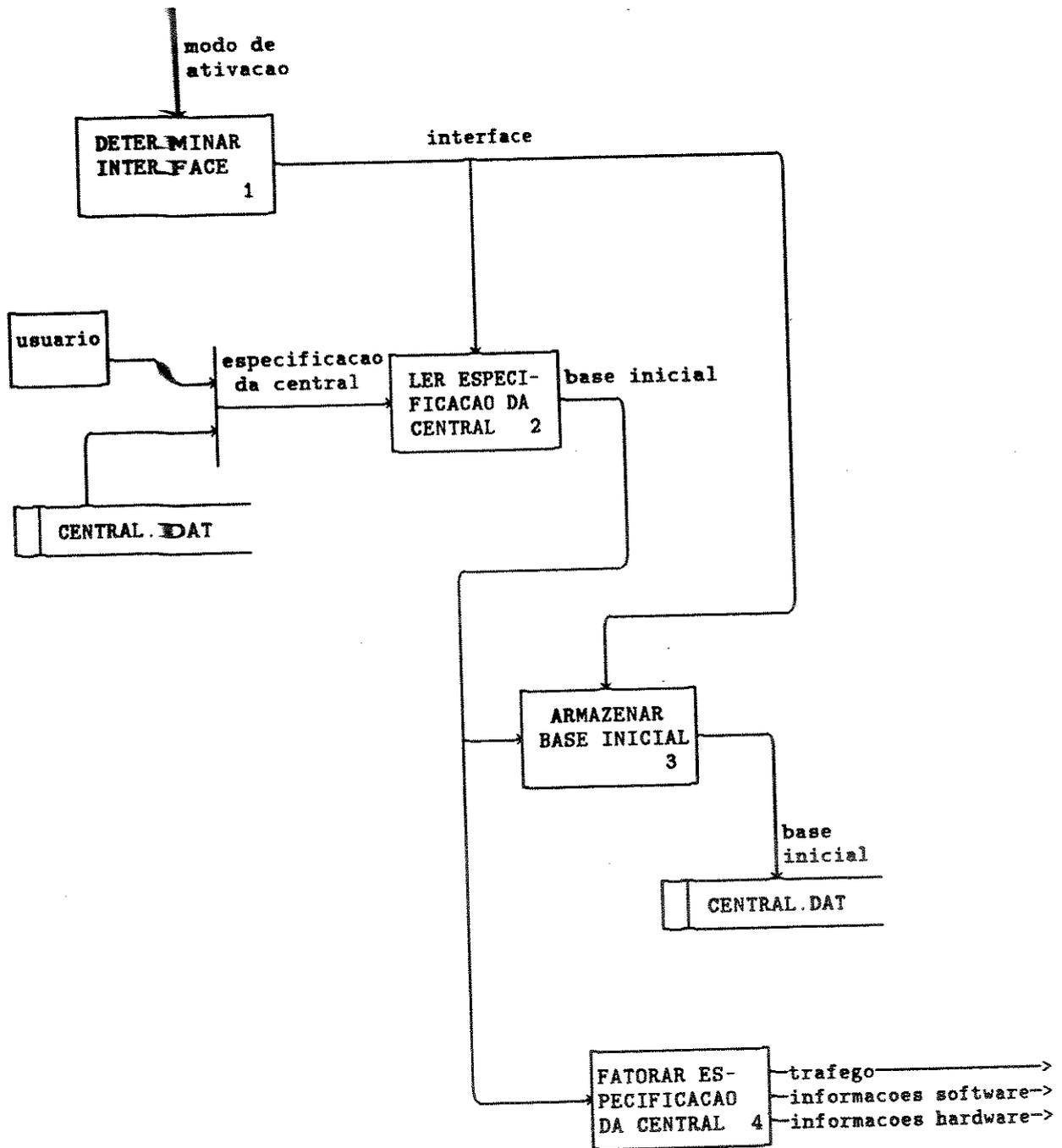
Actigrama A-0 — Sistema de Suporte a Engenharia Aplicada (contexto)



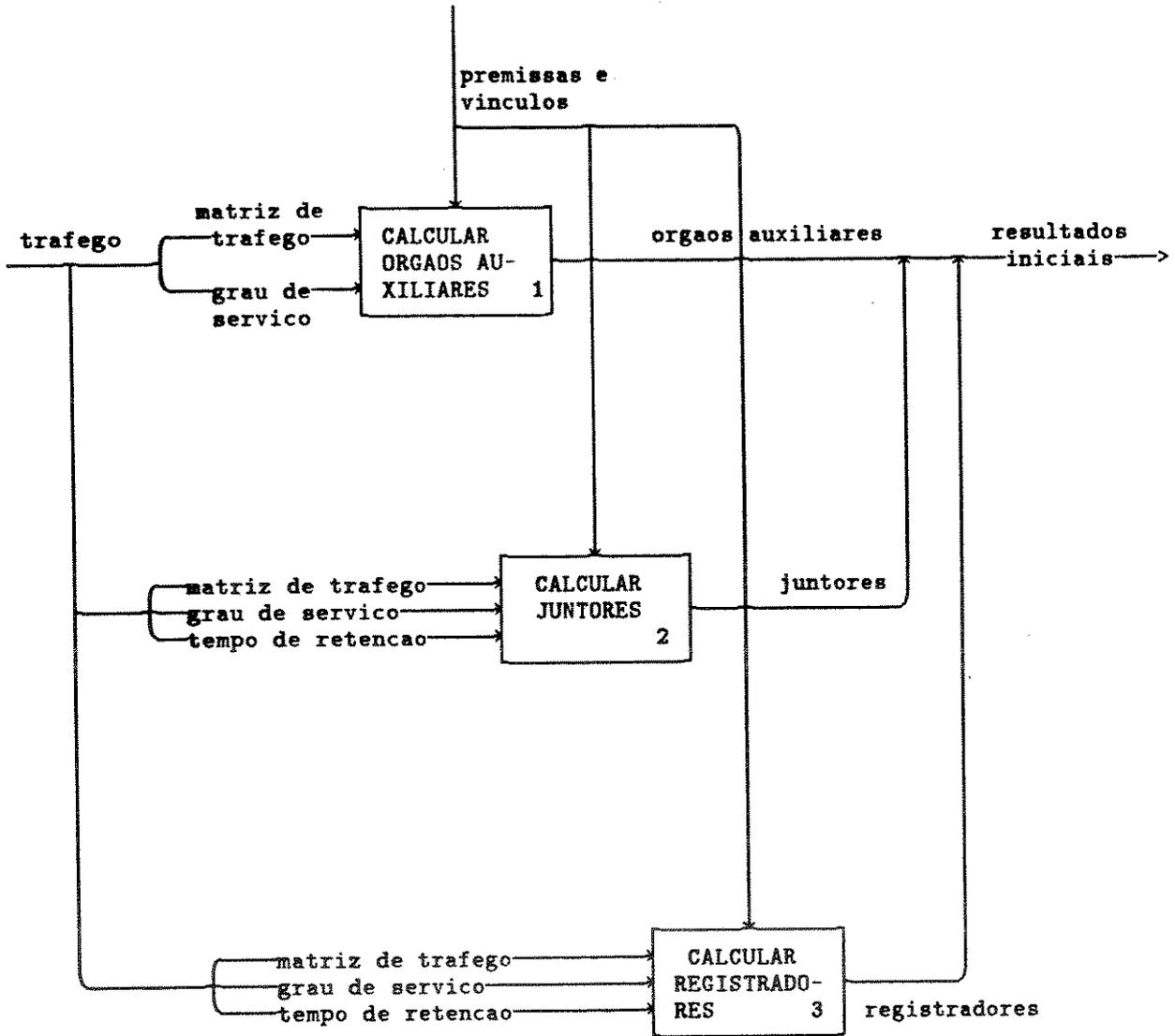
Actigrama AO — SISTEMA DE SUPORTE A ENGENHARIA APLICADA



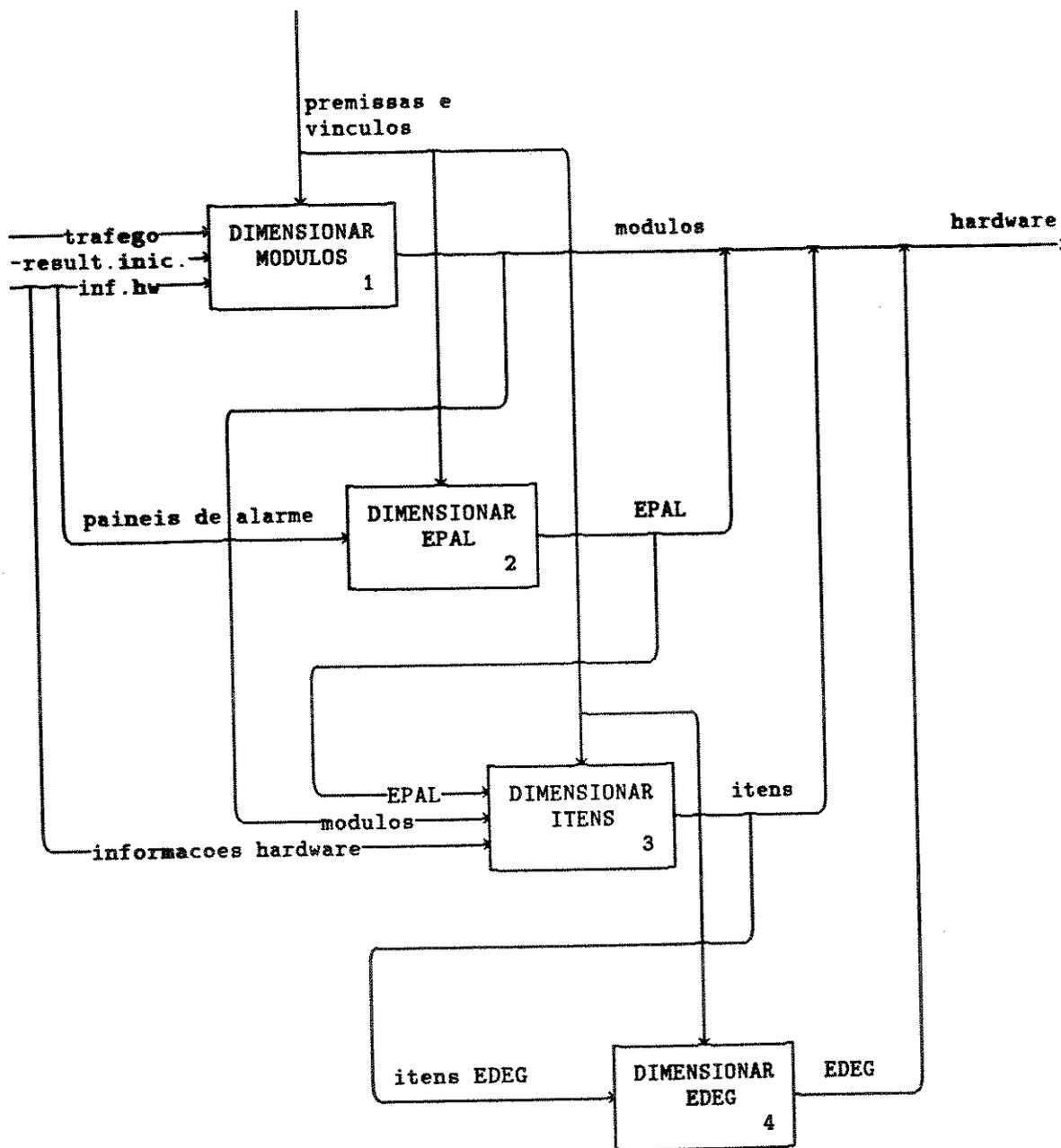
Actigrama A1 — DIMENSIONAR



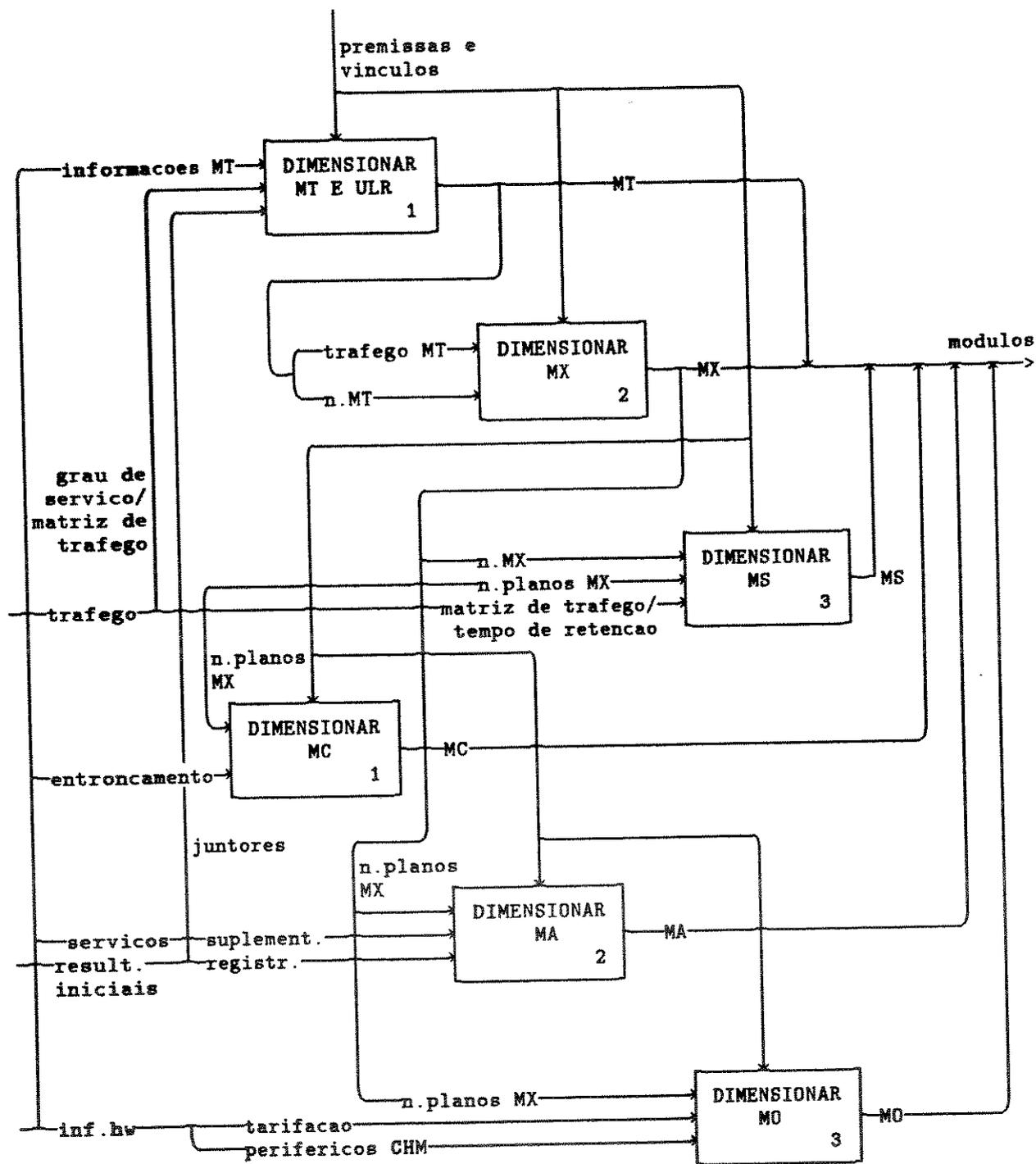
Actigrama A11 — OBTEN ESPECIFICACAO DA CENTRAL



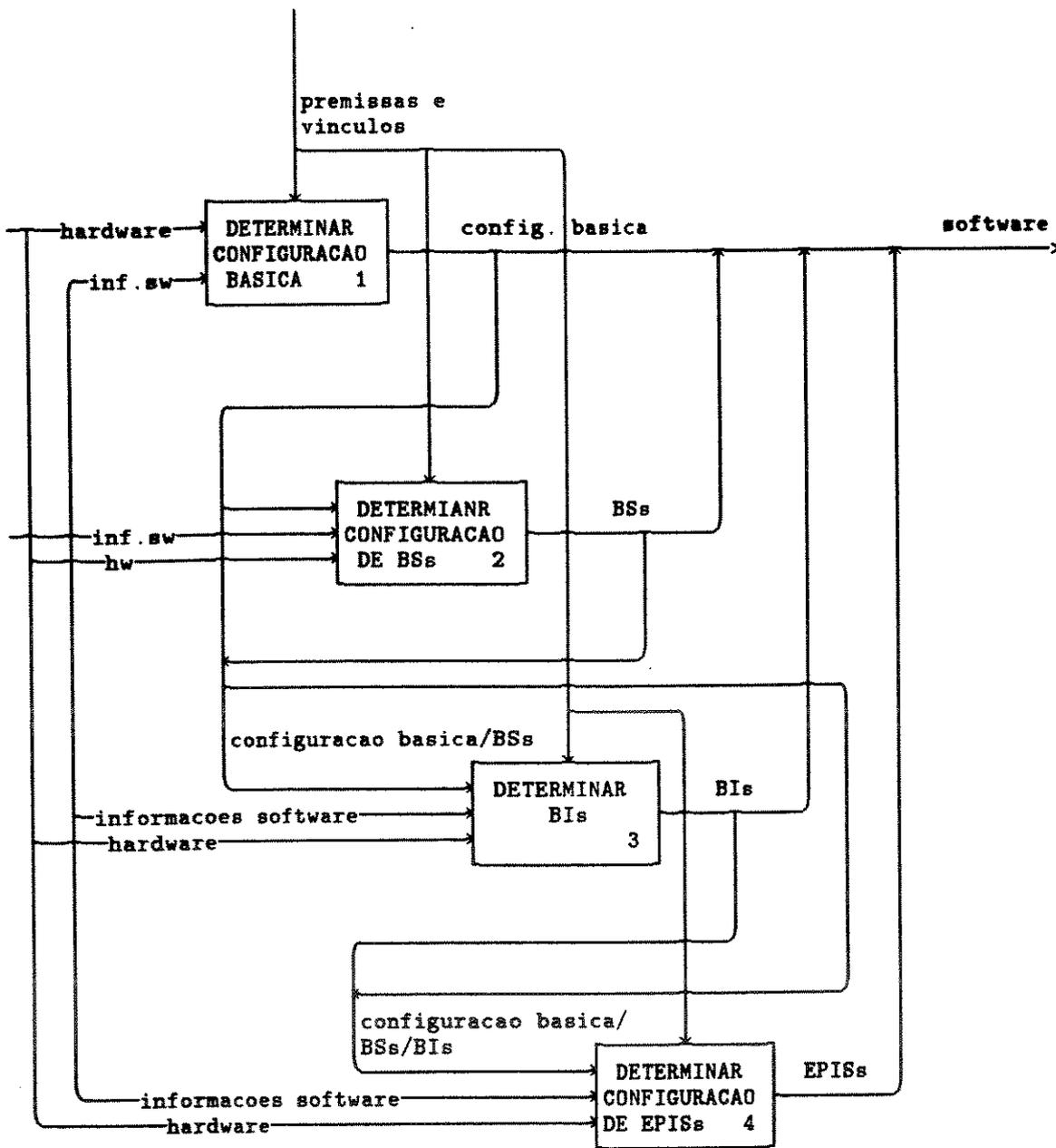
Actigrama A12 — REALIZAR CALCULOS INICIAIS



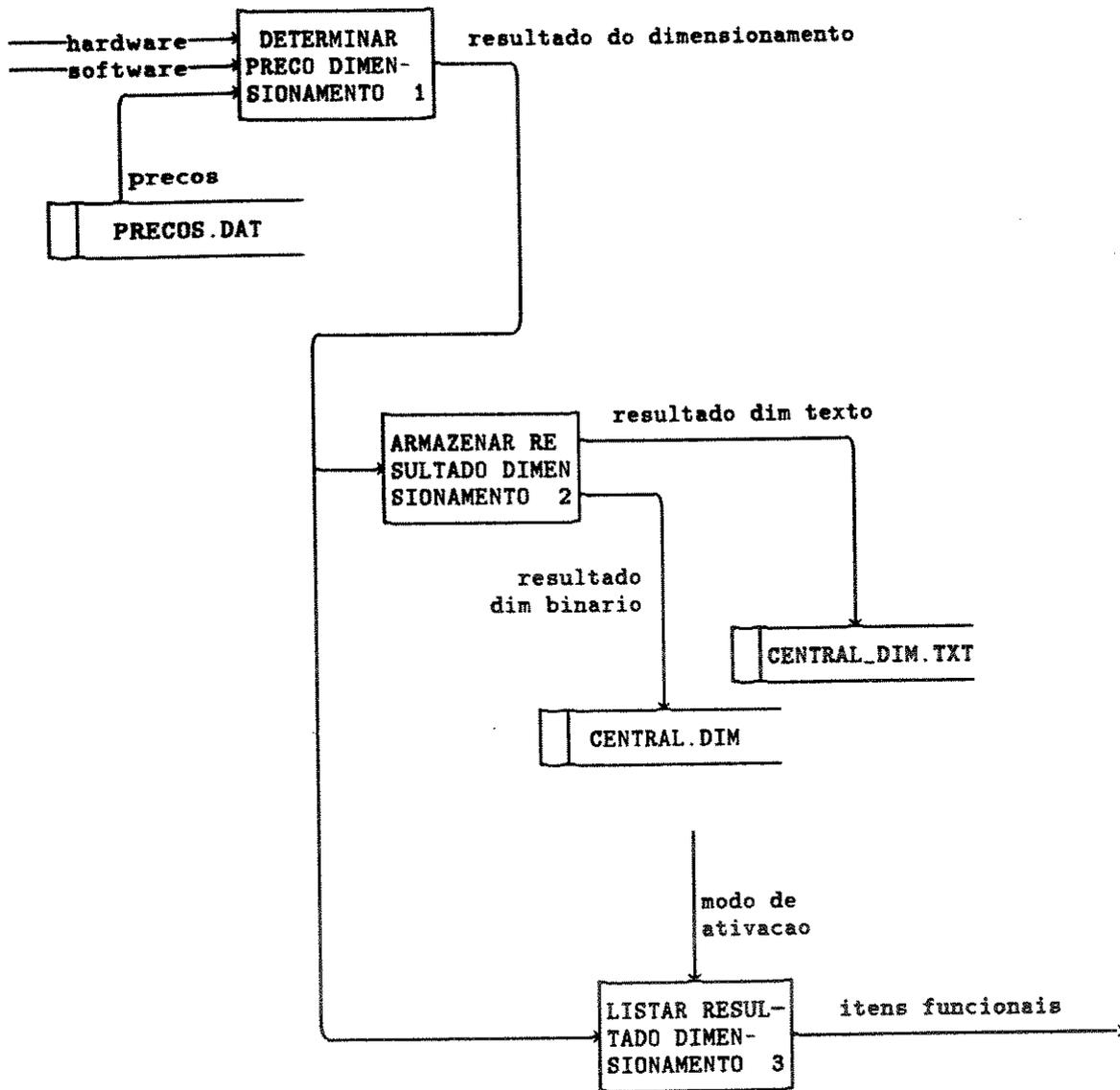
Actigrama A13 — DIMENSIONAR HARDWARE



Actigrama A131 — DIMENSIONAR MODULOS



Actigrama A14 — DIMENSIONAR SOFTWARE



Actigrama A15 — FORNECER ITENS FUNCIONAIS

Apêndice B

Interfaces Internas

Neste apêndice estão reunidos os modelos gráficos de interface entre macro-funções e entre funções da macro-função DIMENSIONAR. Para o leitor interessado nas interfaces das demais macro-funções, recomenda-se a consulta a [Faza 90].

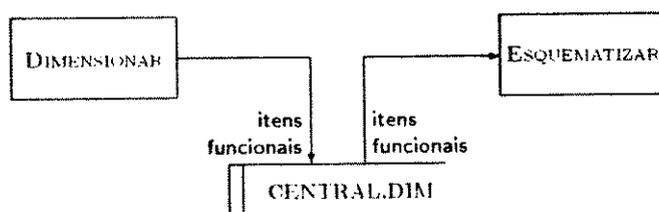


Figura B.1: Interface DIMENSIONAR → ESQUEMATIZAR

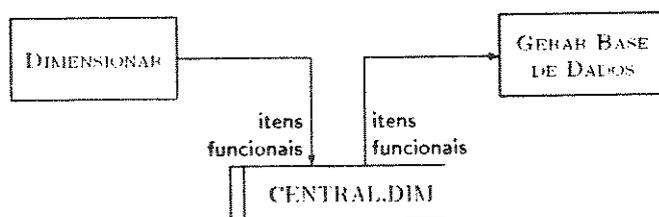


Figura B.2: Interface DIMENSIONAR → GERAR BASE DE DADOS

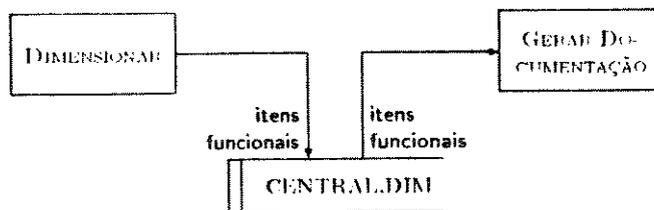


Figura B.3: Interface DIMENSIONAR → GERAR DOCUMENTAÇÃO

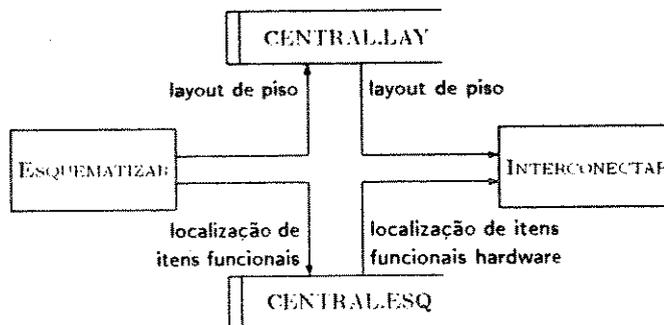


Figura B.4: Interface ESQUEMATIZAR → INTERCONECTAR

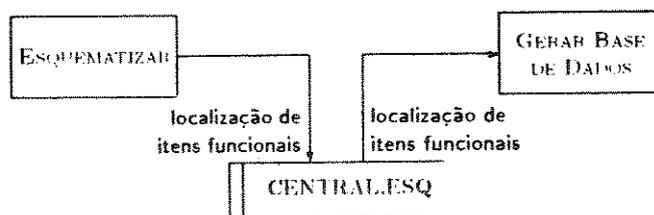


Figura B.5: Interface ESQUEMATIZAR → GERAR BASE DE DADOS

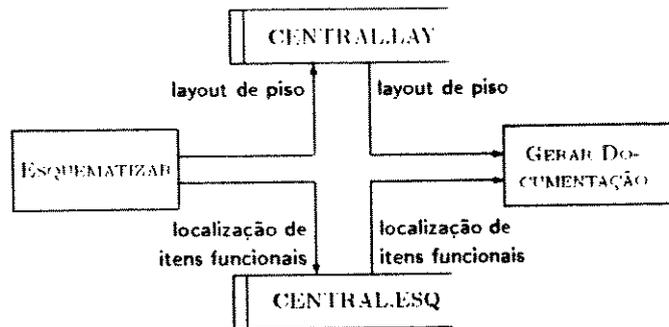


Figura B.6: Interface ESQUEMATIZAR → GERAR DOCUMENTAÇÃO

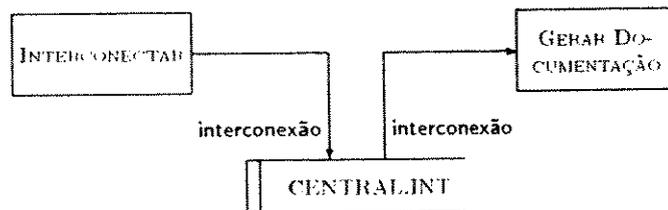


Figura B.7: Interface INTERCONECTAR → GERAR DOCUMENTAÇÃO

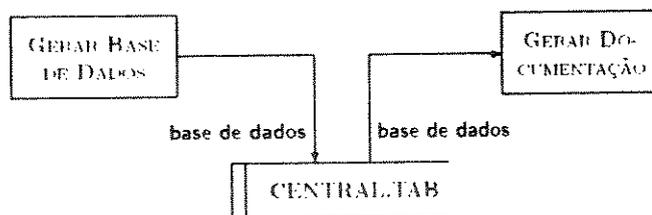


Figura B.8: Interface GERAR BASE DE DADOS → GERAR DOCUMENTAÇÃO



Figura B.9: Interface **OBTEN ESPECIFICAÇÃO DA CENTRAL** → **REALIZAR CÁLCULOS INICIAIS**

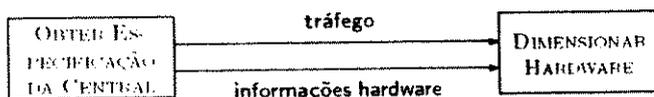


Figura B.10: Interface **OBTEN ESPECIFICAÇÃO DA CENTRAL** → **DIMENSIONAR HARDWARE**

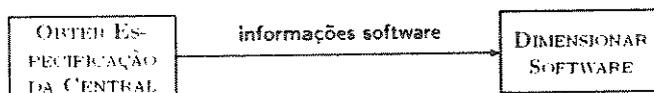


Figura B.11: Interface **OBTEN ESPECIFICAÇÃO DA CENTRAL** → **DIMENSIONAR SOFTWARE**

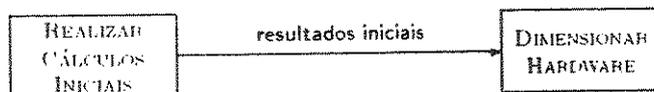


Figura B.12: Interface **REALIZAR CÁLCULOS INICIAIS** → **DIMENSIONAR HARDWARE**



Figura B.13: Interface **DIMENSIONAR HARDWARE** → **DIMENSIONAR SOFTWARE**

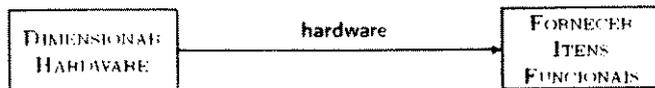


Figura B.14: Interface DIMENSIONAR HARDWARE → FORNECER ITENS FUNCIONAIS

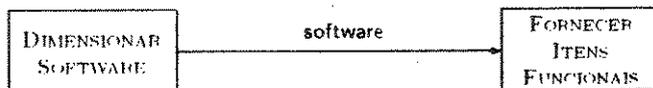


Figura B.15: Interface DIMENSIONAR SOFTWARE → FORNECER ITENS FUNCIONAIS

Bibliografia

- [Akao 89] Akao, C. Y., Imai, K., Tsuchida, K. *Debug Expert System for Switching Systems Software*. NEC Research & Development, n. 95, pp. 107-115. October 1989.
- [Akin 88] Akin, O., Dave, B., e Pithavadian, S. Heuristic Generation of Layouts (HeGel): Based on a Paradigm for Problem Structuring. Em Gero, J. S., ed., *Artificial Intelligence in Engineering: Design*, pp. 413-444, Computacional Mechanics Publication, Southampton, U.K. 1988.
- [Atwo 86] Atwood, M. E., Brooks, R., e Radlinski, E. R. *Causal Models: The Next Generation of Expert Systems*. Electrical Communication, vol. 60(2), pp. 180-184. 1986.
- [Bark 89] Barker, V. E., e O'Connor, D. E. *Expert Systems For Configuration At Digital: XCON And Beyond*. Communications of the ACM, vol. 32(3), pp. 298-318. March 1989.
- [Barr 81] Barr, A. e Feigenbaum, E. A., eds. *The Handbook of Artificial Intelligence*, vol. 1. Morgan Kaufmann, Los Altos, Calif. 1981.
- [Brug 86] van de Brug, A., Bachant, J., e McDermott, J. *The Taming of R1*. IEEE Expert, pp. 33-39. Fall 1986.
- [Call 88] Callahan, P. H. *Expert Systems for AT&T Switched Network Maintenance*. AT&T Technical Journal, vol. 67(1), pp. 93-103. January/February 1988.
- [Char 86] Charniak, E. e McDermott, D. *Introduction to Artificial Intelligence*. Addison-Wesley, Reading, Mass. 1986.

- [DeJe 85] DeJesus, E. J., e Callan, J. P. *PEARL: A Knowledge Based Expert Assisted CAD Tool*. Procc. of the 2nd Conference on Artificial Intelligence Applications, pp. 258-263. IEEE Computer Society Press, Washington, D.C. 1985.
- [Dime90] Manual de Dimensionamento do Sistema TRÓPICO RA. CPqD TELEBRÁS, ELEBRA TELECON, SESA, PHT, STC.
- [Faza 90] Fazano, L. SSEA – Sistema de Suporte à Engenharia Aplicada. Relatório Técnico RT-DCA-017. FEE UNICAMP. 1990.
- [Feig 85] Feigenbaum, E. A. Artificial Intelligence: A Status Report. Em Torrero, E. A., ed., *Next-Generation Computers*. IEEE Press, New York, N.Y. 1985.
- [Flem 88] Flemming, U., Coyne, R., Glavin, T., e Rychener, M. A Generative Expert System For the Design of Building Layouts – version 2. Em Gero, J. S., ed., *Artificial Intelligence in Engineering: Design*, pp. 445-464, Computacional Mechanics Publication, Southampton, U.K. 1988.
- [Fran 89] Frank, W., e Sauve, B. *Expert Systems for Manufacturing and Testing Applications*. Electrical Communication, vol. 63(2), pp. 137-144. 1989.
- [Gunh 86] Gunhold, R., e Zettel, J. *System 12 In-Factory Testing*. Electrical Communication, vol. 60(2), pp. 128-134. 1986.
- [Haye 83] Hayes-Roth, F., Waterman, D. A. e Lenat, D. B., eds. *Building Expert Systems*. Addison-Wesley, Reading, Mass. 1983.
- [Jack 86] Jackson, P. *Introduction to Expert Systems*. Addison-Wesley, Wokingham, England. 1986.
- [Kuzm 88] Kuzmak, S. D., Brittingham, M. D., Gorin, A. L., Milich, G. A., e Shoenfelt, J. E. *Knowledge Based Signal Interpretation*. AT&T Technical Journal, vol.67(1), pp. 104-120. January/February 1988.
- [Lan 90] Lan, M. S., e Panos, R. M. *Printing Press Configuration, A Knowledge Based Approach*. IEEE Expert, vol. 5(1), pp. 65-73. February 1990.
- [Lena 82] Lenat, D. B., Sutherland, W. R., e Gibbons, J. *Heuristic Search for New Micro-circuit Structures: An Application of Artificial Intelligence*. AI Magazine, n. 3, pp. 17-33. 1982.

- [McDe 80] McDermott, J. *R1: An Expert In The Computer System Domain*. Procc. of AAAI-80, pp. 269-271. 1980.
- [McDe 81] McDermott, J. *R1's Formative Years*. AI Magazine, vol.2(2). 1981.
- [McDe 82] McDermott, J. *R1: A Rule-Based Configurer of Computer Systems*. Artificial Intelligence, vol.19(1), pp. 39-88. 1982.
- [McDe 84] McDermott, J. *R1 Revisited: Four Years In The Trenches*. AI Magazine, pp. 21-32. Fall 1984.
- [Oxma 88] Oxman, R., e Gero, J. S. Designing by Prototype Refinement in Architecture. Em Gero, J. S., ed., *Artificial Intelligence in Engineering: Design*, pp. 413-444, Computacional Mechanics Publication, Southampton, U.K. 1988.
- [Ouro 88] Livro Ouro do Sistema TRÓPICO RA. CPqD TELEBRÁS.
- [Pres 87] Pressmann, R. S. *Software Engineering: A Practioner's Approach*. McGraw-Hill, New York, N.Y. 1987.
- [Rich 83] Rich, E. *Artificial Intelligence*. McGraw-Hill, New York, N.Y. 1983.
- [Sche 86] Schelfhout, H. *Customer Application Engineering for System 12 Hardware*. Electrical Communication, vol. 60(2), pp. 135-140. 1986.
- [Than 86] Thandasseri, M. *Expert Systems Application for TXE4A Exchanges*. Electrical Communication, vol. 60(2), pp. 154-161. 1986.
- [Vird 87] Virdhagriswaran, S., Levine, S., Fast, S., e Pitts, S. *PLEX: A Knowledge Based Placement Program for Printed Wire Boards*. Procc. of the 3rd Conference on Artificial Intelligence Applications, pp. 302-305. IEEE Computer Society Press, Washington, D.C. 1987.