

TESE
846

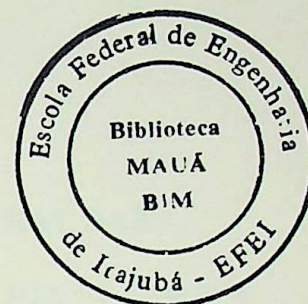
Escola Federal de Engenharia de Itajubá

AMBIENTE INTELIGENTE DE LINGUAGEM
NATURAL PARA RESOLUÇÃO DE PROBLEMAS
DA ÁREA DE ENGENHARIA ELÉTRICA

MARIA SILVANA PERES

Itajubá - 1995

MARIA SILVANA PERES

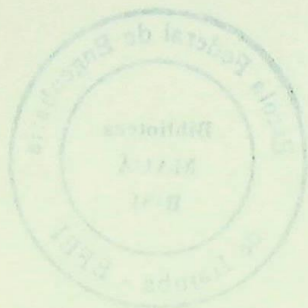


**AMBIENTE INTELIGENTE DE LINGUAGEM NATURAL
PARA RESOLUÇÃO DE PROBLEMAS DA ÁREA DE ENGENHARIA ELÉTRICA**

**Dissertação apresentada à Escola
Federal de Engenharia de Itajubá
para obtenção do título de Mestre em
Engenharia.**

**Itajubá
1995**

CLASS. 681.322:800.92 (043.2)
CUT: n. P 434 a
TOMBO. 846



MARIA SILVANA PERES

ASSISTENTE DE PLANEJAMENTO DE ENSINO SUPERIOR

PARA RESOLUÇÃO DE PROBLEMAS E ÁREAS DE ENSINO SUPERIOR

Diário de Registro de Atividades de Ensino
Lectura de Documentos de Ensino
para obtenção de título de Mestre em
Educação

1992

MARIA SILVANA PERES

**AMBIENTE INTELIGENTE DE LINGUAGEM NATURAL PARA
RESOLUÇÃO DE PROBLEMAS DA ÁREA DE ENGENHARIA ELÉTRICA**

**Dissertação apresentada à Escola
Federal de Engenharia de Itajubá
para obtenção do título de Mestre em
Engenharia.**

**Área de concentração :
Engenharia Elétrica**

**Orientador :
Germano Lambert Torres**

**Co-orientador :
João Bosco Schumann Cunha**

**Itajubá
1995**

Peres, Maria Silvana

Ambiente Inteligente de Linguagem Natural para resolução de problemas da
área de Engenharia Elétrica. Itajubá, 1995.
216 P.

Dissertação (Mestrado) - Escola Federal de Engenharia de Itajubá.
Departamento de Engenharia Elétrica.

1. Introdução 2. Descrição do Problema 3. Linguagem Natural 4. Base de
Conhecimento 5. PROLOG - Programação em Lógica 6. Sistemas Especialistas 7.
Descrição do Sistema 8. Descrição do Programa I. Escola Federal de Engenharia de
Itajubá. Departamento de Engenharia Elétrica.

Aos meus colegas do curso de Pós-Graduação do ano de 1992, que,
incentivaram a continuidade do meu trabalho. À minha mãe, pela sua
inesgotável paciência e compreensão.

A G R A D E C I M E N T O S

Ao amigo e orientador Prof. Dr. Germano Lambert Torres pelas diretrizes seguras e permanente auxílio.

Ao professor, amigo e co-orientador João Bosco Schumann Cunha pelo incentivo e auxílio no desenvolvimento deste trabalho.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq / MCT) pelo apoio financeiro.

Ao apoio técnico dos Engenheiros Guilherme Moutinho Ribeiro e Cláudio Inácio de Almeida Costa.

Ao colega e amigo, Renato Ferreira Fernandes Júnior, que participou no apoio computacional, sem o qual este trabalho não teria tal êxito.

À minha mãe, pela sua paciência e incansável compreensão e apoio.

A todos os colegas de estudo, amigos e familiares que, diretamente ou não, me auxiliaram na elaboração deste trabalho.

SUMÁRIO

Resumo

Abstract

Lista de Abreviaturas

1. INTRODUÇÃO.....	1
2. DESCRIÇÃO DO PROBLEMA.....	4
2.1 - Introdução.....	4
2.2 - O Sistema ESRASE.....	5
2.3 - A Estrutura do Sistema ESRASE.....	5
2.3.1 - Base de Conhecimento.....	6
2.3.2 - Motor de Inferência.....	7
3. LINGUAGEM NATURAL.....	12
3.1 - Introdução.....	12
3.2 - A questão da Linguagem Natural.....	14
3.2.1 - Vantagens.....	18
3.2.2 - Desvantagens.....	18
3.3 - Linguagem Natural e Sistemas Elétricos de Potência - Um exemplo.....	19
3.4 - Processamento da Linguagem Natural em Prolog.....	24
3.4.1 - A gramática em Linguagem Natural.....	25
3.5 - Processamento de Linguagem Natural.....	26
3.5.1 - Análise da Linguagem.....	29

3.5.1.1 - Análise Léxica.....	30
3.5.1.2 - Análise Morfológica e Sintática.....	31
3.5.1.3 - Análise Semântica.....	34
 4. BASE DE CONHECIMENTO.....	 36
4.1 - Introdução.....	36
4.2 - Análise Morfológica.....	37
4.2.1 - Verbo.....	37
4.2.2 - Preposição.....	40
4.2.3 - Conjunção	41
4.2.4 - Artigo.....	42
4.2.5 - Substantivo.....	43
4.3 - Análise Sintática.....	47
4.3.1 - Sujeito.....	47
4.4 - O Prolog e o sistema de Base de Dados.....	50
4.4.1 - Introdução.....	50
4.4.2 - Base de Dados	50
4.4.3 - A Base de Dados Relacional.....	52
4.4.4 - O modelo Relacional.....	54
4.5 - O Modelo Conceitual de Entidades e Relacionamentos.....	56
4.5.1 - Introdução.....	56
4.5.2 - Entidades, Atributos de Entidades.....	58
4.5.3 - Relacionamentos.....	58
4.5.4 - Auto-Relacionamento.....	59
4.5.5 - Particionamento de Conjunto de Entidades.....	60

5. PROLOG - PROGRAMAÇÃO EM LÓGICA.....	61
5.1 Introdução.....	61
5.1.1 - Áreas de Aplicação.....	62
5.1.2 - Principais características.....	63
5.2 - O que é Programação em Lógica.....	64
5.3 - Sintaxe da Linguagem Prolog.....	66
5.3.1 - Identificadores e Variáveis.....	66
5.4 - Fatos e Regras.....	67
5.4.1 - Fatos.....	69
5.4.2 - Regras.....	70
5.4.3 - Linguagens Procedurais "versus" Linguagens Declarativas.....	70
5.5 - Programação em Cláusulas.....	71
 6. SISTEMAS ESPECIALISTAS.....	 73
6.1 - Introdução.....	73
6.2 - Definindo Sistemas Especialistas.....	75
6.2.1 - O Turbo Prolog.....	76
6.2.2 - Regras Heurísticas.....	77
6.2.3 - Busca Heurística.....	79
6.2.4 - Quadro-Negro.....	80
6.3 - Componentes de um Sistema Especialista.....	81
6.4 - Criando um Sistema Especialista.....	83
6.4.1 - Metodologia para desenvolver Sistema Especialista.....	83
6.4.2 - A aquisição do conhecimento.....	86
6.4.2.1 - Modos de aquisição de conhecimento.....	87
6.4.3 - Heurística.....	88
6.5 - Qualidades fundamentais de Sistemas Especialistas.....	89

7. DESCRIÇÃO DO SISTEMA.....	91
7.1 - Introdução.....	91
7.2 - Interface Homem/Máquina : Projeto Prolog.....	92
7.2.1 - O desenvolvimento.....	92
7.3 - Interface com o Usuário.....	93
7.3.1 - Tela de Inserção de Dados.....	94
7.3.2 - Tela de Ajuda ao Usuário.....	95
7.3.3 - Tela de Mensagem.....	98
7.4 - Funções Lógicas.....	100
7.4.1 - Função Análise Léxica.....	100
7.4.1.1 - Rotina de plural das palavras.....	101
7.4.1.2 - Rotina de identificação dos verbos.....	102
7.4.1.3 - Rotina de substantivo (composto).....	103
7.4.2 - Função Análise Morfológica.....	104
7.4.3 - Função Análise Sintática.....	105
7.4.4 - Função Análise Semântica.....	106
 8. DESCRIÇÃO DO PROGRAMA.....	 108
8.1 - A Sentença em Linguagem Natural.....	108
8.1.1 - Detalhando o S-Nominal.....	109
8.1.2 - Detalhando o Verbo.....	110
8.1.3 - Detalhando o Complemento.....	111
8.2 - O Evento Ocorrido / Outros Eventos.....	112
8.2.1 - Topologia da Subestação.....	113
8.2.2 - Associações.....	114
8.3 - A Interface Homem/Máquina.....	114
8.3.1 - Conhecimento da Língua.....	114

8.3.2 - Conhecimento Morfológico.....	115
8.3.3 - Conhecimento Técnico.....	116
8.3.4 - Um Exemplo Aplicativo.....	117

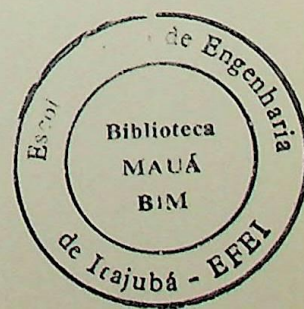
9. CONCLUSÕES.....	122
---------------------------	------------

REFERÊNCIAS BIBLIOGRÁFICAS.....	125
--	------------

ANEXO I	129
----------------------	------------

ANEXO II	170
-----------------------	------------

ANEXO III.....	190
-----------------------	------------



RESUMO

Este trabalho descreve a Interface Homem/Máquina em Linguagem Natural que pode ser aplicado em domínios do problema onde o usuário utiliza da linguagem natural para descrever um evento ocorrido, e tem como responsabilidade registrar o evento ocorrido, assim como, uma lista de parâmetros que o descrevem. Teremos a captação de uma sentença em Linguagem Natural, a análise morfológica e sintática, a interpretação significativa, a correlação dos eventos e a edição de palavras.

Mostraremos algumas vantagens que o usuário leva em usar um sistema especialista, que poderá ser aplicado a várias áreas diversificadas.

O objetivo do aplicativo de Interface Homem-Máquina em Linguagem Natural é o de fornecer mais uma opção ao usuário final de se comunicar com a máquina.

Foi desenvolvido um programa que analisasse a estrutura da frase, em grupos menores de "tokens" e depois em grupos maiores, verificando a gramática que existia por trás do texto. O domínio, que compõe o evento ocorrido formado do adjetivo, verbo, artigo e substantivo, gera eventos que formam uma lista de parâmetros necessários à interpretação da frase, usando, para isto, uma topologia para identificação do evento ocorrido. Desenvolveu-se um programa que interpretasse o texto já classificado e deduzisse outras informações sem que fosse necessário a intervenção do usuário. Os módulos foram agrupados, tornando-se um sistema compacto com uma interface amigável, ficando totalmente transparente sua utilização pelo usuário final.

ABSTRACT

This work describes the Interface Man/Machine into Natural language. It can be applied in problem domains where the user employs from Natural Language (N.L.) to describe an occurred event. Thus, it has as a main responsibility to report the occurred event, as well as, a list of parameters which describes it. We'll have a sentence into Natural language receiving, the Syntax and Morphological Analysis, the meaning interpretation, the Correlation of events and the Words Edition.

We'll show some advantages that the user takes to utilize an expert system, which might be employed into several diversified areas.

The Interface Man/Machine into Natural language system objective is giving one more option to the final user of being able to communicate with the machine.

It was developed a program which could analyse the phrase structure into smaller groups of "tokens" and later on into larger groups, verifying the grammar that there was behind the text. The domain, which sets up the occurred event formed from adjective, verb, article and noun, provides events which form a list of parameters needed to the phrase interpretation, using, for it, a topology for the occurred event identification. It was developed a program that was able to interpretate the text already been classified and deducing some other information without being necessary the user intervention. The modules were grouped, becoming a friendly compact system, becoming fully transparent its use by the final user.

LISTA DE ABREVIATURAS

I.A. : Inteligência Artificial

S.E. : Sistemas Especialistas

ESRASE : Expert System para Restabelecimento Automático de Subestações

RA : Restabelecimento Automático

M.I. : Motor de Inferência

L.N. : Linguagem Natural

SCADA : Supervisory Control and Data Acquisition

M.L.G. : Gramática Lógica Modular

L.F.L. : Linguagem de Forma Lógica

DET : Determinador (artigo)

SEP : Sistemas Elétricos de Potência

E/S : Entradas/Saídas

REA : Relação de Entradas Analógicas

B.D. : Base de Dados

M.E.R. : Modelo Entidade Relacionamento

PROLOG : Programação em Lógica

D.F.D. : Diagrama de Fluxo de Dados

CAPÍTULO 1

INTRODUÇÃO

Um dos campos que mais tem se destacado atualmente na área da ciência da computação é a Inteligência Artificial(I.A.). Ela tem permitido o desenvolvimento de programas para situações que anteriormente não podiam ser resolvidas com sucesso pelas técnicas tradicionais de programação ou que tinham uma solução de difícil implementação e não otimizada.

Logo após a sua criação em meados da década de 50, esta ciência passou por um período de poucas realizações, pois estavam sendo implantadas, sem os devidos cuidados, técnicas que visavam explorar a força de cálculo dos computadores.

O ressurgimento da Inteligência Artificial aconteceu no começo da década de 70, com o advento dos Sistemas Especialistas(S.E.). Estes sistemas visavam explorar uma árvore de soluções privilegiando os caminhos com maior probabilidade de sucesso. Diversas aplicações desses sistemas frutificaram durante as duas últimas décadas.

Por outro lado, um dos campos de grande interesse na computação é a criação de interfaces cada vez mais amigáveis, onde o usuário de um determinado sistema conheça

cada vez menos a ferramenta de desenvolvimento e cada vez mais as potencialidades do aplicativo. Para tal, sistemas baseados em menus e janelas têm estado em voga. O próximo passo é tentar fazer a comunicação homem-máquina (ou seja, usuário-aplicativo) o mais natural possível. Para isto, é conveniente que o programa computacional possa entender o usuário em sua linguagem corrente, ou seja, possa entender seus jargões, expressões e siglas utilizadas no seu dia-a-dia.

Esta dissertação apresenta uma contribuição na área de Processamento de Linguagem Natural em programas de Engenharia de Potência. Ela apresenta o desenvolvimento de estruturas de linguagem e de compreensão que foram incorporadas a um programa computacional. Este trabalho apresenta tais estruturas, bem como uma análise de sua eficiência e dos problemas de implementação. O programa computacional foi testado utilizando-se um aplicativo desenvolvido no âmbito do Grupo de Inteligência Artificial da EFEI, em conjunto com a Companhia Energética de Minas Gerais (CEMIG), denominado ESRASE (Sistema Especialista para o Restabelecimento Automático de Subestações). Este aplicativo tem a finalidade de auxiliar os técnicos e operadores da subestação da CEMIG durante a falta de energia em suas subestações devido a problemas internos ou externos.

A utilização da linguagem natural neste aplicativo visa facilitar a comunicação dos usuários durante o processo de consulta. Ele poderá fornecer os dados em linguagem corrente e obter as informações e a lista de ações através de frases em português. Isto acontecerá através de dois programas de comunicação nas direções homem-máquina e

máquina-homem. O primeiro programa tem a finalidade de traduzir para o contexto do ESRASE, informações e entradas de dados, bem como complementar estes dados utilizando a estrutura física da subestação a ser controlada.

O segundo programa tem a finalidade de converter, do contexto ESRASE para a linguagem portuguesa, as ações e procedimentos de restauração da subestação, possibilitando a explanação ao usuário dos diversos passos a serem tomados e de possíveis medidas de segurança.

Esta dissertação está dividida em 9 capítulos assim distribuídos. O Capítulo 2, Descrição do Problema, apresenta uma breve introdução ao Sistema ESRASE; enquanto os Capítulos 3 e 4, Linguagem Natural e Base de Conhecimento, apresentam o embasamento teórico do trabalho. Os Capítulos 5 e 6 mostram os fundamentos de Programação em Lógica e dos Sistemas Especialistas. A Descrição do Sistema Proposto encontra-se no Capítulo 7, enquanto sua estrutura e implementação é tratado no Capítulo 8, denominados Descrição do Programa.

Este trabalho é complementado por um capítulo de Conclusões e ainda por três anexos explicativos dos modelos utilizados e de um artigo publicado.

CAPÍTULO 2

DESCRIÇÃO DO PROBLEMA

2.1 - INTRODUÇÃO

O fornecimento de energia elétrica nos dias atuais, devido ao imenso progresso tecnológico que estamos vivenciando, tornou-se imprescindível. Deve-se, a cada dia, criar projetos que sejam preventivos e preditivos a uma possível falha no sistema elétrico. Pensando nisso foi desenvolvido o 'Expert System para o Restabelecimento Automático de Subestações', denominado ESRASE[1].

O Sistema foi desenvolvido baseado em uma subestação típica, e foi tomado o cuidado de produzir o Sistema em módulos, tornando-o flexível a novas inclusões de módulos que viessem a completá-lo, ou mesmo adaptá-lo a outras subestações.

Para simular o sistema, o ESRASE precisa das entradas de dados via arquivo. A montagem desse arquivo exige um padrão da linguagem que não caberia ao usuário final conhecer. Foi criado um sistema que fizesse a conexão entre a máquina e o usuário, que poderia ser qualquer funcionário da empresa. Para que o sistema se tornasse mais transparente possível optou-se pela Linguagem Natural.

2.2 - O SISTEMA ESRASE

O ESRASE foi desenvolvido como uma ferramenta de auxílio na operação de uma subestação de energia, visando uma maior otimização das funções, objetivando uma maior confiabilidade dos serviços prestados.

O Sistema é baseado na tecnologia de Sistemas Especialistas. Ele analisa uma ocorrência de algum defeito na subestação, estabelece e executa um Plano Diretor de Ações de Restabelecimento, baseado em um conjunto de regras de restabelecimento (que varia de acordo com a subestação) e as variações dos estados dos componentes da subestação que estão sendo monitorados.

2.3 - A ESTRUTURA DO SISTEMA ESRASE

O Sistema foi desenvolvido em uma estrutura modular, visando uma possível inclusão de novos módulos que viessem a completar o ESRASE [2].

Os módulos principais são :

- Base de Conhecimento;
- Motor de Inferência;
- Base de Dados; e
- Módulo de Interface.

2.3.1 - Base de Conhecimento

Todo conhecimento dos componentes da subestação, desde a inicialização do sistema até os dados medidos temporariamente, e todo o conhecimento heurístico que o especialista da subestação reuniu, estão na Base de Conhecimento do Sistema ESRASE[3].

Esta foi dividida em duas bases :

- os fatos, que são o conhecimento da topologia do sistema, seus dados estáticos e dinâmicos iniciais;

Exemplo :

conexão ("2", "K", "B", "K").

proteção ("2", "K", "21-1", "off")

- as regras, que são os esquemas de proteção, intertravamento e de manobra, de localização de faltas, e os critérios de restabelecimento.

Exemplo :

Um algoritmo :

Se atuou proteção diferencial (87) de um transformador
então isolar o transformador

Em Prolog :

filosofia-proteção (N, CLASSE, "87", __) : -

Ação-ra (proteção, N, CLASSE, isolar ("trafo"))

2.3.2 - Motor de Inferência

Uma ocorrência para o Sistema ESRASE é constituída dos dados que caracterizam uma falha na subestação.

Esses dados são representados por :

- lista dos equipamentos de manobra que foram alterados;
- lista dos equipamentos desenergizados;
- lista dos equipamentos de proteção atuados;
- lista dos componentes que saíram fora da faixa.

A esse conjunto de listas denominamos eventos.

A sequência lógica de funcionamento do motor de inferência é a seguinte :

- INICIALIZAR : ler arquivos pré e pós falta.
- GERAR CENÁRIO : identificar as ocorrências.
- GERAR FATOS : gerar lista de circuitos afetados.

- GERAR AÇÃO RA : gerar lista de ações e orientações definindo o plano inicial de Restabelecimento.
- OTIMIZAR : otimizar o Plano Inicial de Restabelecimento segundo critérios de simplificação.
- ORDENAR : ordenar o Plano de Restabelecimento dentro de uma hierarquia estabelecida, constituindo o Plano Diretor de Restabelecimento.
- EXECUTAR : executar o Plano Diretor de Restabelecimento, seguindo os procedimentos operativos de manobra e restabelecimento.

O Motor de Inferência procura identificar a falha no sistema pela análise das grandezas monitoradas do sistema pré e pós-falta, armazenados em uma lista de fatos que compõem o "evento". E, a partir desses eventos, o Motor de Inferência gera recomendação de ações de restabelecimento em função dos critérios pré-estabelecidos[4].

Exemplo : No caso de atuação da proteção 1T87, teremos como consequência a abertura do disjuntor da alta (4K4) e do disjuntor da baixa (12F4), ocasionando a desenergização dos transformadores T1 e T2, e com consequente indicação nas medidas analógicas monitoradas (Tensão e Frequência) de todos os componentes envolvidos.

- ação recomendada pelo(a) : **eqpto**

Circuito 1T Ação : restabelecer("trafo")

- ação recomendada pelo(a) : **proteção**

Circuito 1T Ação : isolar("trafo")



- ação recomendada pelo(a) : **medida**

Circuito **12F** Ação : fechar("4")

- ação recomendada pelo(a) : **chave_manobra**

Circuito **12F** Ação : fechar("4")

- ação recomendada pelo(a) : **medida**

Circuito **4K** Ação : fechar("4")

- ação recomendada pelo(a) : **chave manobra**

Circuito **4K** Ação : abrir_todos_disjuntor_geral_baixa

- ação recomendada pelo(a) : **chave_manobra**

Circuito **4K** Ação : abrir("elac")

Além de outras 26 recomendações para cada um dos componentes que foram consequentemente desenergizados do tipo :

- ação recomendada pelo(a) : **eqpto**

Circuito **1F** Ação : restabelecer("alimentador")

- ação recomendada pelo(a) : **medida**

Circuito **1F** Ação : fechar("4")

A esse conjunto de ações recomendadas, chamou-se de "Plano Inicial de Restabelecimento".

De posse desse "Plano Inicial de Restabelecimento", o ESRASE compara as recomendações entre si, simplificando e eliminando dentro de critérios pré-definidos. No exemplo da ocorrência citada, as ações recomendadas pela **medida** e pela **chave_manobra** do circuito 12F são idênticas, então uma delas é eliminada. No caso das ações recomendadas para o circuito 1T, a ação **isolar("trafo")**, conflita com a ação **restabelecer("trafo")**. Neste caso, por causa de um critério estabelecido, **isolar("trafo")** elimina **restabelecer("trafo")**.

Após esta etapa, o ESRASE hierarquiza e ordena as ações recomendadas resultantes, dentro de diretrizes estabelecidas. No exemplo da ocorrência no transformador, as 26 ações recomendadas pelo "Plano Inicial de Restabelecimento", se reduziriam às seguintes :

- ação recomendada pelo(a) : **proteção**

Circuito **1T** Ação : isolar("trafo")

- ação recomendada pelo(a) : **eqpto**

Circuito **2T** Ação : restabelecer("trafo")

- ação recomendada pelo(a) : **medida**

Circuito **BF** Ação : boot("colapso de tensão")

Assim, o "Plano Inicial de Restabelecimento" é otimizado e ordenado gerando o "Plano Diretor de Restabelecimento".

O passo seguinte do ESRASE é a execução deste "Plano Diretor de Restabelecimento", onde cada ação recomendada é decomposta em um conjunto de ações primárias, denominado "Plano de Manobra", para então ser executada.

CAPÍTULO 3

LINGUAGEM NATURAL

3.1 - INTRODUÇÃO

Temos a linguagem natural como uma das formas mais humanas de manifestação externa da atividade mental. A linguagem é um meio muito rico para comunicação entre pessoas. Comunicação esta, que pode ser de idéias, emoções, desejos, experiências e outras atividades mentais. Formas, como : posturas, gestos, ações também podem exprimir emoções e raciocínio, mas é na linguagem que vemos a marca de humanidade[5].

Este aspecto da linguagem será interessante no estudo de alguns métodos de implementar, com restrições, a forma de comunicação entre pessoas e máquinas.

Um dos problemas na comunicação homem-máquina, em linguagem natural, será a implementação de uma forma de interpretação automática próxima a uma expressão em linguagem humana.

Interpretar a linguagem natural, aqui, é transformar para outra linguagem adequada, usada pela máquina, sempre procurando manter a intenção do usuário.

Dificuldades aparecerão na forma da linguagem interna da máquina escolhida. A linguagem interna deverá ser expressiva e será um dos pontos mais importantes da implementação do interpretador. A linguagem interna funcionará como um sistema simbólico de codificação de significados dependente da aplicação. A linguagem natural escrita, que é a língua portuguesa, é a linguagem de nosso interesse.

Em interfaces simples, pode-se usar uma lista de padrões fixos, onde se compara com a entrada, segundo algum critério. As listas são muito extensas e inviáveis para uma interface com sintaxe menos restrita e vocabulário de tamanho mediano, devido ao fenômeno de explosão combinatória na interação entre os componentes das frases. Se for identificado, um dos padrões como equivalente, a frase será interpretada.

A interface da Linguagem Natural requer um ambiente de facilidades computacionais de um lado, e de operações pessoais do outro. Este ambiente interessa a processos de transmissão e administração de dados, processamento de dados, e representação de modelos.

As aplicações da Linguagem Natural são mais facilmente implementadas quando a interface é direcionada a uma área técnica bem específica. Isso faz com que a variedade de possibilidades de vocabulário, bem como estruturas gramaticais, sejam reduzidas, tornando possível a compreensão dos comandos. Neste trabalho, a linguagem natural foi empregada para aquisição de dados referentes aos eventos ocorridos no sistema existente. Portanto, a

base de dados foi carregada com as palavras mais usadas neste meio técnico, bem como as estruturas gramaticais escolhidas de modo a atender as frases.

Internamente entradas e saídas (E/S) marcam o uso de um armazenamento de fatos reunidos por uma base de dados, pela interface do usuário. Este armazenamento é chamado "memória de trabalho", "base de dados dinâmica" ou "inferências do quadro-negro". Por determinação de segurança e diagnóstico de falhas deve-se ter uma alimentação imediata de informações, usando uma estrutura de dados que concorde com a parte externa.

Um dos primeiros sistemas de processamento de linguagem natural foi o programa ELIZA [6], usado em pesquisas de Inteligência Artificial.

3.2 - A QUESTÃO DA LINGUAGEM NATURAL

A interface com o usuário de um programa é um dos pontos fundamentais a serem analisados. Existem várias formas de facilitar a interação do homem com o computador. Dentre elas podemos citar : ambientes amigáveis, menus, sistemas de ajuda (help) sensíveis ao contexto, interfaces gráficas, uso de dispositivos apontadores (mouse), telas sensíveis ao toque, etc. Entretanto, a maneira mais natural de comunicação é a linguagem. É de grande interesse o estudo de técnicas que possibilitem ao computador compreender frases escritas, e destas extrair as informações necessárias.



Pode-se dizer que um programa de linguagem natural é um programa capaz de transformar uma frase, ou um texto, escrito em português (por exemplo) em uma estrutura lógica compreensível e processável por um computador. Após esta tradução, o programa deve ser capaz de interpretar as informações adquiridas e usá-las de maneira adequada para os seus objetivos.

A linguagem natural pode ser caracterizada como [7] :

- permite declarações em vários níveis;
- é altamente combinatorial;
- as declarações formam uma sequência compacta de palavras;
- as declarações são altamente redundantes;
- as declarações são sequenciais;
- pode ser usada para expressar declarações próprias;
- têm um conjunto limitado de palavras e regras, pelo qual se formam declarações;
- compreende pontuação, letras e ideogramas.

Os programas em Linguagem Natural (L.N.) apresentam a vantagem de simplificar a interface do usuário, permitindo que se realize diversos objetivos utilizando um único comando.

As linguagens declarativas, como as aplicadas para o desenvolvimento de E/S, permitem facilidades para formulação de variáveis, e orações na forma de strings, que se

aproximam da linguagem natural. Assim o nível de comunicação com o programador será comum, onde, o código de dados tem o mesmo sentido para todos, formando uma linguagem comum.

A linguagem natural permite que o operador abrevie os nomes, isto alivia seu processamento, telas e condições de armazenamento da base de dados. As declarações são básicas, simplificando formações de análises.

A linguagem natural não é somente uma questão de uma interface de E/S para o operador, mas de integração de E/Ss, programas de controle, operadores e serviço de dados dentro de um sistema.

- L.N. de saída : as facilidades da linguagem natural são usualmente implementadas na forma de textos. Os textos são retirados por regras, e reduzidos a breves comentários ou comandos, da base de conhecimento, e apresentados na tela.

E/Ss necessitam de uma apropriada interface usuário.

- L.N. de entrada : permite o uso de menus ou boxes, oferecendo um conjunto de opções na escrita e formulação das frases.

Em geral uma interface simples será composta de dois módulos. Um primeiro módulo, essencial, será o tradutor de linguagem natural para interpretação da frase para a representação interna. O outro módulo fará o acesso a aplicação, à partir desta

interpretação e de seu conhecimento pragmático do funcionamento da aplicação, sendo este totalmente dependente da aplicação considerada[8].

Teremos um banco de dados que estará implementado previamente, podendo rodar a aplicação e a interface em linguagem natural em ambientes separados, se não tivermos acesso ao código fonte do aplicativo. Ou então a interface poderá ser integrada com o aplicativo, com potenciais vantagens quanto ao aspecto de eficiência. O banco de dados será implementado em PROLOG, como um conjunto de fatos. As consultas ao banco de dados serão portanto "*queries*"(consultas) em PROLOG.

O processamento de linguagem natural é a técnica para levar o computador a compreender as linguagens humanas. Os cientistas que estudam o processamento de linguagem natural esperam criar hardware e software que lhes permitam consultar o computador na forma de perguntas em português. Desta maneira o computador passará a seguir as instruções. Hoje, para que possamos atender a esta finalidade usaremos comandos próprios da linguagem.

As solicitações serão feitas em Linguagem Natural. Isso significa que haverá compreensão de perguntas ou comandos digitados em Português e, por isso, será considerado como sendo um sistema dotado de Inteligência Artificial. A construção do programa estará baseada na busca de memória, dando, assim, uma tolerância mínima a erros. O sistema terá um conhecimento da gramática portuguesa. A análise sintática realizada será do tipo "top-down". Deverá interpretar as solicitações do usuário,

identificando todos os termos que formam a frase e classificando-os, sendo esta, escrita em linguagem natural e, à partir daí, executar a ação solicitada pelo usuário.

Temos que, é de vital importância a interação *HOMEM-HOMEM* (pesquisador/informante) para se fazer evoluir a interação *HOMEM-MÁQUINA* com interface em linguagem natural [7].

3.2.1 - Vantagens

O uso de programas em L.N. :

- Simplificam a interface do usuário, facilitando a operação;
- Permite ao usuário realizar vários objetivos com um único comando.

3.2.2 - Desvantagens

O uso de programas em L.N. :

- Ocupam muito espaço de memória;
- Devido a grande quantidade de informações, alguma velocidade é perdida;

- A chance de um programa em LN interpretar um comando incorretamente, depende da complexidade dos códigos escritos para interpretar o seu comando.

3.3 - LINGUAGEM NATURAL E SISTEMAS ELÉTRICOS DE POTÊNCIA

- UM EXEMPLO -

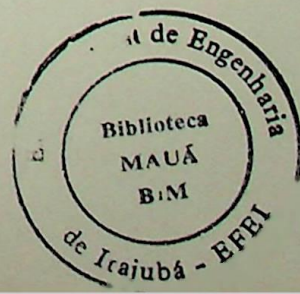
Esta seção apresenta um exemplo retirado de [9].

Em Sistemas Elétricos de potência, uma aplicação ideal de E/S será integrada dentro de facilidades que não necessitam preparação dos dados, nem que o operador se "mova sobre" uma interface especial.

Uma proveitosa aplicação de E/S melhorará a situação para o operador, obtendo mais resultados. Isto significa que as tarefas requeridas ao operador serão distribuídas por E/S, e que as E/S de mensagem no controle da mesa serão mais flexíveis, permitindo uma interação próximas da linguagem natural.

Olhando a saída da divisão de controle das aplicações dos distantes Sistemas de Potência (em sub-estações, em séries simuladoras e planejamento de operações) se tem um conjunto de dados que pode ser operado sobre um campo menos diversificado.

E/S usam uma incompatível estrutura de dados no início.



A figura abaixo dá uma visão do processo dos dados do sistema SCADA.

Os círculos escuros marcam as possíveis associações de pontos para extração de dados em E/S.

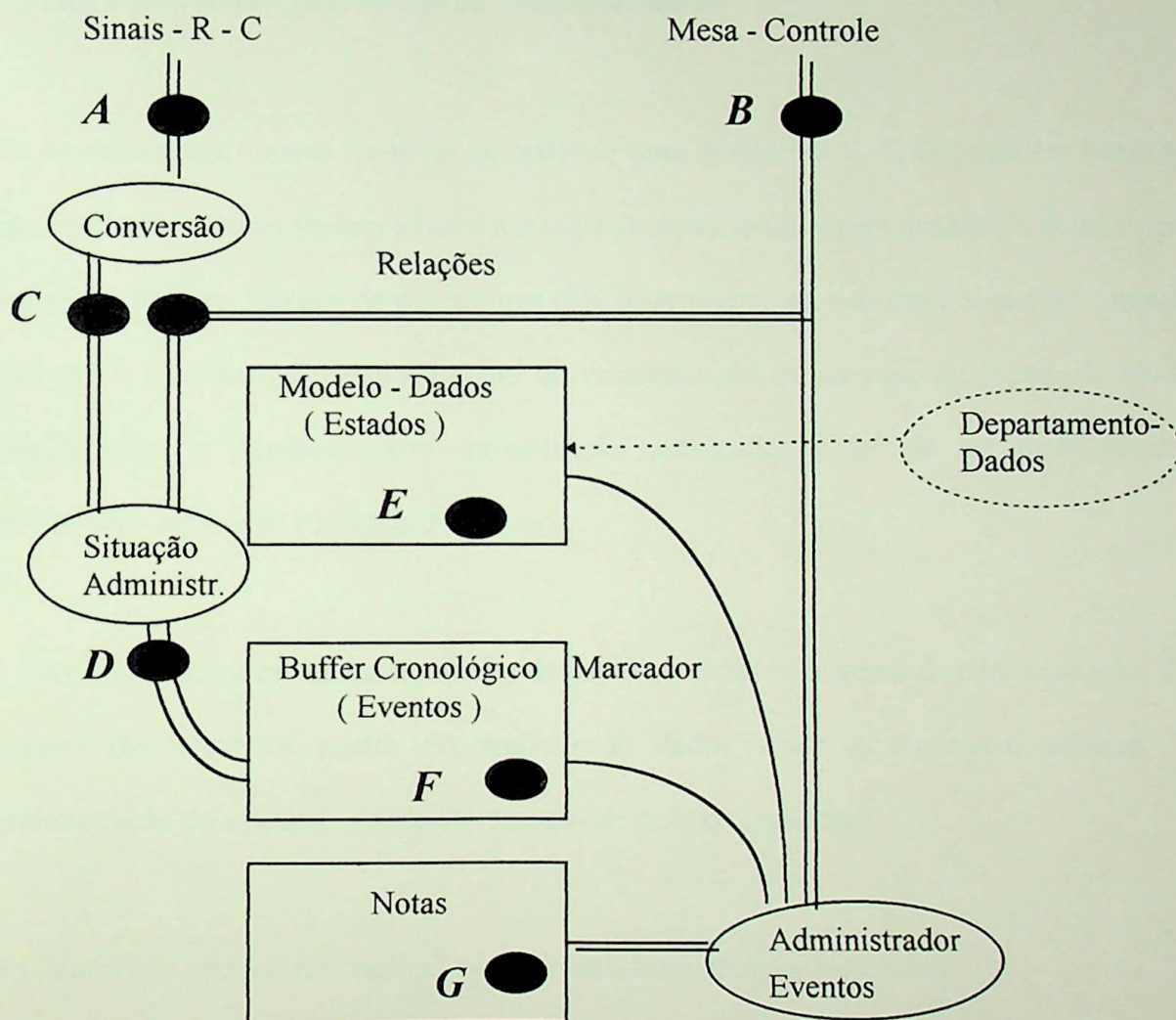


Figura 3.1 : Centro de Controle de Sistema de Dados

As duas principais fontes A e B de informações são as entradas na mesa de controle e os sinais remotamente transmitidos (estados dos equipamentos ou valores medidos).

A : Sinais de controle remoto são códigos compactos, e contêm uma mistura de eventos (estados novos e alterados) e repetidamente estados transmitidos. Primeiramente eles são filtrados e convertidos para código de modelo de dados.

B : As mensagens entram via mesa de controle para armazenar E, F, G e contêm notas ou eventos. Estes eventos podem adicionar conjunto de atributos operacionais ("trabalho pendente") de itens no modelo de dados, mas eles podem também substituir sinais de controle remoto no caso de mal funcionamento de comunicação ou no caso de partes da chave elétrica não ser equipada para comunicação automática e ter de ser manualmente transportada após uma chamada de telefone.

C : As duas extensões das fontes são unidades de posição no sistema de administração. Os eventos são alterados dentro do modelo de dados vindo a ser novos estados. A administração do estado é gerada em eventos de tempo (segundos)

D : dentro dos eventos do "buffer" para futura administração dos eventos.

Tipos de dados e armazenagem de dados simbolizados por blocos centrais :

E : O modelo de dados é usado para administrar o estado de todos os itens do Sistema de Potência. Ele contêm a identificação das chaves elétricas, alarmes, medidores e pontos de

medida, seus estados e outras inter-ligações topológicas. Além disso, o acesso aos dados e instruções operacionais para processamento podem ser incluídas. O modelo de dados é dependente de um serviço-de-dados, seguindo as extensões e alterações dos equipamentos para ser controlada na rede.

F : Eventos do “buffer”, todos os eventos operacionais são introduzidos em um tempo-rotulado na ordem cronológica para outras administrações de eventos.

G : O censura “buffer” transporta informações resultadas de operadores, “shift a shift”. Ele contém uma linguagem natural crítica considerando estados especiais ou eventos. Os itens ou eventos recebem um marcador no modelo de dados ou eventos do “buffer” respectivamente, que é visualizado nos diagramas de textos ou alarmes.

No sistema de controle presente, o modelo de dados pode ser uma coleção de listas dedicadas, uma separada base de dados ou partes de um processo de base de dados que contém descrições de subestações, controle remoto e listas por ligação do teclado, etc.

Usualmente a chave elétrica, alarmes, valores medidos e leitura de medidores são distribuídos em diferentes processamentos “path” da fonte. Valores medidos são diretamente distribuídos dentro do modelo de dados, e produz eventos somente em casos de violação de limite, ou grande inclinação de um medimento.

Em Sistemas de Potência, a linguagem é usada pelo operador na escrita e na fala. Internamente usa números ou códigos alfanuméricos.

A interface do usuário é feita :

- como inscrições em diagramas;
- saídas lógicas;
- quadros-negros e menus;
- na lista do serviço de dados.

Aplicações de E/S são desenvolvidas para :

- verificar a posição dos equipamentos;
- reduzir alarmes e diagnosticar falhas;
- restaurar a rede;
- análise de contingência (análise de falha);
- previsão de carga;
- unidade e empreendimento na planta.

A análise das falhas é mais complexa em redes elétricas pois requer um conhecimento adicional do estado dos equipamentos.

As E/S são para restauração da rede. Com uma interface de um programa, transferem os dados, avaliam as respostas e reingressam uma decisão. Se a base de dados

não oferecer uma resposta, os fragmentos dos textos serão juntados dentro de uma questão para ser respondida pelo operador.

3.4 - PROCESSAMENTO DA LINGUAGEM NATURAL EM PROLOG

O Prolog têm especiais aplicabilidades no processamento da L.N. em três aspectos[7] :

- a linguagem natural pode ser escrita diretamente como programas em Prolog, tanto que a análise e a síntese da L.N. podem ser executadas eficientemente por execução direta do Prolog.
- a representação semântica de textos de L.N. em lógica são a mostra natural do Prolog porque o Prolog é uma conexão fechada à lógica.
- a inferência com estas representações lógicas pode frequentemente ser efetuado por inferência do Prolog, ou por processos de inferência desenvolvidos no início do Prolog.

As aplicações do Prolog no processamento da L.N. será baseado em um tipo de gramática chamada Gramática Lógica Modular (MLG) designada para uma sintaxe de tratamento modular e semânticos.

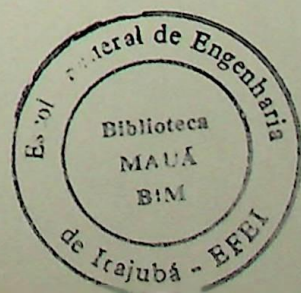
3.4.1 - A Gramática em L.N.

Em geral, a finalidade de uma gramática em L.N. é de associar estruturas de análise de sentenças. Estruturas de Análise podem ser representações semânticas, que dão significado de sentenças e são usadas em inferência. Estruturas de análise também são estruturas sintáticas (ou análise sintática), que representam a divisão de uma sentença dentro das frases, subfrases, subsubfrases, etc. Algumas gramáticas associadas a ambos tipos de estruturas de análise de sentenças, e a gramática podem ser divididas dentro de um componente sintático, que produz análise sintática e um componente de interpretação semântica, que acompanha a análise sintática e produz representação semântica [10].

Uma expressão de uma linguagem possui sintaxe e semântica. A sintaxe da expressão refere-se às regras. A semântica refere-se ao significado intencionado de uma expressão. Os computadores são excelentes na interpretação de sintaxe, mas fracos no que se refere a semântica [6].

Outros tipos de gramáticas podem não produzir análise sintática, mas produzir representações semânticas diretamente. A Gramática Lógica Modular faz uma separação dentro de um componente sintático e um componente de interpretação semântica.

O formalismo usado para representação semântica é uma versão de predicado lógico chamado LFL (Linguagem de Forma Lógica). A LFL visa uma representação geral do significado das sentenças de linguagem natural, incluindo um adequado tratamento de



quantificação e relatadas construções envolvendo campos. Desde então a linguagem de representação semântica é a "meta" da linguagem de um sistema gramatical.

3.5 - PROCESSAMENTO DE LINGUAGEM NATURAL

Comunicação entre pessoas e computadores é uma área em que se começa a se fazer pesquisa. A última meta, é a de capacitar o usuário "falar" com o computador com uma linguagem tal como o Inglês ou Português, e o computador ajudar a falar nesta linguagem. A verdadeira inteligência artificial pode ser capaz disto. Pessoas serão capazes de explicar qual tarefa ele quer executar e o computador responder. Se você conhece uma linguagem de computador você sabe como escrever um programa de computador para fazer exatamente o que você quer. Computadores são tão inteligentes como as pessoas podem fazê-lo; eles não podem raciocinar por eles mesmos, nós temos de dar as instruções específicas para que sejam executadas. Para um computador entender uma linguagem temos de utilizar os elementos básicos da linguagem e entrar com os dados no computador, temos de projetar computadores e programas de computadores que possam acessar estas informações.

A criação deste tipo de sistema de comunicação entre pessoas e computadores é chamado de "processamento de linguagem natural".

Se você têm um robô de inteligência artificial que pode fixar coisas, você pode dar o comando :

1 - desligar a chave com uma luva.

Na frase nós reconhecemos que a chave não estará envolvida em uma luva . Nós podemos reconhecer o erro e corrigi-lo; o robô é que deverá estar utilizando uma luva ao desligar a chave. O robô precisaria saber mais que o significado das palavras e sua relação com outras, ou fará tentativas para achar a chave envolvida em uma luva. As sentenças têm estruturas idênticas, o robô pode não ter um conhecimento da gramática, mas pode ser capaz de associar descrições e objetos e precisa de regras especiais para entender.

Na inteligência artificial, o robô, pode ser capaz de analisar sentenças em relação a outras sentenças. Por exemplo :

1 - João abriu o conector.

2 - Ele consertou o defeito.

O "Ele" na segunda sentença se refere a "João" na primeira sentença. Sem a sentença 1, a sentença 2 pode ter pouco significado. A linguagem humana têm referência a um "contexto sensitivo". A sentença 2 têm sentido se for interpretada após a sentença 1. Quando uma sentença não depende de outra para ser interpretada é chamado de "contexto livre".

O objetivo do aplicativo de Interface Homem-Máquina em Linguagem Natural é o de fornecer mais uma opção ao usuário final de se comunicar com a máquina.

Um ambiente em L. N., baseado em conhecimento, permite que o usuário participe do processo de execução da tarefa-meta, explicitando interativamente sua intenção. Usando sua linguagem de comunicação cotidiana. Esta forma de comunicação adotada é a própria linguagem do usuário, ou seja, a Língua Portuguesa.

O conhecimento resultante das interações com o usuário deverá ser “aprendido” pelo ambiente. Essa concepção tornará o ambiente capaz de mapear um conhecimento dentro do domínio.

Os parâmetros são fornecidos pelo usuário, e o sistema os exigirá de forma “amigável”, explicitando a necessidade dos mesmos, baseado no conhecimento armazenado.

O ambiente poderá ser usado em diversas aplicações que necessitam o intercâmbio de informações dos parâmetros e dos procedimentos.

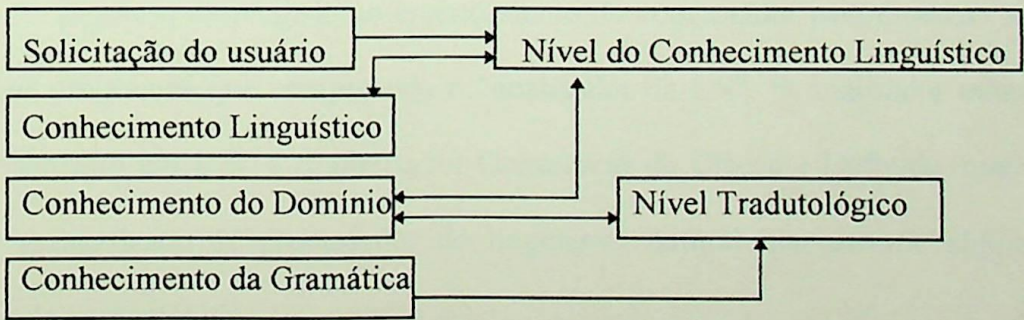


Figura 6.3 : Visão geral do ambiente em Linguagem Natural

Utilizamos como exemplo aplicativo o Sistema ESRASE, que foi desenvolvido como uma ferramenta de auxílio na operação de uma subestação de energia, visando uma maior otimização das funções, objetivando uma maior confiabilidade dos serviços prestados. O sistema é baseado na tecnologia de Sistemas Especialistas, isto é, são um subconjunto dos sistemas baseados no conhecimento. Ele analisa uma ocorrência de algum defeito na subestação, estabelece e executa um Plano Diretor de Ações de Restabelecimento e as variações dos estados dos componentes da subestação que estão sendo monitorados[29].

Procurou-se montar as idéias principais da interface desde a obtenção do texto, o tratamento de cada item do texto, a classificação, a transformação, até a obtenção do texto na linguagem acessível pela máquina.

3.5.1 Análise da Linguagem

O processo empregado no entendimento do computador para entender sentenças é produzir programas que compreenda o "analisador da LN". O analisador mais usado no processamento em L.N. é o analisador Gramatical de Cláusula Definida, que é um tipo muito generalizado de processador de linguagem natural que permite algum nível de análise de contexto[10].

Um processador de L.N. usando uma Gramática de Cláusula Definida divide uma expressão em partes. Inicia com substantivo e verbo, e define nomes, verbos, preposições e outras partes da linguagem. A operação deste analisador é top-down (um exemplo pode ser visto nos próximos itens).

Três funções básicas da análise da linguagem são :

1 - Análise Léxica (analisando palavras)

2 - Análise Sintática (analisando a posição de palavras na sentença, incluindo regras de gramática)

3 - Análise Semântica (analisando o significado das sentenças por si mesmo em relação a outras sentenças).

3.5.1.1 - Análise Léxica

A "Análise Léxica" é a divisão das sentenças em palavras e pontuações ou pausas. As próprias palavras podem ser divididas dentro de raízes, prefixos e sufixos. Por exemplo:

palavra -----> continuamente

raiz -----> continua

sufixo -----> mente

A palavra "independe" consiste de :

prefixo -----> in

padrão -----> depende

Um lexico ou dicionário define palavras, mas conseguir que o computador as entenda no contexto é uma tarefa considerada mais difícil.

3.5.1.2 - Análise Morfológica e Sintática

Dividir uma sentença em partes é um passo vital para o computador entender a linguagem. Pode-se traduzir as regras de gramática e sintaxe dentro de uma forma que o computador possa controlar[10].

O texto digitado pelo usuário é quebrado em palavras (tokens). Gera-se uma lista que será processada pelo programa, utilizando-se da base de dados. Será então feita uma pesquisa no vocabulário, a procura da palavra.

Uma sentença (S) é geralmente composta de uma frase substantivo (NP) e uma frase verbal (VP) que pode ser representada como :

S --> NP, VP

Uma frase substantivo (NP) pode ser quebrada em um determinador (DET) (tal como um artigo, conjunção, pronome, etc.) e um substantivo (N), representado como :

$$NP \rightarrow DET, N$$

Uma frase verbal (VP) pode também ser quebrada em um verbo (V) seguido por outra frase substantivo (NP), como :

$$VP \rightarrow V, NP$$

Uma frase substantivo (NP), entretanto, pode também ser um simples substantivo (N) :

$$NP \rightarrow N$$

Na procura das palavras são utilizados alguns algoritmos de comparação. Assim é possível que uma palavra escrita de maneira incorreta seja aceita, desde que tenha um coeficiente de semelhança grande com a palavra do banco de dados, evitando que o programa faça perguntas ou peça confirmações.

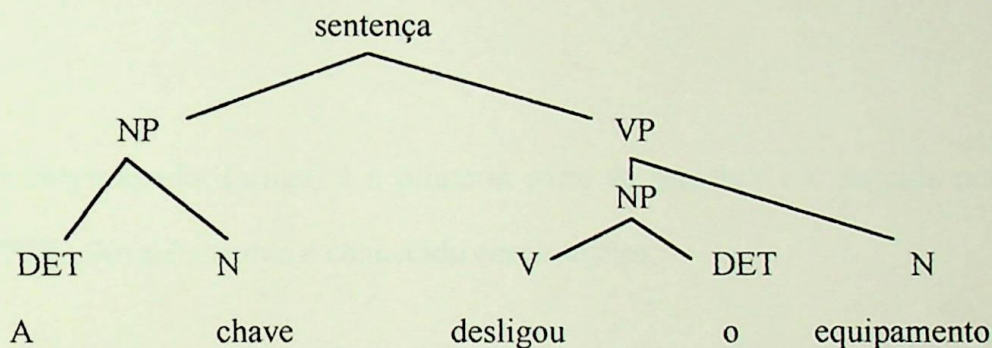
Tendo reconhecido todas as palavras, o programa executa a análise morfológica propriamente dita, associando as palavras a sua função morfológica : adjetivo, pronome, substantivo, conjunção, etc.

De posse da morfologia das palavras é possível deduzir uma estrutura gramatical em que o texto se encaixe. Este processo é feito pela quebra da lista obtida na morfologia em blocos menores, sucessivamente, formando as funções sintáticas.

Um caminho comum de divisão de uma sentença é pela criação de um analisador gramatical em árvore, que é uma representação esquemática da estrutura sintática de uma sentença. A sentença :

A chave desligou o equipamento

pode ter uma estrutura da forma :



As palavras da sentença são identificadas e cada palavra é classificada pelo tipo. A palavra "A" é um determinador (DET), "chave" é um substantivo (N), "desligou" é um verbo (V), "o" é um determinador (DET), e "equipamento" é um substantivo (N).

No projeto de interfaces em Linguagem Natural pode valer à pena implementar a análise morfológica para economizar memória de máquina, pois, assim, não é preciso armazenar todas as palavras derivadas de uma mesma raiz.

3.5.1.3 - Análise Semântica

O computador, após ter classificado as palavras de uma sentença, faz a análise semântica, isto é, tenta produzir um significado para a sentença. Em sistemas de I.A. adapta-se um conjunto de regras para se estabelecer um caminho que o computador possa usar. Por exemplo, na frase usada anteriormente poderíamos ter um conjunto de regras em uma base de conhecimento para interpretar a sentença :

Frase : A chave desligou o equipamento

 Det Subst. Verbo Det Subst.

Regra 1 :

SE um determinador (artigo) é a primeira parte da sentença e é seguida por um substantivo, ENTÃO o substantivo é conhecido como sujeito.

Regra 2 :

SE o verbo sucede um sujeito, ENTÃO o verbo informará o que o sujeito faz.

Regra 3 :

SE o determinador (artigo) seguido por um substantivo sucede o verbo, ENTÃO se têm um objeto (direto).

Regra 4 :

SE uma sentença é formada por substantivo, verbo e objeto direto, ENTÃO o objeto é o complemento verbal do sujeito.

Para uma melhor compreensão analisamos que, na regra 1 identificamos a chave como um sujeito. Na regra 2 dizemos que a chave desligou. Nas regras 3 e 4 identificamos o equipamento como um objeto direto. Aqui pudemos perceber como o processador de L.N. usando uma base de conhecimento de regras sintáticas, semânticas e lexicas processa o "compreensão" da sentença.

Um processador de L.N. pode ser usado em sistema de Inteligência Artificial (I.A.), permitindo que os dados sejam passados verbalmente [11].

Verificar o que	Processador	Sujeito(chave)	
a chave faz...	----> Linguagem	---> Objeto(equipamento)	---> I.A.
	Natural	Verbo(desligou)	

O processamento em L.N. pode livrar o computador das limitações das complexas linguagens de computador.

Provavelmente, os programas possibilitarão ao computador entender as pessoas e "falar" com elas usando o Inglês ou Português. Tem-se aperfeiçoado os trabalhos neste sentido, afim de se obter um computador que seja genuinamente "inteligente"[12].



CAPÍTULO 4

BASE DE CONHECIMENTO

4.1 - INTRODUÇÃO

Um dos objetivos deste trabalho é registrar o evento ocorrido, que está relacionado a nossa base de conhecimento específica voltada ao Sistema ESRASE para que haja uma interpretação em linguagem natural do evento ocorrido. Construímos uma base de conhecimento com termos da língua portuguesa, para que fosse realizada a análise morfológica e sintática da frase. À partir daí ocorrerá a interpretação significativa e a correlação dos eventos que utiliza da base de conhecimento do sistema ESRASE. Esta base de conhecimento poderá ser formada por outros termos que a frase será também interpretada de acordo com a área específica. Utilizamos também de um Modelo Entidade Relacionamento para mostrar de forma esquemática a estrutura da informação.

A base de conhecimento é uma coleção de arquivos de dispositivos externos de armazenamento, que guardam as informações de interesse do Domínio do Problema e das responsabilidades do sistema[13].

Para o desenvolvimento do programa foi necessário um estudo aprofundado da gramática da língua portuguesa.

Colocamos aqui alguns pontos considerados mais importantes para a interpretação e execução do programa, com exemplos extraídos do mesmo.

À seguir, como foi feita a :

4.2 - ANÁLISE MORFOLÓGICA

4.2.1 - Verbo

É uma palavra que exprime ação, estado, fato ou fenômeno. O verbo reveste diferentes formas para indicar a pessoa, o número, o tempo, o modo e a voz. Os tempos situam o fato dentro de determinado momento e divide-se em : presente, passado e futuro. O modo indica as diferentes maneiras de um fato se realizar e divide-se em : indicativo, subjuntivo e imperativo. A voz divide-se em : ativa e passiva[14].

Principais verbos encontrados na gramática SEP :

Verbos auxiliares : Estar

Verbos regulares :	(1ª conjugação) :	Ligar
		Operar
		Desligar
		Quebrar
		Consertar
	(2ª conjugação) :	Restabelecer
	(3ª conjugação) :	Abrir
		Cair

Verbos irregulares : Ser

Tratamento do Verbo no Programa PARSEER :

Na análise morfológica do verbo foi criado uma rotina que analisa o radical (apenas as duas primeiras letras) do verbo encontrado na base com os radicais dos verbos existentes no Banco de Dados; pois, a cada verbo, existiria uma infinidade muito grande de perguntas devido a grande quantidade de possíveis flexões existentes para os verbos. Caso o radical não exista na Base de Dados (teste mencionado acima), o verbo é classificado como irregular e é testado apenas com os irregulares.

A flexão do verbo está armazenado no B.D. em forma de lista, o objeto é conjugação. A flexão para os três tipos de conjugação foram efetuados. Apesar de haver muitas repetições de flexões do verbo foi mantido a integridade da conjugação repetindo essas flexões devido a facilidade de visualização na consulta do B.D..

Exemplo da rotina[31] :

...

is_verb(X) :-

verif_rad (X, Y),

conjugação ("R", TOKL),

lista (X, Y, TOKL);

conjugação("I", TOKL),

verb_irr (X, TOKL).

verif_rad (X, Y) :- frontstr (2, X, STR1, STR2),

verb (Y),

frontstr (2, Y, VSTR1, VSTR2),

STR1=VSTR1.

lista (X, Y, [TOK/TOKL]) :- concat (Y, TOK, X).

lista (X, Y, [_/TOKL]) :- lista (X, Y, TOKL).

verb_irr (X, [TOK/TOKL]) :- X=TOK.

verb_irr (X, [_/TOKL]) :- verb_irr (X, TOKL).

...

Exemplo de listas do Banco de Dados :

- Presente (indicativo) : [o,as,a,amos,ais,am]

- Pretérito perfeito : [ava,avas,ava,ávamos,áveis,avam]

- Presente (subjuntivo) : [e,es,e,emos,eis,em]

4.2.2 - Preposição

É uma palavra invariável que liga um termo dependente (objetos indiretos, complementos nominais, adjuntos, etc) a um termo principal, estabelecendo uma relação entre ambos. Isoladamente as preposições são palavras vazias de sentido[14].

Dividem-se em essenciais (as que sempre foram preposições) e acidentais (funcionam como preposição). Nos interessa apenas as preposições essenciais : a, ante, até, após, com, contra, de, em, entre, para, por, sem, sob, sobre.

Por exemplo :

“O disjuntor de transferência 1K5 está fechado.”

“A chave de manobra inicial está aberta.”

“Mudar a chave 1K5 de aberto para fechado.”

No programa,

prep(“de”).

prep(“em”).

prep(“para”).

prep(“por”).

...

is_prep(X) :- prep(X).

temos : Como a preposição é uma palavra invariável seu teste é bem simples, não exigindo consistência nenhuma.

4.2.3 - Conjunção

É uma palavra invariável que liga orações ou termos da oração[14].

Divide-se em : coordenativas e subordinativas (ligam duas orações subordinando umas as outras). Nós só nos preocuparemos com as coordenativas, como : e, mas, ou, logo, portanto, que, porque, pois, nem.

Por exemplo :

“Ligar a chave 1K7 e 1K2.”

“Desligar a chave que liga a chave de manobra.”

No programa :

...

conj(“e”).

conj(“mas”).

conj(“ou”).

conj(“que”).

...

is_conj(X) :- conj(X).

temos: Como a conjunção é também uma palavra invariável, seu teste é bem simples não exigindo consistência nenhuma.

4.2.4 - Artigo

É uma palavra que se antepõe aos substantivos para determiná-los. Indica-lhes, ao mesmo tempo, o gênero e o número. Isoladamente, os artigos são palavras de todo vazias de significado. Dividem-se em definidos e indefinidos[14].

Os definidos (o,a,os,as) determinam os substantivos de modo preciso.

Exemplo :

“O disjuntor 1K4 está fechado.” (um disjuntor conhecido, determinado)

Os indefinidos (um, uma, uns, umas) determinam os substantivos de modo vago, impreciso, geral.

Exemplo :

“Uma chave de transferência quebrou.” (uma chave indeterminada)

No programa,

...

artigo(“o”).

artigo(“a”).

artigo(“um”).

artigo(“uma”).

...

is_artigo(X) :- artigo(X), testa_gen(X).

is_artigo(X) :- artigo(Y), concat(Y,"s",X), testa_gen(X).

is_artigo(_) :- gen="",nro="".

testa_gen(X) :- tipo=[a,as,uma,umas],

tipo=X, gen="F", testa_pl(X);

gen="M", testa_pl(X).

testa_pl(X) :-

STR_len(X,tem),tam1=tam-1,frontstr(tam1,X,_,STR2),STR2="S",nro="P",nro="1".

Obs: adiciona-se um flag indicando o gênero e o número do artigo. Caso não seja artigo o flag terá branco.

4.2.5 - Substantivo

Os substantivos dividem-se em substantivos próprios (aplicam-se a um ser particular), comuns (referem-se a seres ou coisas de mesma espécie) ou abstratos (designam qualidades, sentimentos, ações e estados dos seres dos quais se podem separar e sem os quais não poderiam existir)[14].

Banco de substantivos da gramática SEP :

Relé; Relé geral; Relé 1K50, ...; Disjuntor; Disjuntor de Transferência;

Disjuntor 1K4, ...; Chave; Chave Seletora; Chave de Manobra Inicial;

Chave 1K3, ...; Circuito; Equipamento; Cenário; outros...

A seguir, algumas implementações realizadas para tentar formar uma gramática legível :

1ª) Podemos descrever os substantivos da gramática simplesmente como vemos, enumerando-os um a um. Como exemplo, temos :

substantivo(“chave”).

substantivo(“disjuntor”).

substantivo(“chaves”).

Mas, deste modo, não teremos informações sobre o gênero e/ou sobre o número do substantivo.

2ª) Podemos utilizar a informação do gênero pela simples posição do substantivo no fato, temos :

substantivo(“ “, “chave”).

substantivo(“disjuntor”, “ “).

Onde a primeira posição do fato substantivo designa o masculino, e a segunda o feminino.

Quando o substantivo não apresenta um dos gêneros, basta apenas colocar um flag de não existente.

Deste modo o Banco de Dados é organizado, economizando códigos e também se conseguiria testar o artigo ou o verbo da frase, para saber se foi corretamente empregado.

Para se testar o número do substantivo, como são na maioria regulares, utiliza-se regras no programa.

Um exemplo, onde a rotina testa para todos os substantivos acrescentando a eles “s”, e “es” [31]:

`is_noun(X) :- noun(X).`

`is_noun(X) :- noun(Y), concat(Y,"s",X).`

`is_noun(X) :- noun(Y), concat(Y,"es",X).`

Implementação da rotina :

- Substantivos terminados em vogais, acrescenta-se “s” (excessões : terminados em “ão” flexionam-se em “ões” ou “ãos”);
- Substantivos terminados em “r”, “z” ou “s”, acrescenta-se “es”;
- Substantivos terminados em “al”, “el”, “il”, “ol”, “ul”, flexionam-se o “l” final por “is”;
- Substantivos terminados em “m”, troca-se por “ns”.

Utilizando no programa :

- 1) Quando você já têm o artigo, portanto já sabe o gênero do substantivo, você vai direto na posição respectiva do gênero do substantivo.



Por exemplo :

“O disjuntor está fechado.”

À partir da análise do artigo “o” você pode ir direto a procura da primeira posição do fato substantivo, pois já sabe que ele deverá ser também masculino.

2) Na ausência do artigo, há necessidade de se testar os dois gêneros.

3) Uma terceira forma encontrada para tratar o substantivo é, além de referenciar o gênero no fato, referenciar também o plural dentro do mesmo fato. Por exemplo :

substantivo(relé, relés, “ “, “ “).

substantivo(“ “, “ “, chave, chaves).

Portanto não existiria rotinas de tratamento do plural no programa. Mas em compensação o B.D. teria um considerável aumento, sabendo que a maioria dos plurais dos substantivos são de forma regular, ou seja, bastando acrescentar “s” ou “es” teríamos um desperdício de memória por excesso de informações no B.D.

OBS : VÍRGULA : o programa deverá ter também o teste da vírgula, utilizado para separar palavras ou orações justapostas .

Por exemplo : “A chave, o disjuntor e o relé são elementos ...”

No programa :

...

comma(“,”).

...

is_comma(X) :- comma(X).

Feita a Análise Morfológica de cada termo da oração em separado (transformando toda a frase em uma lista morfológicamente correta), resta agora a Análise Sintática de cada termo da frase.

4.3 - ANÁLISE SINTÁTICA

Examina a estrutura de uma frase, dividindo-a em orações e reconhece a função sintática dos termos de cada oração.

São dois os termos essenciais da oração : o sujeito (é o ser de quem se diz alguma coisa) e o predicado (é aquilo que se afirma do sujeito).

4.3.1 - Sujeito

É constituído por um substantivo ou pronome.. Em torno do núcleo podem aparecer palavras secundárias (artigos, adjetivos, locuções adjetivas, etc.)[14].

Por exemplo :

O disjuntor está fechado.

sujeito

onde : “o” é artigo

“disjuntor” é substantivo (núcleo do sujeito)

O disjuntor de transferência está fechado.

sujeito

onde : “o” é artigo

“disjuntor” é substantivo (núcleo do sujeito)

“de transferência” é locução adjetiva.

O relé 1K21 abre o disjuntor 1K4.

sujeito

Tipos de frases encontradas na gramática SEP :

- 1 - O disjuntor 1K5 está fechado.
- 2 - Os disjuntores 1K7 e 1K8 estão fechados.
- 3 - O disjuntor 1K7 e a chave 2K9 estão fechados.
- 4 - A chave 1K5 pode ser aberta.
- 5 - Uma chave desligou.
- 6 - Uma chave seletora está aberta erradamente.
- 7 - O disjuntor de transferência 1K5 está transferido.
- 8 - O relé geral 1K5 abre o disjuntor 1K4.

Para nós o substantivo Disjuntor e a locução adjetiva de transferência será interpretado como um único substantivo : “disjuntor de transferência”. Também segue esse mesmo sistema, as palavras :

chave geral

relé geral

chave seletora

chave de manobra inicial

disjuntor de transferência

Formas imagináveis da frase (PARSER) [1]:

- Frase 1 :

frase(sujeito(artigo(“o”), subst(“disjuntor”), loc_adj(prep(“de”), adjetivo(“transferência”))), predicado(verbo(“está”), loc_adj(none, adj(“quebrado”))).

- Frase 2 :

frase(s_nominal(artigo(“o”), subst(“relé”), loc_adj(none, adj(“geral”))), s_verbal(verbo(“abrir”), presente), s_nominal(artigo(“o”), subst(“disjuntor”), loc_adj(none, adjetivo(“1K4”)))).

4.4 - O PROLOG E O SISTEMA DE BASE DE DADOS

4.4.1 - Introdução

Prolog é uma linguagem de computador baseada em lógica que permite ao programador definir regras e fatos (as regras são equivalente a “procedures” ou subrotinas em uma linguagem convencional, e os fatos são equivalentes a itens de dados), e perguntas, mais comumente chamadas “Queries” [16].

A base de dados é justamente um conjunto organizado de arquivos que nos permite armazenar informações sobre alguma coisa que nós conhecemos. Uma base de dados relacional, por exemplo, armazena as informações em uma tabela de duas dimensões (linhas e colunas) onde na primeira linha temos o nome da tabela, na segunda linha o nome dos campos em colunas, e, logo abaixo, os registros[15].

4.4.2 - Base de Dados

O conjunto das palavras e expressões que servem para resolver um problema dentro de um sistema específico é denominado banco de fatos. Este se associa a um conjunto maior onde se engloba também o conhecimento adquirido de união (interna) entre esse conjunto, criado através de regras.

Para se evitar uma grande massa de dados na base de dados optou-se pela inclusão apenas dos termos essenciais da oração :

- artigo
- adjetivo
- substantivo
- verbo

Quanto a classificação do gênero das palavras, não foi tomado nenhum cuidado especial devido ao sistema ser específico a uma área de energia. Neste caso, a grande gama de palavras tem um único gênero, portanto, obteve-se um banco de dados mais compacto.

Uma atenção especial foi dada aos verbos, que são em número muito grande de flexões e de formas. Neste caso, houve uma divisão em dois fatos :

- verbo no infinitivo
- conjugações.

O fato "verbo no infinitivo" contém a informação do radical do verbo e de sua conjugação.

O fato "conjugações" contém a informação do tipo do verbo (regular ou irregular), da conjugação e uma lista com o sufixo de todos os modos e tempos verbais possíveis.

Exemplo :

verb ("atu", "ar").

conjugação ("r", "ar", ["o", "as", "a", "amos", "ais", "aram"])

A interligação da palavra em língua portuguesa na linguagem do sistema foi feita através da associação com a palavra identificada para o ESRASE..

Exemplo :

associa("relé", "proteção")

Assim, o Banco de Dados é estruturado à partir de fatos e regras que devem retratar todo o vocabulário do usuário, bem como a conexão com o vocabulário do ESRASE..

4.4.3 - A Base de Dados Relacional

Fundamentalmente, a idéia de armazenamento de dados é uma noção de registro. Um registro é um conjunto de itens de dados, ou em outras palavras, um registro é um conjunto de itens de dados que são conectados em um mesmo caminho. Um registro consiste de um número de campos nomeados[17].

O Prolog, quando unido a uma base de dados relacional, provê um sistema de base de dados que pode ser usado para uma variedade de aplicações. Ele pode ser usado para o

desenvolvimento de um Sistema Especialista e para construção de ferramentas de controle de base de dados baseada em regras. Ele pode ser visto como uma extensão natural de uma linguagem de consulta[19].

As funções mais comuns de uma base de dados são criação, acesso e inserção de dados.

Em adição a estas funções, o sistema de base de dados salienta os problemas de[20]:

- integridade : válido somente quando é permitido inserir dados;
- divisão : os dados podem ser acessados por mais de um programa ao mesmo tempo;
- concorrência : os dados podem ser inseridos, ao mesmo tempo, por mais de um programa
- independência dos dados : os programadores não precisam conhecer as dificuldades de processamento de arquivos; alterações no armazenamento físico dos dados não causam alterações em programas que acessam os dados.
- as facilidades com que as “queries” são produzidas : é proporcionado um alto nível interativo da linguagem de consulta.
- segurança : todas as transações da base de dados são ligadas, assim que a base de dados for restaurada, após ter sido destruída.
- minimização de redundância : mínimo de duplicação dos dados;
- eficiência : facilidades de criação de índices.

4.4.4 - O Modelo Relacional

Uma base de dados relacional é um conjunto logicamente relatado numa tabela de duas dimensões. Uma tabela é equivalente a um conjunto de fatos todos com o mesmo número de componentes. Todas as suas colunas ou campos possuem um rótulo ou, como são conhecidos, nome da coluna[21]. Exemplo de uma base de dados relacional :

Relação de Entradas Analógicas

NOME	DESCRIÇÃO	ESCALA	LIMITES
1K-V	Tensão no vão 1k	180 0	152-124
1K-I-Vm	Corrente fase Vm vão 1K	RTC 0	-
1K-I-Az	Corrente fase Az vão 1K	RTC 0	-
1K-I-Br	Corrente fase Br vão 1K	RTC 0	-

Na representação binária da tabela, a função chave pode ser o nome do relacionamento que ele identifica, como por exemplo [14] :

```
LIMITES(1K-V, 152-124)
DESCRIÇÃO(1K-I-Br, Corrente fase Br vão 1K)
ESCALA(1K-I-Az, RTC 0)
```

A representação binária de uma tabela é mais simples de ler. O nome das colunas, que são necessárias para o entendimento da tabela, são representados por simbolos declarados binários na representação binária.

O mais importante do ponto de vista computacional, é que a representação binária pode expressar regras gerais[20]. Por exemplo :

LIMITES(X, -) <-- PERTENCE(X, R.E.A)

ESCALA(X, Y) <-- PERTENCE(X, R.E.A),

NOME(X, Y)

Em geral uma base de dados terá campos de tipos alfanumérico, real e inteiro. É necessário definir a tabela que estamos fazendo uso, dando nome as colunas e o tipo e tamanho de cada campo[21]. Estas informações são usualmente armazenadas em uma tabela que é conhecida como um dicionário de dados, como no exemplo a seguir :

DICIONÁRIO

NOME da TABELA	NÚMERO da COLUNA	NOME do CAMPO	TIPO do CAMPO	TAMANHO do CAMPO
R.E.A.	1	Nome	C	10
R.E.A.	2	Descrição	C	25
R.E.A.	3	Escala	C	8
R.E.A.	4	Limites	I	8

OBS : R.E.A. - Relação de Entradas Analógicas

Em uma base de dados relacional uma coluna particular é acessada pelo nome. No Prolog, não existem nomes associados com campos particulares mas a posição é significativa.

Na ordem de minimizar o tempo de acesso é possível definir uma "chave de campo". Estes campos terão um índice criado, permitindo acesso direto de recuperação do registro todo, sempre que o valor de uma chave de campo é conhecida.

4.5 - O MODELO CONCEITUAL DE ENTIDADES E RELACIONAMENTOS

4.5.1 - Introdução

O Modelo Entidade e Relacionamento (M.E.R.), foi introduzido por Chen[13]. O modelo serve para a representação das estruturas de informações, não contendo uma linguagem para a representação da manipulação.

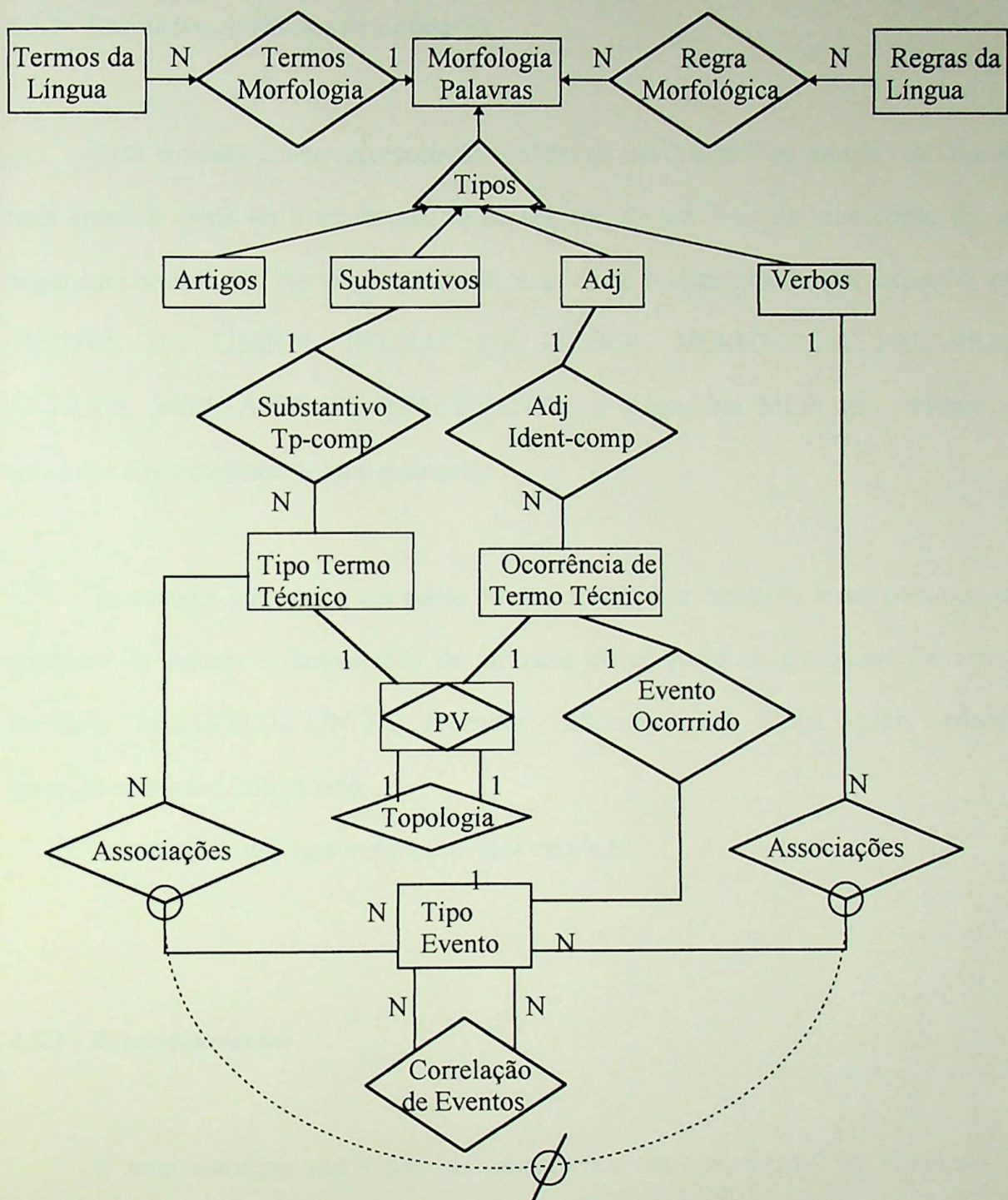


Figura 4.1 : Estrutura do conhecimento de um sistema genérico.

Maiores detalhes sobre o quadro especificado podem ser encontrados no Anexo 1.

4.5.2 - Entidades, Atributos de Entidades

Uma entidade é uma representação abstrata de um “objeto” do mundo real. Assim, uma entidade pode ser a representação de um ser, de um fato, de uma coisa, de um organismo social, etc. No diagrama usado, a entidade é representada, por exemplo, por TERMOS DA LÍNGUA, REGRAS DA LÍNGUA, MORFOLOGIA PALAVRAS, ARTIGOS, SUBSTANTIVOS, TIPO EVENTO, e outros. No MER um conjunto de entidades é representado por um retângulo.

As funções que levam um ponto de um conjunto de entidades a um ponto de um conjunto de valores é denominado de atributos do conjunto de entidades. Exemplo : Entidade - TERMOS DA LÍNGUA, Atributos - Substantivo, Adj, Artigo, Verbo. Apenas o nome do atributo é interessante.

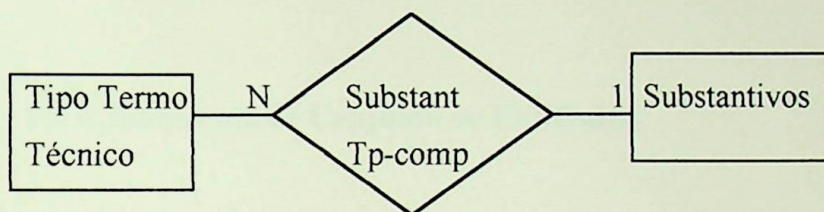
O atributo é uma característica de uma entidade.

4.5.3 - Relacionamentos

É uma estrutura que indica as associações entre elementos de conjuntos de entidades. No exemplo, podemos citar como relacionamentos : TERMOS MORFOLOGIA, EVENTO OCORRIDO, TOPOLOGIA, ASSOCIAÇÕES, e outros. A representação diagramática desse conjunto é o losango. É portanto o nome que contém o significado dos pares no mundo real, isto é, a semântica.

Os símbolos “1” e “N”, determinam a classe do relacionamento. Diz-se : relacionamento de classe “N” para “1”, ou “muitos-para-um”, representando-se esse fato pelo símbolo “N:1”. Podem ser : “1:1” (um-para-um), “N:1”, “N:N” (muitos-para-muitos) OU “1:N”.

Do diagrama :



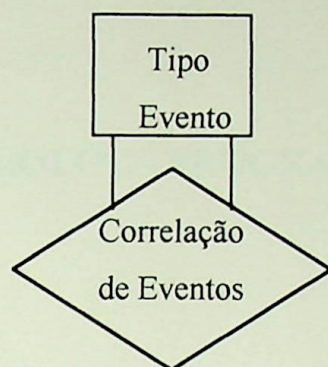
“N : 1” : Cada Substantivo é Substantivo de um número qualquer de Termo Técnico(N) e cada Tipo Termo Técnico é Tipo-componente de no máximo um Substantivo.

Em um relacionamento “N:N”, qualquer elemento de um dos dois conjuntos de entidades pode ocorrer em um número qualquer de pares do relacionamento.

4.5.4 - Auto-Relacionamento

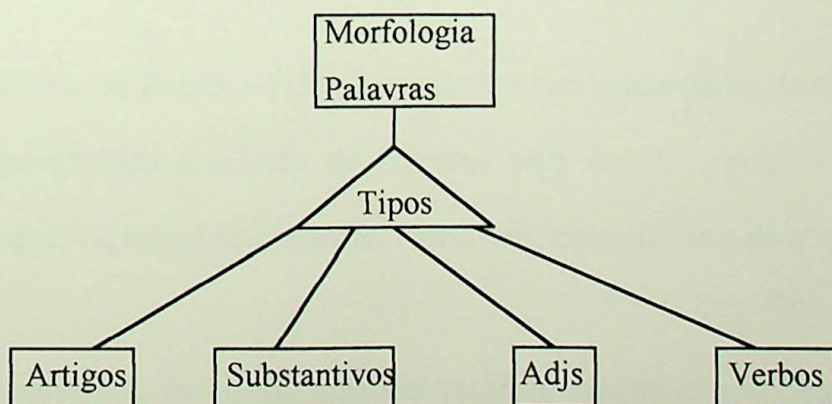
Se R é um relacionamento que relaciona elementos de um conjunto de entidades E a elementos desse mesmo conjunto E, R é denominado de um auto-relacionamento.

Exemplo :



4.5.5 - Particionamento de Conjunto de Entidades

Existem muitos tipos práticos em que um conjunto de entidades representa elementos do mundo real que se subdividem em categorias com atributos parcialmente distintos. Exemplo, do diagrama MER, onde a Morfologia de Palavras é classificada por tipos como Artigos, Substantivos, Adjs e Verbos :



O particionamento tem nome e pode ser representado por um triângulo.

CAPÍTULO 5

PROLOG - PROGRAMAÇÃO EM LÓGICA

5.1 - INTRODUÇÃO

Prolog é uma linguagem interativa que permite resolver problemas que envolvem representação simbólica de objetos e seus relacionamentos.

Programação em lógica significa utilizar um sistema formal para resolução de problemas [16].

Fatos que contribuíram com o desenvolvimento da Programação em Lógica :

- o trabalho de Robinson[15] e outros sobre prova automática de teoremas, datado de 1970, que possibilitou a criação de sistemas para decidir parcialmente se uma sentença é consequência lógica de outras, de forma bem mais eficiente do que se supunha;
- Kowalski[18], introduziu uma interpretação procedimental para conjuntos de cláusulas, que facilita o desenvolvimento (e entendimento) de um programa lógico em cláusulas;

- A Linguagem Prolog foi projetada e implementada por Colmerauer e seu Grupo de Inteligência Artificial, em Marseille (1970-1972), com a escrita do primeiro interpretador Prolog (1973).
- Uma versão mais eficiente deste interpretador foi implementada em 1975, em FORTRAN.
- A partir de 1979, o Prolog, passou a atrair um amplo interesse a nível mundial com a implementação de uma versão mais eficiente que incluía o primeiro compilador Prolog escrito em Prolog, conhecido como DEC-10 PROLOG.
- a introdução da Linguagem PROLOG e o projeto japonês de computador de quinta geração, década de 80, popularizou e motivou o uso de técnicas de programação em lógica;
- se beneficiou das aplicações baseadas na representação e processamento de conhecimento, sistemas especialistas.

5.1.1 - Áreas de Aplicação

- Engenharia de software
- Banco de Dados relacionais
- manipulação simbólica de fórmulas matemáticas

- construção de compiladores
- projetos apoiados por computador
- Inteligência Artificial :
 - . engenharia do conhecimento
 - . prova automática de teoremas
 - . sistemas especialistas
 - . processamento de linguagem natural
 - . robótica.

5.1.2 - Principais Características

- orientada para processamento simbólico;
- permite recuperação dedutiva de informação;
- representa uma implementação da lógica como linguagem computacional;
- permite a obtenção de respostas alternativas;
- suporta definições recursivas para descrição de processos e problemas;
- permite descrever a especificação e a codificação de programas em uma mesma linguagem;
- representa programas e dados através de cláusulas.

5.2 - O QUE É PROGRAMAÇÃO EM LÓGICA

Programa em lógica é um conjunto finito de sentenças onde descrevemos um certo problema ou uma determinada situação, portanto, não é uma descrição de um algoritmo ou procedimento para resolução de um problema, como programas em Pascal[17].

Programação em Lógica enfatiza a estrutura lógica de um problema, não a construção de procedimentos usados para resolvê-lo.

Um programa Prolog consiste de uma série de fatos e regras usada dentro da base de dados interna do Prolog. A base de dados interna é parte do sistema Prolog. Embora seja chamada base de dados, ela armazena regras e fatos e não deveria ser confundida com qualquer sistema de base de dados externo. As regras definem relacionamentos entre fatos e regras dele mesmo, e estes relacionamentos são usados para se responder as “queries”. Devido ao usuário somente especificar os itens de dados e os relacionamentos entre regras e fatos, o Prolog é chamado de não-procedural, pois o usuário não especifica o algoritmo para resolver o problema mas sim os relacionamentos.

Não devemos utilizar a Programação em Lógica somente para testar as contradições mas, também, para incluir respostas extraídas do programa e explicações correspondentes as correções das respostas. Paralelamente, devemos também nos preocupar com a construção das respostas e explicações à partir das refutações.

A Programação em Lógica usa um formalismo computacional, exemplificando um estilo formal de programação identificado como Programação Declarativa.

A definição de um programa em lógica é a descrição axiomática de um certo problema. A criação de programas em lógica envolve a definição de uma linguagem de primeira ordem com símbolos não-lógicos correspondentes a importantes conceitos do problema e, a definição de um grupo de sentenças que prende a estrutura lógica do problema.

A Programação em Lógica é semelhante ao processo de especificação de software, já o programador define as mais importantes propriedades do problema, ao invés de descrever sua solução [16].

A Programação em Lógica ainda considera a especificação axiomática de um problema como um formalismo executável, que pode ser considerado como um programa normal.

A Programação em Lógica não está limitada ao teste de que uma sentença é consequência lógica de outras. A noção de resposta a uma chamada é ampliada para incluir informação efetivamente extraída do programa em lógica e a noção de estratégia de dedução é expandida para incluir técnicas para construção de respostas a partir de deduções[14].

Um programa em lógica consiste de um conjunto finito de cláusulas, frequentemente pertencentes a classes especiais, já os procedimentos de refutação para cláusulas fazem o papel de máquinas para execução de programas em lógica.

A notação em cláusulas é importante pois facilita a construção de sistemas para Programação em Lógica.

5.3 - SINTAXE DA LINGUAGEM PROLOG

Símbolos do alfabeto básico do Prolog :

- pontuação : (,) , . , ' ,
- conectivos : & (para conjunção)
 <-- (para implicação)
- todas as letras do alfabeto de A a Z (maiúsculas e minúsculas)
- dígitos de 0 a 9
- especiais : * , + , ...

5.3.1 - Identificadores e Variáveis

- Identificador simples : cadeia de letras e dígitos, iniciando em minúsculo.

Ex. : disjuntor, relé.

- Identificador especial : cadeia de letras, letras especiais (espaço) e dígitos delimitada por apóstrofo.

Ex : '1K7', '2K50-N'.

- Identificador numérico : cadeia de caracteres numéricos representando números.

- Constante : identificador simples, especial ou numérico.

- Variáveis : cadeia de letra e dígitos iniciadas em maiúsculo.

Ex. : ARTIGO, SUBSTANTIVO.

5.4 . FATOS E REGRAS

Um sistema Prolog consiste de fatos e regras, e execução de “queries”. As respostas as “queries” dão valores as variáveis a procura de todas as soluções. Quando é feito uma query o Prolog responde "yes" (pode ser um sucesso ou uma afirmação do que foi consultado) ou "no". Na resposta à query o Prolog sempre pega o caminho mais fácil, sendo que a ordem dos fatos na base de dados é significativa. Se por um caminho percorrido ocorrer falha, ele tentará outro caminho para se chegar a uma solução, satisfazendo as submetas. Se não houver uma ou mais soluções a resposta final será "no"[5].

A resposta de uma consulta a um programa em lógica não se limita apenas a indicar que uma suposição acerca da informação contida no programa é falsa ou verdadeira. A resposta pode efetivamente exibir informação extraída do programa e vir acompanhada de uma explicação, expressa em termos da refutação que a gerou.

Estas são características próprias da Programação em Lógica.

As noções de consulta e resposta são uma das principais características de Programação em Lógica, não estando presentes no desenvolvimento clássico da lógica.

A inexistência de uma resposta correta deve ser entendida como uma consulta não ser consequência lógica de um programa em lógica, e o fato da substituição vazia ser uma resposta correta deve ser entendido como sendo uma consequência lógica do programa em lógica.

Um programa em lógica deve ser visto de forma semelhante a um programa ou um banco de dados no sentido tradicional, exceto que a execução do programa em lógica é um processo de dedução [16].

Para um programa em lógica definindo um algoritmo, uma chamada para o algoritmo será formulada como uma consulta. Note porém que não há distinção entre parâmetros de entrada e de saída, ou seja, um programa em lógica pode definir algoritmicamente tanto uma função quanto a sua inversa. Para um programa em lógica

definindo um banco de dados dedutivo, uma consulta têm um sentido mais próximo do usual, exceto que, novamente, o processo de recuperação de informação envolve agora dedução.

5.4.1 - Fatos

Composto por um Predicado e por Argumentos. O Predicado define a relação existente entre os argumentos; e o Argumento são os objetos que devem ser relacionados. Um fato é aquilo que já é conhecido.

Forma geral :

Pred (arg1, arg2, ...) .

(.) período

Exemplo de fato :

subst("RELÉ DE TEMPERATURA DO ÓLEO").

adj("1FRD01").

verb("ATU", "AR").

5.4.2 - Regras

É uma propriedade ou uma relação conhecida e considerada verdadeira quando um conjunto de outras relações é verificado. É aquilo que pode ser inferido de fatos conhecidos.

Exemplo de regra :

sujeito(SUJEITO) : - artigo(ARTIGO),	Existirá um sujeito se existir um
subst(SUBSTANTIVO),	artigo seguido de um substantivo
adj(ADJETIVO).	e um adjetivo.

5.4.3 - Linguagens Procedurais "Versus" Linguagens Declarativas

Uma vez que Prolog é uma linguagem declarativa, ela exige que você declare regras e fatos a respeito de determinados símbolos, para depois perguntar-lhe se uma determinada meta segue de uma forma lógica destas regras e fatos. A máquina dedutiva que faz esta verificação é parte do próprio Prolog. Muitas das características únicas do Turbo Prolog envolvem interações de procedimentos com o núcleo declarativo do Prolog. Permitem que você, por exemplo, manipule logicamente características como gráficos e sons que não são tipicamente parte do Prolog ou programação lógica.

A distinção entre linguagens procedurais e linguagens declarativas é uma das razões pela qual uma implementação de linguagem como Turbo Prolog é uma ferramenta tão boa para aplicações de IA, especialmente quando comparada com outras linguagens.

5.5 - PROGRAMAÇÃO EM CLÁUSULAS

As cláusulas em Prolog são formadas por fatos e regras, que são as frases que compõem a linguagem[19].

Um programa em lógica consiste de um conjunto finito de fórmulas. Porém, a maioria dos sistemas para Programação em Lógica considera que um programa é um conjunto finito de cláusulas, que representam fórmulas em uma notação simplificada.

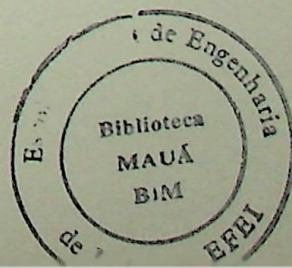
A notação para cláusulas mais frequente em Programação em Lógica considera que uma cláusula é uma expressão da forma :

$$A_1, \dots, A_m \leftarrow B_1, \dots, B_n$$

onde $A_1, \dots, A_m, B_1, \dots, B_n$ são fórmulas atômicas.

Se $x_1, \dots, x_k$ são as variáveis ocorrendo na cláusula, então ela deve ser interpretada

como : para todo $x_1, \dots, x_k$, A_1 ou ... ou A_m valem se B_1 e ... B_n valerem.



Se $n=0$, então a interpretação se resume a : para todo $x_1, \dots, x_k$, A_1 ou ... ou A_m valem;

Se $m=0$, a interpretação será : para nenhum $x_1, \dots, x_k$, B_1 e ... e B_n valem;

e se $m=n=0$, a cláusula será sempre falsa.

Programação em Cláusulas admite apenas programas expressos por conjuntos de cláusulas e consultas representadas apenas por uma cláusula. Quando comparada com Programação em Lógica, a única diferença substancial está em que consultas em cláusulas podem expressar apenas consultas conjuntivas genéricas. Esta restrição torna mais simples o significado de uma consulta expressa por uma cláusula. Os sistemas para Programação em Lógica são baseados em procedimentos de refutação para cláusulas.

Exemplo de cláusula :

`is_subst(X) :- subst(X), !.` Uma palavra X é um substantivo se for declarada na base de dados como um substantivo.

Onde X pode ser substituído por : "relé", "relé de tempo", "chave", "proteção",...

CAPÍTULO 6

SISTEMAS ESPECIALISTAS

6.1 - INTRODUÇÃO

Sistemas Especialistas (S.E.) são programas que imitam algumas habilidades humanas. Contêm informações (isto é, um banco de dados), e uma ferramenta para a compreensão de informação e busca de respostas corretas as questões feitas ao banco de dados (isto é, uma máquina de dedução). Aplicações típicas para sistemas especialistas podem ser encontradas na medicina (onde o computador é solicitado para diagnosticar doenças), na geologia (na qual o computador identifica onde pode ser encontrado petróleo), e outras. Isto significa termos um computador que faça coisas inteligentes, tais como compreender a língua (inglês, português), que desempenhem atividades como o ser humano ou planeje e tome decisões por ele[23].

Os sistemas especialistas são um sub-conjunto dos sistemas baseados no conhecimento. Um sistema baseado no conhecimento não pode ser chamado de sistema especialista até que sua base de conhecimento esteja bastante próxima a de um especialista [24].

Sistemas Especialistas são programas para resolver problemas. Estes resolvem problemas difíceis que necessitam especialização. Eles são condicionados a base de conhecimento porque sua execução depende criticamente da utilização de fatos e heurística usado pelo especialista. Representam uma área de pesquisa da Inteligência Artificial (I.A.) que envolve paradigmas, ferramentas, e estratégias de desenvolvimento[25].

Sistemas Especialistas (S.E.) podem se identificar com programas que utilizam técnicas de I.A., distinguindo controle de conhecimento, com uma área de aplicação com conhecimento envolvendo incertezas e utilizando-se de métodos heurísticos. S.E.'s constituem uma das mais importantes sub-áreas da Inteligência Artificial.

S.E.'s possuem, basicamente, a seguinte estrutura :

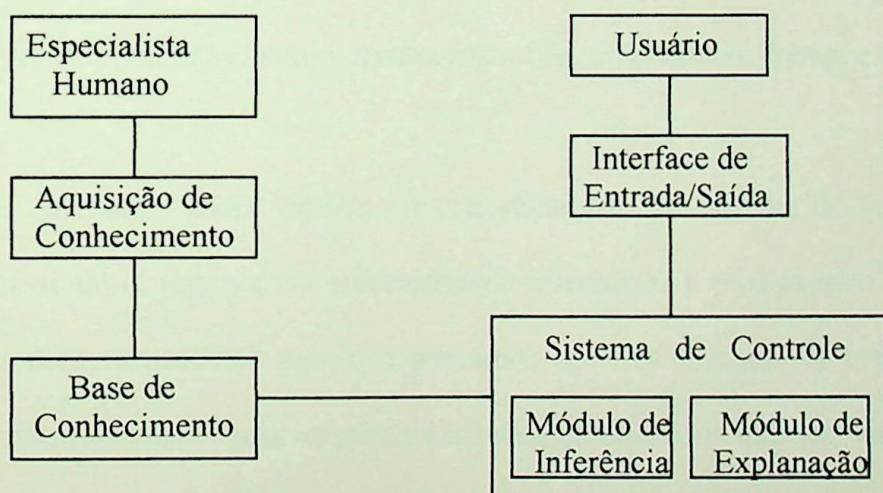


Figura 6.1 : Desempenho especialista depende criticamente de um conhecimento especialista.

6.2 - DEFININDO SISTEMAS ESPECIALISTAS

Sistemas Especialistas diferem em importantes caminhos de sistemas de processamento de dados convencional e sistemas desenvolvidos em outros ramos da I.A.. Em contraste ao sistema de processamento de dados tradicional, aplicações da I.A. geralmente envolve várias distinções de características, que inclui representação simbólica, inferência simbólica, e busca heurística[26]. Sistema Especialista difere de ampla classe de tarefas da I.A. em vários aspectos :

- executa difíceis tarefas em níveis especialistas de atuação;
- enfatiza estratégias de domínios específicos para resolução de problemas ;
- emprega auto-conhecimento para entendimento sobre processos de inferência e fornece explicações ou justificações para conclusões;
- resolve problemas dentro das seguintes categorias : interpretação, diagnóstico, projeto, planejamento, monitoramento, reparo, instrução, prognóstico, debug, e controle.

As metas “goals” podem ser estendidas em um Sistema de Inteligência Artificial através de fatos, regras e um mecanismo de inferência. Temos os domínios “domains” que são as áreas específicas para que possamos projetar sistemas de I.A. Os fatos e regras necessitam alcançar todas as metas em todos os domínios que são virtualmente infinitos. Isto leva um longo tempo, se, antes, podemos registrar um grande conjunto de conhecimento requerido para realizar uma tarefa. Além disso, não existe um computador que possa depositar ou processar alguma coisa abordando muitas informações. O que é

necessário é que se limite estas áreas que contêm muitas informações que podem ser configuradas dentro de um programa de computador.

Os domínios, na prática, contêm um controlável conjunto de informações. Um sistema de I.A. criado para resolver problemas em um domínio particular é chamado "Sistema Especialista". Todo o conhecimento no sistema especialista é fornecido por pessoas que são especialistas neste domínio. Todos os sistemas especialistas são baseados em atualizadas informações técnicas para um dado domínio.

6.2.1 - O Turbo Prolog

Turbo Prolog tem uma estruturação interna para a criação de banco de dados e tem uma máquina dedutiva pronta para ser processada. A única coisa que se precisa fazer é informar ao programa as regras que se deve seguir, e ele descobre seu próprio caminho para a informação apropriada. Muitas outras linguagens de programação exigem que se escreva as regras, as explique, especifique os percursos, e de uma forma geral trabalhe em um nível de detalhes muito maior para criar a máquina dedutiva antes de começar a obter a informação necessária. Ainda o banco de dados do Turbo Prolog é escrito com as mesmas estruturas utilizadas para escrever regras - aprenda um fato sobre o problema e saberá como tratar com ambos[27].

6.2.2 - Regras Heurísticas

Suponhamos que se tenha um grupo de especialistas para resolver um problema : todos os bairros da cidade tiveram interrupção de energia, e qualquer disjuntor ou relé-de-proteção pode estar danificado e ter causado o desligamento da chave-geral. A meta é determinar exatamente qual disjuntor causou o desligamento da chave-geral, ou qual relé-de-proteção não desligou o disjuntor, para que seja substituído.

Um Sistema Especialista para resolver este tipo de problema poderia, por exemplo, usar os seguintes fatores : qual o disjuntor mais antigo, qual relé-de-proteção ou disjuntor passou por manutenção primeiro, qual deles apresentou maior número de problemas nos últimos anos, e outros, que poderão ser utilizados para a localização do disjuntor ou relé-de-proteção danificado, ou então a verificação de um outro problema na rede. Estas informações são reunidas e é localizado o relé danificado, utilizando-se de um conjunto de regras na localização precisa dos fatores. Estas regras são as chamadas "regras heurísticas". Regras Heurísticas são regras que somente um especialista pode conhecer, não são formuladas mas possuem conhecimento[28].

Exemplos :

Regra H1 :

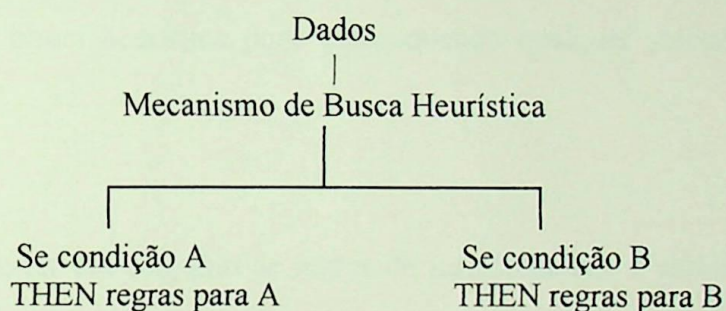
Se existe um fator ao longo da rede dentro do espaço delimitado do problema, ENTÃO este fator é que causou o desligamento do relé-geral.

Regra H2 :

Se existe mais de um fator ao longo da rede dentro do espaço delimitado do problema, ENTÃO escolhe-se o fator mais utilizado para identificação dos problemas nos últimos anos.

Produzir um mecanismo para suprimir o impossível seria tornar os processos mais eficientes por redução do número de caminhos de opiniões considerando as atividades que satisfaçam a meta. Em um Sistema Especialista, um mecanismo para suprimir o desnecessário que utilize regras heurísticas é chamado de "mecanismo de busca heurística".

Um mecanismo de busca heurística pode ser representado em caminhos como [11] :



Um mecanismo de busca heurística pode focalizar um fator mais provável que causa o problema, o Sistema Especialista pode então adaptar outro conjunto de regras para verificar o fator, utilizando-se de uma regra heurística, ou seja, regra prática, onde somente uma pessoa experiente com o problema conheceria esta regra. Por exemplo, certos relés com anos sem manutenção poderiam ser testados primeiro.

Regras Heurísticas produzem breves divisões : passos intermediários são omitidos porque estes passos são improváveis para controlar uma solução rápida de um problema.

6.2.3 - Busca Heurística

Limite de tempo e depósito de consumo disponível sempre existe em problemas de busca. Para grandes problemas este método não é prático porque muitos nós são visitados antes de uma solução ser encontrada. Isto ilustra os fenômenos de explosão combinatorial que aparece quando na resolução de um problema falta conhecimento suficiente de guia no processo inferencial[23].

Para muitas aplicações é possível de encontrar informações específicas do domínio de guia do processo de busca e reduzir a soma computacional. Isto é chamado de informação heurística, e o procedimento de busca que usa-o é chamado de método de busca heurística. A busca heurística pode parar quando qualquer solução satisfatória é encontrada.

Simplesmente ter um conjunto de regras de inferência não é suficiente, também é necessário conhecimento sobre resolução do problema como guia de aplicação destas regras, sem o qual ocorre uma explosão combinatorial, resultando na derivação de muitas correções.

Na resolução de problemas deve-se conter explosões combinatoriais. A busca heurística incorpora conhecimento sobre o espaço de soluções de busca, contendo uma explosão combinatorial. A heurística pode envolver grande quantidade de computação simbólica.

6.2.4 - Quadro-Negro

Este é frequentemente um caso que um problema pode ser também complexo para um sistema especialista resolver por si mesmo. No exemplo, um sistema especialista poderia ser capaz de identificar que o problema ocorreu em algum disjuntor, mas se precisássemos localizar o disjuntor, outro tipo de especialista seria necessário. O sistema especialista necessita de certas informações que incluídas no sistema especialista localizam a origem do problema. Então um segundo sistema especialista traria cálculos específicos e mecanismos de busca heurística para resolver o mesmo problema e recomendar qual disjuntor deverá ser substituído, neste problema particular. A comunicação de informações entre sistemas especialistas é feita por um mecanismo chamado "quadro-negro". O quadro-negro é um espaço na memória do computador em que as informações que estão armazenadas, como um sistema especialista, são "mantidas". Por isso, qualquer outro sistema especialista pode se referir a ele se necessitar de informações necessárias para o alcance das "goals" (metas)[11].

O quadro-negro é uma estrutura que contém informações que podem ser vistas por cooperação de sistemas especialistas. O que o sistema faz com estas informações vai depender da aplicação. O conceito de quadro-negro para informar, modificando e observando as alterações, é muito importante para a cooperação do sistema.

6.3 - COMPONENTES DE UM SISTEMA ESPECIALISTA

Representação idealizada de um Sistema Especialista. Um único S.E. não contém todos os componentes aqui apresentados, mas um ou mais componentes[23].

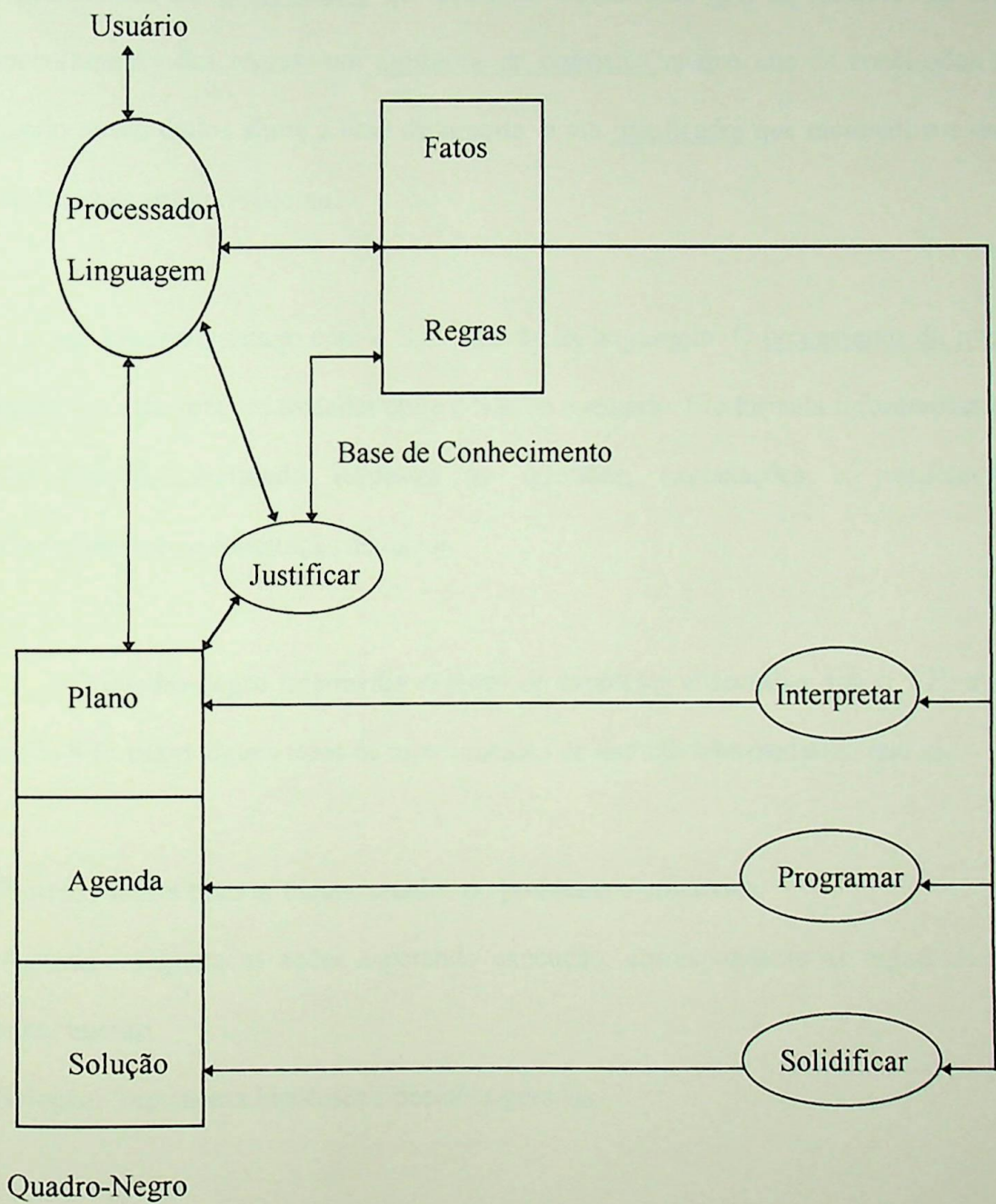


Figura 6.2 : Anatomia de um Sistema Especialista (quadro-negro).

O S.E ideal contém um processador de linguagem de comunicação entre o usuário e o sistema; um "quadro-negro" para registrar resultados intermediários; uma base de conhecimento compreendendo fatos como planejamento heurístico e regras para resolução de problemas; um interpretador que usa estas regras; uma lista de controle da ordem de processamento das regras; um ajustador da consistência que une as conclusões prévias quando novos dados altera a base de suporte; e um justificador que racionaliza e esclarece o comportamento do sistema.

O usuário interage com o S.E. através da linguagem. O processador da linguagem intermedia informações trocadas entre o S.E. e o usuário. Ele formata informações geradas pelo sistema; incluindo respostas as questões, explanações e justificações do comportamento, e solicitação de dados.

O quadro-negro intermedia registro de hipóteses e decisões que o S.E. manipula. Vários S.E. usam alguns tipos de representação de decisão intermediária, que são :

- Projeto - inclui planos, metas, estados do problema e contextos.
- Agenda - registra as ações esperando execução, correspondente as regras da base de conhecimento.
- Solução - representa hipóteses e decisões geradas.

A lista de controle mantém o controle da agenda e determina qual ação pendente pode ser executada.

O interpretador executa a escolha dos itens da agenda usando regra correspondente da base de conhecimento. Geralmente ele valida as condições relevantes das regras, liga variáveis nestas condições de soluções particulares de elementos do quadro-negro, e produz estas alterações do quadro-negro.

O ajustador da consistência tenta manter uma representação consistente das soluções de emergência.

6.4 - CRIANDO UM S.E.

Para criar um Sistema Especialista, a equipe deve consistir de um especialista e um engenheiro de conhecimento, reunir fatos, regras, e regras heurísticas para um domínio e organiza-lo dentro de programas de I.A.. Para isto deve-se ter um conhecimento de I.A.[23].

6.4.1 - Metodologia para desenvolver S.E.

Uma das características de S.E. é o caminho que ele busca para soluções. A escolha do método de busca é afetado por muitas características de um domínio, tal como o tamanho do espaço de soluções, erros nos dados, e a utilidade de abstrações. Inferência está

no centro da razão de um sistema, e falhas de organização podem devidamente resultar na resolução de problemas que são ineficientes, simples ou inseguros[29].

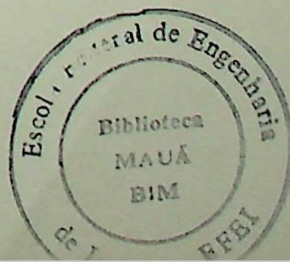
Sistemas são tipicamente construídos por um processo de desenvolvimento iterativo. Após o projeto inicial e implementação, o sistema cresce incrementalmente em largura e profundidade.

A seguir, os estágios na evolução de um S.E. :

PRIMEIRA FASE :

Envolve identificação e conceitualização do problema.

- IDENTIFICAÇÃO : durante a identificação, o engenheiro de conhecimento e o especialista trabalham juntos para identificar a área do problema e definir seu escopo. Identifica os participantes no processo de desenvolvimento, determina os recursos necessários, e decide as metas ou objetivos da construção do S.E..
- CONCEITUALIZAÇÃO : durante a conceitualização, o especialista e o engenheiro do conhecimento explicam os conceitos chaves, relações e características das informações necessárias para descrever o processo de resolução do problema num dado domínio.



SEGUNDA FASE :

Envolve formalização, implementação e teste de uma arquitetura apropriada para o sistema, incluindo constantes reformulações de conceitos, representação de reprojeto, e refinamentos do sistema implementado.

-**FORMALIZAÇÃO** : envolve mapeamento de conceitos chave e relação dentro de uma representação formal sugerida por algumas ferramentas de desenvolvimento de S.E. ou linguagem. O engenheiro de conhecimento pode selecionar a linguagem e, com ajuda do especialista, representar os conceitos básicos e relações com a sub-estrutura da linguagem.

- **IMPLEMENTAÇÃO** : durante a implementação, o engenheiro de conhecimento combina e organiza o conhecimento formalizado de produção compatível com o fluxo de informações característico do problema. O conjunto resultante de regras e estruturas de controle associados define um programa protótipo capaz de ser executado e testado.

- **TESTE** : o teste envolve a avaliação da performance do protótipo do programa e revisão conforme o padrão definido pelo especialista no domínio do programa.

6.4.2 - A Aquisição do Conhecimento

O conhecimento de um domínio têm muitas formas. Quando este conhecimento é seguro, fixo e regulado, os algoritmos de programa de computador que resolvem os problemas no domínio são mais apropriados do que um heurístico. Quando o conhecimento é dependente, mal codificado e parcialmente condenado, o S.E. engloba uma heurística aproximada não mais apropriada. Este tipo de conhecimento é raramente formulado em um estilo que permita simples tradução dentro de um programa. As técnicas de desenvolvimento tem emergido como metodologia dominante na área de S.E.. Assim o processo de extração do conhecimento de um especialista e codificação na forma de programa é um importante e difícil problema. Este processo é denominado "aquisição do conhecimento". Ele envolve definição do problema, implementação, e refinamentos, como representação de fatos e relações adquiridas de um especialista[30].

A aquisição do conhecimento é a transferência e transformação da habilidade de resolver problemas de um conhecimento origem de um programa. O potencial de origem de um conhecimento inclui o especialista humano, livros texto, base de dados, e experiências. O especialista, não necessariamente é o melhor profissional em uma área de conhecimento. O especialista estará esclarecido de uma coleção de fatos especializados, procedimentos e regras pelo domínio de uma pequena área. É necessário um analista ou engenheiro do conhecimento para que haja comunicação com o especialista e o programa.

6.4.2.1 - Modos de aquisição do conhecimento

A aquisição do conhecimento é uma pequena parte na construção de um S.E.. O engenheiro do conhecimento será um intermediário para ajudar um especialista a construir um sistema. Antes o engenheiro do conhecimento tinha um menor conhecimento do domínio do que o especialista, entretanto, os problemas de comunicação impediam o processo de transferência dentro de um programa. O vocabulário inicialmente usado pelo especialista era inadequado para resolver problemas; assim o engenheiro do conhecimento e o especialista passaram a trabalhar juntos. Um dos mais difíceis aspectos para o engenheiro do conhecimento é a tarefa de ajudar o especialista na estrutura do domínio do conhecimento, identificar e formalizar os conceitos do domínio.

O conhecimento para em S.E. pode ser adquirido em vários caminhos, todos envolvendo transferência de habilidades necessárias para alta-performance na resolução de problemas no domínio de uma origem de um programa. A origem é geralmente um especialista humano mas pode também ser dados empíricos, estudos de caso, ou outras origens da qual um especialista humano tenha adquirido conhecimento. O processo de tradução do conhecimento da origem para o programa pode ser executado por um engenheiro do conhecimento ou por um programa.

1o.) O especialista interage com um engenheiro do conhecimento ou programa para desenvolver o sistema : O engenheiro do conhecimento conversa com o especialista, mas o domínio do conhecimento é separado do resto do programa em estruturas de dados simples

chamado base de conhecimento. As principais vantagens são transparência e flexibilidade - amanhã a base de conhecimento pode ser mais facilmente estendida e modificada em alterações claras atuais ou o conhecimento ser embutido em códigos manuais.

2o.) O especialista familiarizado com a tecnologia do computador pode interagir mais diretamente com o S.E. via um programa de edição inteligente. Aqui o especialista conversa com o programa de edição antes que com um engenheiro do conhecimento.

6.4.3 - Heurística

A área de Sistemas Especialistas investiga métodos e técnicas para construção de sistemas homem-máquina com especialidade em resolução de problemas. O especialista possui conhecimento sobre um domínio particular, entende os problemas do domínio, e até resolve alguns destes problemas. O conhecimento pode ser adquirido de publicações, fatos, e teorias de livros textos e referências. O especialista humano adquire um conhecimento que não é publicado em textos públicos. Este conhecimento consiste de regras que podem ser chamadas de heurística. A heurística possibilita ao especialista humano produzir suposições onde for necessário, para reconhecer caminhos promissores do problema, e para tratar efetivamente com erros e dados incompletos. Esclarecer e reproduzir tal conhecimento é a tarefa central na construção de um Sistema Especialista.

6.5 - QUALIDADES FUNDAMENTAIS DE SISTEMAS ESPECIALISTAS

Uma característica dos S.E. é a transparência. Isto significa que o sistema é compreensível. S.E. frequentemente separa o conhecimento do domínio específico do conhecimento geral na resolução de problemas melhorando a transparência. Isto permite a análise e modificação da base de conhecimento contendo o conhecimento do domínio específico[23].

Quando falamos de S.E., a palavra especialista é necessária. Projetos devem garantir que as características de um especialista possam ser identificadas. Pesquisadores acreditam que um dos mais importantes fatores que um S.E. pode ter é a capacidade para explanação. A resolução de um problema difícil é mais favorecida por uma explanação em termos de princípios básicos de um domínio. A direção sugerida por este fator é a construção de S.E. baseados nos fundamentos de domínio prático.

Habilidade : a mais importante meta no emprego de S.E. é de alcançar o alto nível de execução que um especialista consegue em algumas tarefas. Deve-se considerar significativa a alta qualidade de produção dos resultados em tempo mínimo.

Alto-desempenho : o primeiro e mais óbvio critério para S.E. é que ele resolva os problemas a que eles são aplicados. Execução hábil depende da quantidade e qualidade do conhecimento incorporado dentro do programa

Embora o desempenho de uma tarefa é parte do comportamento especialista, ele não é suficiente para determinar que um sistema é especialista. O caminho seguido é que é considerado. O comportamento especialista parece exigir que seja evitado uma revista completa do grande número de hipóteses em favor da rápida eliminação de muitas possibilidades em cada mudança inferencial. S.E. requer uma conduta inteligente.

Outras caracterizações limitantes de um S.E. :

- estreito domínio de habilidades
- linguagem limitada para expressar fatos e relações
- princípio limitado sobre problemas e métodos de soluções (requer ajuda de um engenheiro de conhecimento)
- linguagem estilizada de entrada e saída
- explanação estilizada da linha de raciocínio;
- baixo conhecimento do seu campo e limitações;
- base de conhecimento extensa, mas baixa disponibilidade de ajuda para decisões de projeto inicial.

Para se ter um Sistema Especialista é necessário definir que tipos de informações serão necessárias na formação da base de conhecimento, para que se possa ter uma interpretação do evento ocorrido e, à partir daí, definir os outros eventos que foram gerados à partir da análise do evento ocorrido.

CAPÍTULO 7

DESCRIÇÃO DO SISTEMA

7.1 - INTRODUÇÃO

O objetivo do aplicativo de Interface Homem-Máquina em Linguagem Natural é o de fornecer mais uma opção ao usuário final de se comunicar com a máquina e, no nosso caso específico, com o Sistema ESRASE. Esta forma de comunicação adotada é a própria linguagem do usuário, ou seja, a Língua Portuguesa.

Como o projeto de uma Interface em Língua Portuguesa é de extrema complexidade, achou-se de primeira necessidade a criação de um projeto básico, voltado especificamente ao Sistema ESRASE. Mas tomando o cuidado de deixá-lo pronto para novas inclusões de módulos que viessem completar o software.

Assim, procurou-se montar as idéias principais da interface desde a obtenção do texto, o tratamento de cada item do texto, a classificação, a transformação, até a obtenção do texto na linguagem acessível pela máquina.

7.2 - INTERFACE HOMEM-MÁQUINA - PROJETO PROLOG

7.2.1 - O Desenvolvimento

A interface em linguagem natural foi desenvolvida em Linguagem Turbo PROLOG, versão 2.0, utilizando-se de algumas funções do Turbo Prolog Toolbox, ambos da empresa Borland International[31].

No desenvolvimento do programa optou-se pela estrutura modular. Esta técnica de modularização procura diminuir ao máximo a inter-relação entre módulos, deixando-os mais independentes entre si. Esta técnica aumenta a flexibilidade e a facilidade de alterações e inclusões de novos módulos dentro do sistema.

O propósito inicial do projeto é desenvolver um projeto básico que atinja o objetivo proposto, que é a construção e o entendimento de frases usando termos referentes a uma base de dados específica para que seja interpretada e analisada. O tema é de uma grande complexidade e para se contornar todas as possibilidades seria gasto muito tempo, o que não é o propósito.

Inicialmente foi desenvolvido um programa que analisasse a estrutura da frase, em grupos menores de "tokens" e depois em grupos maiores, verificando a gramática que existia por trás do texto.

Em seguida, desenvolveu-se um programa que interpretasse o texto já classificado. O programa transformaria as frases do texto em linguagem natural em linguagem ESRASE.

Por último, desenvolveu-se um programa que, à partir dessas informações, conseguisse deduzir outras informações sem que fosse necessário a intervenção do usuário.

Finalmente, todos os módulos foram agrupados, tornando-se um sistema compacto com uma interface amigável, ficando totalmente transparente sua utilização pelo usuário final.

7.3 - INTERFACE COM O USUÁRIO

A interface com o usuário fornece a comunicação entre o usuário e o sistema, permitindo ao usuário a manipulação do Banco de Dados (inserir, deletar e alterar os dados dos registros), envio de mensagens e obtenção de respostas do sistema. Para que seja efetuada esta comunicação foram criadas as seguintes telas[31] :

- tela de inserção de dados;
- tela de ajuda do usuário;
- tela de mensagens.

7.3.1 - Tela de Inserção de Dados

A tela de inserção de dados o usuário pode escrever a frase em linguagem natural

Arquivos Editar Listar Simular Manobrar Sair

ENTRADA DE EVENTOS EM LINGUAGEM NATURAL

Escreva o evento em linguagem natural :
O disjuntor 3K4 atuou

Figura 7.1 : Tela de Inserção de Dados

7.3.2 - Tela de Ajuda ao Usuário

A tela de ajuda ao usuário aparece somente quando o usuário escreve uma frase com alguma palavra escrita incorretamente e o programa conseguiu encontrar alguma palavra semelhante no banco de dados.

Esta tela é apresentada na forma de menu de barra e o usuário poderá escolher a palavra que lhe convêm, escrita de forma correta, ao invés de reescrevê-la.

Arquivos Editar Listar Simular Manobrar Saidas

ENTRADA DE EVENTOS EM LINGUAGEM NATURAL

Escreva o evento em linguagem natural :
O dijuntor 3K4 esta deligado

(a)



Arquivos Editar Listar Simular Manobrar Sairas
ENTRADA DE EVENTOS EM LINGUAGEM NATURAL

>> Nao reconheco palavra : DIJUNTOR

POSSIVEIS SUBSTANTIVOS
DISJUNTOR

(b)

Arquivos Editar Listar Simular Manobrar Sairas
ENTRADA DE EVENTOS EM LINGUAGEM NATURAL

>> Nao reconheco palavra : DIJUNTOR

>> Nao reconheco palavra : DELIGADO

POSSIVEIS ADJETIVOS
DESLIGADA
DESLIGADO

(c)

Arquivos Editar Listar Simular Manobrar Sair

ENTRADA DE EVENTOS EM LINGUAGEM NATURAL

Escreva o evento em linguagem natural :
A protecao 3K4 atuou

(d)

Arquivos Editar Listar Simular Manobrar Sair

ENTRADA DE EVENTOS EM LINGUAGEM NATURAL

NRO RELE INVALIDO PARA ESTE CIRCUITO :4

RELES VALIDOS P/ CIRCUITO 3K
67n - rele de sobrecorrente direcional
67 - rele de sobrecorrente direcional
51n - rele de sobrecorrente temporizado
51 - rele de sobrecorrente temporizado
50n - rele de sobrecorrente instantaneo
50 - rele de sobrecorrente instantaneo
32 - rele direcional de potencia
27 - rele de subtensao
21-3 - rele de distancia
21-2 - rele de distancia
21-1 - rele de distancia

(e)

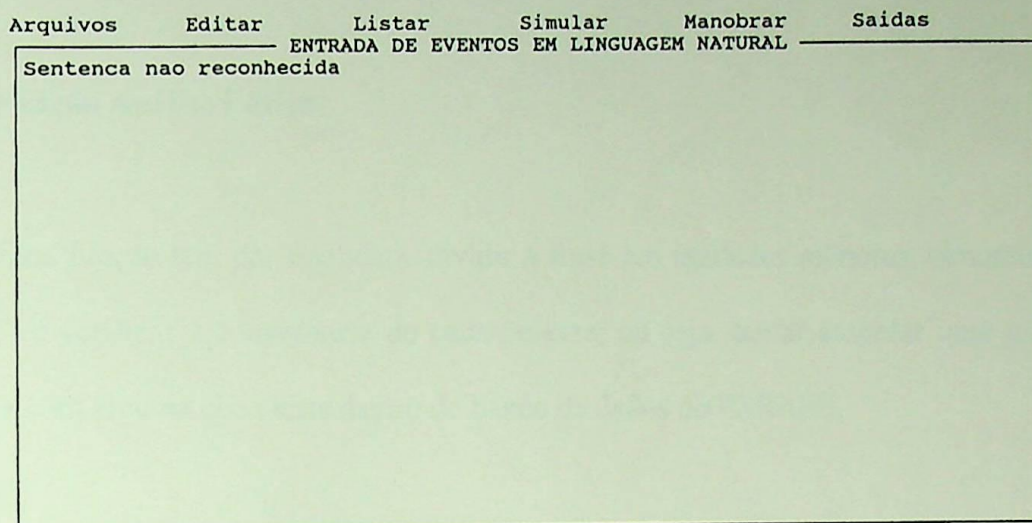
Figura 9.2 : Tela de Ajuda ao Usuário

7.3.3 - Tela de Mensagem

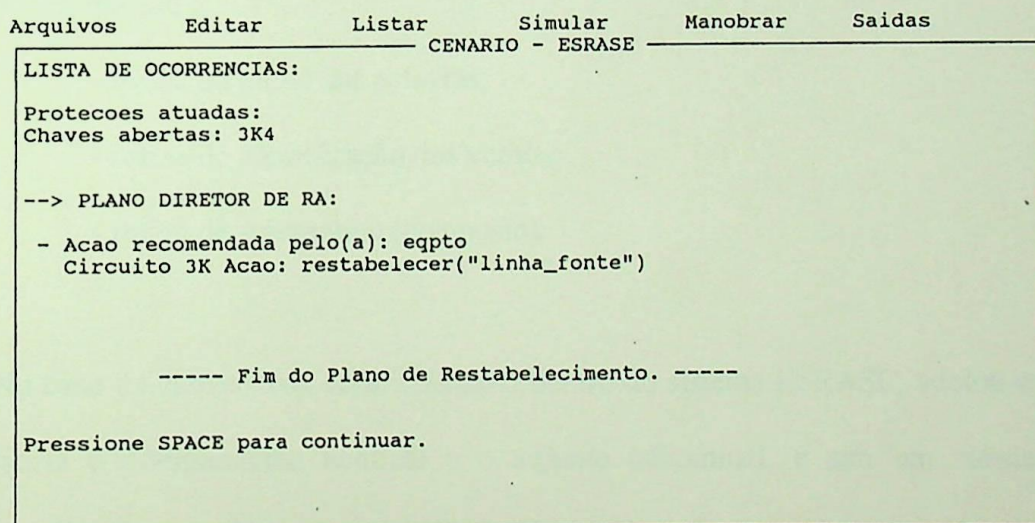
As mensagens que são fornecidos pelo programa são somente mensagens de erro. Como, por exemplo, "Palavra não reconhecida", que é escrita quando o programa não encontra nenhuma palavra parecida no banco de dados; ou, "Sentença não reconhecida", quando uma determinada sentença não apresenta função sintática nenhuma.

Arquivos	Editar	Listar	Simular	Manobrar	Saidas
ENTRADA DE EVENTOS EM LINGUAGEM NATURAL					
Escreva o evento em linguagem natural :					
O homem 3K4 caiu					

(a)



(b)



(c)

Figura 9.3 : Telas de Mensagem

7.4 - FUNÇÕES LÓGICAS

7.4.1 - Função Análise Léxica

Esta função tem por finalidade dividir a frase em unidades menores, denominadas "tokens", e verificar a consistência de cada palavra, ou seja, tentar associar uma palavra qualquer com alguma que exista dentro do banco de dados do ESRASE.

Para identificação da palavra foram montadas algumas rotinas com regras lexicas para flexibilizar ao máximo a busca pelas palavras e minimizar o tamanho do banco de dados do sistema.

As rotinas são :

- rotina de plural das palavras;
- rotina de identificação dos verbos;
- rotina de substantivo (composto).

No caso do substantivo, relacionando-o dentro do sistema ESRASE, adotou-se que não existiria o complemento nominal e o adjunto adnominal, e sim um substantivo composto. Como exemplo temos : "relé", "relé geral", "relé de temperatura do óleo", e outros.

Na pesquisa da palavra, como ainda não posso identificar o sentido morfológico da mesma, o programa pesquisa todo o banco. Para isto criou-se algoritmos de comparação para procura da palavra. Com isso é apresentado ao usuário um possível banco de palavras parecidas, caso não seja identificada a palavra no banco de dados.

7.4.1.1 - Rotina de plural das palavras

Foi montada uma rotina de plural, baseada nas regras da língua portuguesa, que é somente utilizada caso não encontre a palavra no banco de dados. A procura pelo seu plural é feita da seguinte forma :

Exemplo frase entrada : "Os relés 1K21 e 2K27 atuaram"

Em uma primeira análise do substantivo "relés", o programa não encontra o substantivo. No banco de dados somente estão as palavras no singular. Em uma segunda análise o programa separa o radical "rel" e aplica em todas as palavras do banco que começam com este radical, unindo o radical às opções do plural para formar e encontrar a palavra.

7.4.1.2 Rotina de Identificação dos Verbos

Foi criado um algoritmo para verificar a existência de verbos no banco de dados.

Na maioria dos verbos da língua portuguesa existe uma primeira parte da palavra que não varia (radical), e uma segunda parte variável, denominados verbos regulares. Em uma minoria, existem verbos que variam por inteiro, denominados verbos irregulares.

O programa separa o radical da palavra a ser analisada e por comparação dos radicais encontra os que mais se encaixam com esse no banco de dados. E a partir das listas de flexões de modo e tempo para uma determinada conjugação o programa tenta encontrar o verbo. Caso não encontre, ele testará o verbo com todos os verbos irregulares.

Exemplo :

"Os relés 1K21 e 2K27 atuaram"

Na análise do verbo "atuaram", temos no banco de dados :

verb ("atu", "ar")

verb ("oper", "ar")

verb ("abr", "ir")

conjugação ("r", "ar",["ada", "ado", "ou", "amos", "aram"])

O programa separa o radical (determinado no programa como sendo composto pelas duas primeiras letras da palavra) "at", procura as palavras semelhantes e encontra :

verb ("atu", "ar").

De posse deste fato ele procura pelo verbo através da conjugação.

7.4.1.3 - Rotina de substantivo (composto)

Foi criado um algoritmo que trata o substantivo com o complemento nominal ou adjunto adnominal como sendo simplesmente um substantivo composto.

Como exemplo podemos citar :

No banco de dados : subst("relé").

subst("relé geral").

subst("relé de temperatura e óleo").

Este procedimento foi adotado pois não é a intenção do pacote computacional classificar a frase nos mínimos detalhes. E com isso a análise para os próximos passos se tornaria mais transparente.

Na procura de cada palavra, como ainda não foi identificado o sentido morfológico desta, ocorre uma pesquisa em todo o banco de dados. Para isso foram formulados alguns algoritmos de comparação para otimizar esta procura. Assim, é possível que uma palavra escrita de maneira incorreta seja aceita, desde que tenha um grande coeficiente de semelhança com uma das palavras do banco de dados.

Por exemplo : O usuário escreve a palavra, classificada como substantivo, "dijuntor". Como podemos perceber a palavra foi escrita incorretamente, a grafia correta seria "disjuntor". O programa não encontra no banco de dados a palavra "dijuntor", mas procura as mais parecidas e mostra ao usuário um conjunto com as possíveis opções de palavras corretas.

7.4.2 - Função Análise Morfológica

Esta função tem por finalidade identificar morfológicamente cada palavra de uma frase, gerando uma lista com todas as palavras e suas respectivas identificações.

Existem os "tokens" que são apenas termos auxiliares na frase. Eles não são termos essenciais ao entendimento da idéia expressa pelo usuário. Dentro do programa os "tokens" são identificados e ignorados na análise da frase, não fazendo parte da lista de identificação das palavras. Os "tokens" podem ser a vírgula "," e a conjunção "e".

Exemplo :

```
artigo("O"),  
subst("DISJUNTOR"),  
adjetivo("2K21-1"),  
verb("ATUADO").
```

7.4.3 - Função Análise Sintática

Esta função tem por finalidade deduzir uma estrutura gramatical em que a frase se encaixe. Este processo é feito pela junção da lista obtida dos blocos menores, identificando suas funções sintáticas.

Uma sentença pode ser classificada sintaticamente como frase nominal e/ou frase verbal.

Exemplo : "O RELÉ 2K27 ATUOU".

Ficaria classificado sintaticamente como[31] :

```
sent([snominal-asa(artigo("O"), subst("RELÉ"), adjetivo("2K27"))], verb("ATUOU")).
```


7.4.4 - Função Análise Semântica

Esta função tem por finalidade processar o texto escrito pelo usuário. Através da utilização da frase devidamente classificada, é montado uma base de dados com os eventos descritos pelo usuário.

O programa se utiliza de uma base de conhecimento pré-estabelecido, onde através de regras e fatos determina os termos do texto que são importantes. A esses termos é associado um evento do sistema, e com as informações adicionais do texto completa-se o fato. Este fato representa uma ocorrência para o Sistema ESRASE.

Feita a efetivação da ocorrência na base de dados, a informação passada pelo usuário em Linguagem Natural é devidamente transformada na Linguagem Prolog para o Sistema ESRASE.

Exemplo 1 :

Ocorrência em Linguagem Natural :

"O RELÉ DE SUBTENSÃO DO CIRCUITO 2K ATUOU".

Ocorrência em Linguagem Prolog para o ESRASE :

evento-proteção("2", "K", "27").

Exemplo 2 :

Ocorrência em Linguagem Natural :

"O RELÉ 2K25 E O DISJUNTOR 2K4 ESTÃO ABERTOS".

Ocorrência em Linguagem Prolog para o ESRASE :

evento-proteção("2", "K", "25").

evento-chave-manobra("2", "K", "4").

CAPÍTULO 8

DESCRIÇÃO DO PROGRAMA

8.1 - A SENTENÇA EM LINGUAGEM NATURAL

O usuário, do sistema, deverá digitar uma frase referente a algum acontecimento de falha na subestação. O computador deverá “entender” ou não a frase digitada para que seja feita a análise, à partir de uma divisão da mesma em unidades menores denominadas tokens”.

A frase, que é denominada de sentença, pode ser formada por sintagma nominal, no programa caracterizado como s-nominal; verbo e complemento, num nível hierárquico maior.

Uma frase pode ser formada por :

- s-nominal + verbo

Ex. : A proteção 2K27 atuou.

- s-nominal + verbo + complemento

Ex. : A proteção 27 atuou no circuito 2K.

- verbo + complemento

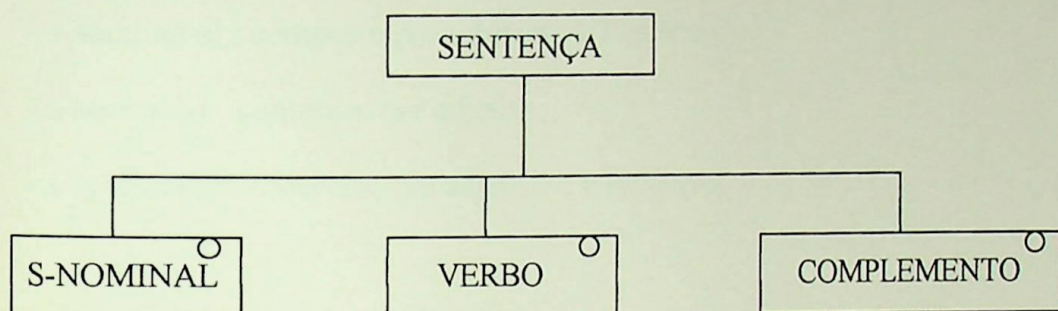
Ex. : Caiu medida 2K.

OBS : - O s-nominal é a classificação sintática da frase.

- Uma frase do tipo : Sobretensão no circuito 2K

é aceita pelo programa, pelo jargão típico do usuário.

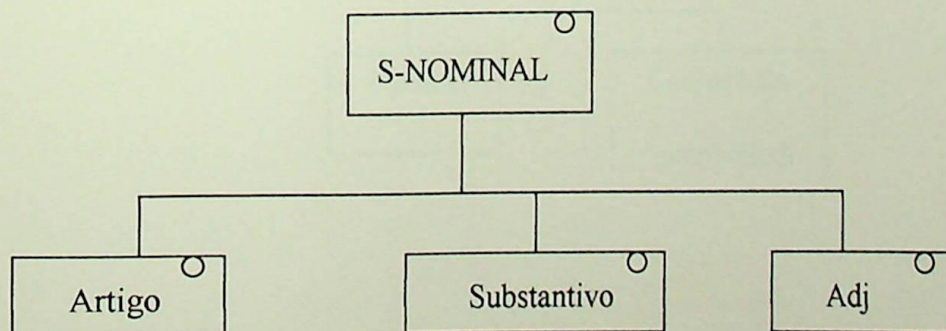
Fluxo de Dados (dentro do Sistema)



onde : ○ significa e/ou

8.1.1 - Detalhando o S-NOMINAL

O s-nominal corresponde ao sujeito ou objeto de uma frase. Pode ser composto de artigo, substantivo e adjetivo.



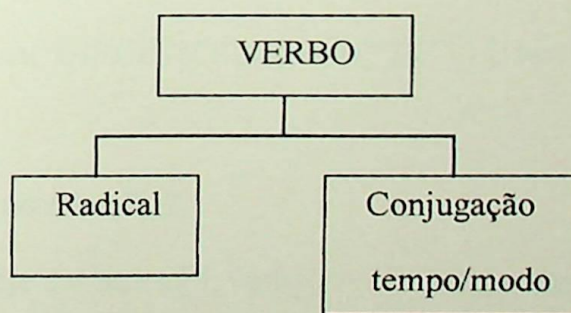
Pode ser formado por :

- s-nominal-asj : composto por artigo, substantivo e adjetivo.
- s-nominal-as : composto por artigo e substantivo.
- s-nominal-s : composto por substantivo.
- s-nominal-sj : composto por substantivo e adjetivo.
- s-nominal-j : composto por adjetivo.
- s-nominal-js : composto por adjetivo e substantivo.

8.1.2 - Detalhando o VERBO

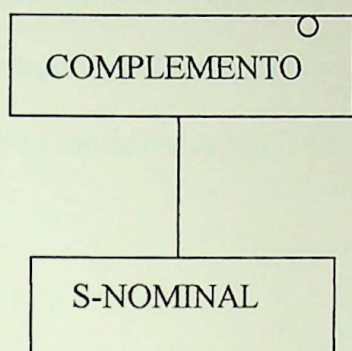
O verbo é composto pelo radical ('verb') e sua classificação de tempo/modo.

O radical ainda possui uma informação da conjugação do verbo. Esta conjugação serve para que se determine a classificação tempo/modo.



8.1.3 - Detalhando o COMPLEMENTO

O complemento é formado pelo s-nominal existente, já definido.



Exemplo :

- A proteção 3K21-3 atuou.

Do programa : Prolog Object = sent ([snominal-asj(artigo("A"),"PROTEÇÃO",
adjetivo("3K21-3"))],verb("ATUOU"))

- O relé de tempo do circuito 2K abriu.

Do programa : Prolog Object = sent ([snominal-as(artigo("O"),"RELE DE TEMPO"),
snominal-sa("CIRCUITO",adjetivo("2K"))], verb("ABRIU"))

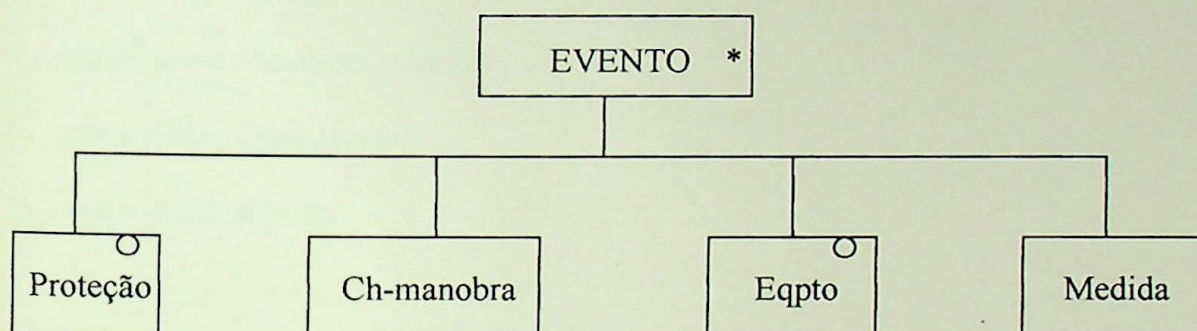
- Atuou proteção no relé 6K32.

Do programa : Prolog Object = sent ([], verbp("ATUOU",[snominal-s("PROTEÇÃO"),
snominal-sj("RELE", adjetivo("6K4"))]))

8.2 - O EVENTO OCORRIDO / OUTROS EVENTOS

O Evento é gerado à partir da análise sintática e morfológica da frase, que é constituído por um número mínimo de dois, para cada frase analisada.

É composto pelo evento proteção, evento chave de manobra, evento equipamento e evento medida, onde todos são E/OU, exceto, chave de manobra e medida.



onde :

○ : significa e/ou

* : significa que pode haver vários.

Exemplo :

evento-proteção("6", "K", "67");

evento-chave-manobra("6", "K", "4")

evento-eqpto("6", "K", linha-carga("SE1", "SE5"))

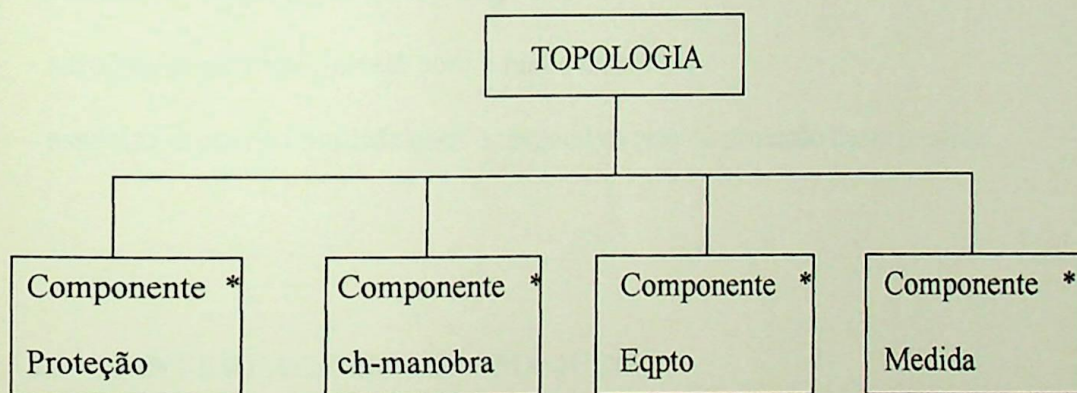
evento-medida("6", "K", 0, 0, 0, 0)

8.2.1 - Topologia da Subestação

Fruto do Sistema Estrate, descreve toda a topologia da subestação. Cada subestação possui uma topologia diferente. Abaixo, descreve-se apenas os elementos utilizados no Sistema.

É composto de :

- componente proteção
- componente chave-de-manobra
- componente equipamento
- componente medida



onde : * significa que pode haver vários.

Exemplo :

- proteção ("12", "F", "50", "off").

onde : 12 é o número do relé

F é a classe

50 é o número do relé

off é o estado do componente

- chave-manobra (“1”, “K”, “5”, “off”)
- eqpto (“3”, “K”, linha-fonte(“SE1”, “SE3”, “Iniciador”), “on”)

8.2.2 - Associações

O Auxiliar é apenas usado para :

- associar o substantivo com o flag indicativo do evento;
- associar o número do relé com o nome do relé; e
- associar o evento medida com o respectivo relé de proteção desta medida.

8.3 - A INTERFACE HOMEM/MÁQUINA

8.3.1 - Conhecimento da Língua

O ambiente utilizado no conhecimento, traduzirá uma solicitação de ação do usuário em L.N.. À partir daí, ele verificará a frase e solicitará um complemento caso a frase esteja incompleta, procurando-se detectar a “intenção” do usuário através de

perguntas em L.N.. As solicitações do usuário são feitas em L.N.. Usa-se do conhecimento linguístico da Língua Portuguesa, para que seja detectado a intenção do usuário. Este conhecimento, fornecido através de regras, será padrão para todos os aplicativos, correspondente à morfologia das palavras[14].

É importante salientar que o nosso objetivo não é o de analisar a frase nos mínimos detalhes, e sim de interpretar a intenção do usuário. Portanto nos preocupamos em fazer uma base de conhecimento simples, mas nem tão pouco eficaz para nosso propósito.

8.3.2 - Conhecimento Morfológico

À partir da determinação do domínio que se deseja trabalhar, basta agora adequar cada elemento deste domínio aos parâmetros que regem o sistema.

Existem no sistema palavras (ou termos da língua) que estarão sempre presentes, não importando qual sistema especialista é destinado. Esses termos auxiliam o usuário a completar um fato, uma ação. Existem palavras que são chaves para o sistema e que variam de acordo com o sistema especialista, estas deverão ser devidamente classificadas.

É através destas palavras chaves que o sistema conseguirá, através do motor de inferência, gerar os eventos ocorridos, relatados pelo usuário.

A morfologia das palavras compõe-se da classificação das palavras como artigo, substantivo, adj e verbo.

8.3.3 - Conhecimento Técnico

Dentro de um determinado sistema especialista existirão determinados parâmetros que serão tratados como chaves dentro do conhecimento.

Estes parâmetros serão responsáveis pela identificação do “tipo componente” e do “componente” que desenvolverá a “ação”.

Para o sistema, o “tipo-componente” sempre será um substantivo, o “componente” sempre será um adjetivo e a “ação” será representada pelo verbo.

Como exemplo, citamos um sistema especialista de uma vídeo locadora, onde um “cliente”, uma “fita” seriam os “tipos-componente” para o sistema. Já o elemento cliente, e o elemento fita, ou seja, o código de identificação do cliente e o número de identificação da fita, serão os adjetivos. Os verbos são os identificadores da ação, portanto, também identificadores do evento ocorrido.

Existem determinados sistemas especialistas onde o verbo não têm um significado importante na identificação do evento. Nestes casos o identificador será o substantivo.

Como um exemplo, para determinar os eventos da falta ocorridos em uma subestação, onde proteção, relé, disjuntor são os substantivos, 2K4, 5K21-3 são os adjetivos, o verbo neste caso sempre significará uma ação de desligamento.

8.3.4 - Um Exemplo Aplicativo

- CAPTAR SENTENÇAS EM LINGUAGEM NATURAL

Com uma base de dados já montada, e a ocorrência de um evento decorrente de uma subestação, é elaborado uma sentença que descreva o evento em Linguagem Natural, partindo da forma própria do usuário de se expressar[31].

A frase deverá ser referente a algum evento ocorrido no sistema.

À partir da elaboração e a digitação da mesma, a frase é dividida em unidades menores, denominadas “tokens”. Verifica-se a consistência de cada palavra, ou seja, tenta-se associar cada uma das palavras com alguma que exista dentro do banco de dados do ESRASE.

- ANÁLISE MORFOLÓGICA E SINTÁTICA

Para que seja feito a análise morfológica e sintática da frase, o Sistema faz uso de uma base de dados, aqui denominado PALAVRAS, que corresponde a um conjunto das palavras e expressões que servem para descrever um problema dentro de um sistema específico, termos da língua portuguesa e termos técnicos, denominado banco de fatos.

- Análise Morfológica

Esta função tem por finalidade identificar morfológicamente cada palavra de uma frase, gerando uma lista com todas as palavras e suas respectivas identificações.

- Análise Sintática

Esta função tem por finalidade deduzir uma estrutura gramatical em que a frase se encaixe. Este processo é feito pela junção da lista obtida dos blocos menores, identificando suas funções sintáticas.

O próximo passo seria a interpretação significativa da frase já analisada, de onde formou-se uma lista classificada morfológicamente e agrupada sintaticamente em grupos maiores.

- INTERPRETAÇÃO SIGNIFICATIVA

O programa se utiliza de uma base de conhecimento pré-estabelecido, onde através de regras e fatos determina os termos do texto que são importantes. A esses termos é associado um evento do sistema, e com as informações adicionais do texto completa-se o fato.

- CORRELAÇÃO DE EVENTOS

À partir do evento ocorrido descrito pelo usuário, já analisado morfológica e sintaticamente, verifica-se seus relacionamentos, suas ocorrências, determina-se as consequências referentes a este evento e tenta-se descobrir as causas que o ocasionaram.

Feita a efetivação da ocorrência na base de dados, a informação passada pelo usuário em Linguagem Natural é devidamente transformada na Linguagem Prolog.

Após a frase ser classificada e analisada, é gerado um conjunto de eventos.

- EDITAR D.B.

É composto por palavras, formado por :

- Termos da Língua Portuguesa : composto de verbos, artigos, substantivos e adjetivos

- Termos Técnicos : composto de substantivo e adjetivos

Como o Sistema usa uma base de dados específica, esses dados poderão ser constantemente atualizados, ou processadas alterações, ou mesmo deletados. É constituído à partir dos dados digitados e específicos para uma aplicação. Esses dados correspondem ao verbo, artigo, adj, substantivo.

Associados a este processo está o D.B. ASSOCIAÇÕES, que armazena as palavras editadas pelo usuário e que serão associadas a um flag indicativo para que sejam classificadas e associadas aos “tokens”.

Poderemos verificar a comunicação interface homem-máquina em Linguagem Natural através da figura que segue, e obter maiores informações no Anexo II :

CAPÍTULO 9

CONCLUSÃO

A crescente necessidade da utilização da informática para o controle de processos e para a ajuda e aconselhamento aos usuários em situações e tarefas complexas têm gerado a necessidade da implementação de sistemas que tenham um desempenho cada vez melhor tanto na qualidade da análise quanto em sua velocidade e confiabilidade. Neste quadro, os sistemas baseados em regras de produção têm ganho um espaço importante, pois vêm ocupar uma lacuna que anteriormente não podia ser coberta eficientemente pela informática.

Uma das áreas que não estavam sendo contempladas era a da tradução de textos e da linguagem natural. Por tradução, entende-se o processo pelo qual uma linguagem (de programação, meta-linguagem, ou uma linguagem humana) é transformada em uma outra com estrutura, elementos e regras gramaticais específicas. A linguagem natural neste contexto é a língua portuguesa, que deve ser compreendida pela máquina, mesmo contendo alguns erros gramaticais ou informações incompletas.

Esta dissertação de mestrado teve a finalidade de fornecer uma contribuição no domínio da linguagem natural aplicada aos processos de controle em engenharia de

potência. Ela apresentou e analisou diversas estruturas de linguagem e procedimentos de armazenamento de dados, que melhor se coadunam com as informações tratadas nesta área de conhecimento.

Após esta etapa de análise, escolheu-se um aplicativo que pudesse receber as estruturas e procedimentos escolhidos, gerando-se assim um exemplo ilustrativo do processo. O aplicativo escolhido foi o ESRASE, que tem por finalidade auxiliar o operador de uma subestação em seu processo restaurativo. Este aplicativo foi escolhido por dois motivos principais. Primeiramente, por ser ele escrito em Turbo-Prolog, seus dados de entrada tinham uma estrutura bastante complexa para os não-programadores. O segundo motivo foi o módulo de explanação dos resultados que fornecia informações sobre os procedimentos de chaveamento de disjuntores e chaves utilizando-se uma estrutura bastante rígida e não amigável, fato que poderia ocasionar falsas interpretações ou ações errôneas. Como vantagem adicional ao processo, pode-se citar que o uso da linguagem natural aumenta a rapidez da implementação das ações do restabelecimento, bem como as padroniza e uniformiza.

Um outro aspecto interessante, que favoreceu ainda mais esta aplicação foi que o processo de interpretação do usuário do sistema (ou seja, o operador da subestação) é normalmente afetado por eventuais pressões psicológicas, pois o mesmo utiliza o sistema em casos de desligamentos ou panes em que diversos consumidores estão privados do recebimento de energia.

A implementação desses programas aconteceu um Turbo-Prolog, por ser esta linguagem própria ao armazenamento de regras de produção dos sistemas especialistas. Além disto, como o aplicativo utilizado tinham sido escrito nesta linguagem não ocorreriam problemas de integração e compatibilidade, os quais devem ser tratados individualmente e fugiriam ao escopo desta dissertação.

O sistema proposto foi testado para diversas situações junto ao ESRASE apresentando bom resultado na interpretação das sentenças nos dois caminhos de tradução. Entretanto, a aceitação do usuário ao sistema deverá ser examinada no decorrer do processo de sua implantação, integrando o operador, o máximo possível, nas definições do projeto final.

Como continuação deste trabalho, pode-se citar a necessidade da investigação de novas funções e sua integração ao processo lingüístico de tradução tendo como objetivo aumentar a capacidade de explanação do sistema. Outro aspecto que pode ser estudado com maior profundidade é o armazenamento de informações de diferentes tipos gerando um sistema mais flexível para a manipulação do conhecimento.

Este trabalho poderá vir a ser complementado também com um módulo de treinamento e outro visual e auditivo, que possibilite uma melhor utilização dos novos recursos da informática.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] - Ribeiro, Guilherme M., - “Aplicação de Sistemas Especialistas e Engenharia do Conhecimento no Restabelecimento Automático de Subestações”, Tese de Mestrado - EFEI - 1993
- [2] - Ribeiro, Guilherme M. and Lambert-Torres, G., “ESRASE - Expert System for Automatic Restoration of Substations”,. III Simposium of Specialists of Electric Operational and Expansion Planning, Belo Horizonte, 1992.
- [3] - Lambert-Torres, G., Valiquette, B., Ribeiro, G.M., and Do, X.D., “ProControl - A Hybrid Expert System for Power System Protection and Control”, II International Symposium on Expert System Applications in Power Systems (ESAP), Melbourne, Austrália, 1993.
- [4] - Lambert-Torres, G., “Inteligência Artificial”, Escola Federal de Engenharia de Itajubá (EFEI), Notas de Aula, Itajubá-MG, 1992.
- [5] - Bittencourt, Guilherme, - “JAI I - Tópicos em Representação de conhecimento”, Tese de Mestrado, INPE - São José dos Campos - S.P.
- [6]- Weizenbaum, J., “ELIZA - A Computer Program for the Study of Natural Language Communication between Man and Machine”, Communication of the ACM, Vol. 9, Número 1, Jan.1966
- [7] - Setzer, Valdemar W. - “Bancos de Dados - Conceitos, Gerenciadores, Projeto Lógico, Projeto Físico “, 3a Edição Revista, Editora Edgard Blucher Ltda, 1985.

- [8] - Savadovsky, Pedro, - "Introdução ao Projeto de Interfaces em Linguagem Natural", Projeto Estra - Editora SID Informática SA, 1988
- [9] - Rumpel, Dieter; Krost, Gerhard, - "Natural Language Interface and Database Issues in Applying Expert Systems to Power Systems", Proceedings of the IEEE, Vol. 80, número 5, May 1992
- [10] - Wallace, M., Communicating with Database in Natural Language, Ellis Horwood Limited, 1984
- [11] - Levine, Robert I.; Drang, Diane E.; Edelson, Barry, - "A Comprehensive Guide to AI and Expert Systems", McGraw Hill - 1986
- [12] - Brachman, Ronald J.; Fikes, Richard E.; Levesque, Hector J., - "Krypton : A Functional Approach to Knowledge Representation", IEEE Computer, October 1983
- [13] - Cunha, João B. S., - "Banco de Dados para Informações Acadêmicas e de Produção Científica", Tese de Mestrado, PUC/RJ - 1984
- [14] - Cunha, Celso F. da - "Gramática da Língua Portuguesa", 6a. Edição, Rio de Janeiro, FENAME, 1980
- [15] - Robinson, Phillip R., - "TURBO PROLOG - Guia do Usuário", Tradução : Lars Gustav Erik Unonius, Revisão Técnica : Geraldo Coen, McGraw - Hill, 1987
- [16] - Li, Deyi, - "A PROLOG Database System", Department of Computer Science, Editora Heriot-Watt University, Edinburgh, UK, 1984

- [17] - Casanova, Marco A.; Giorno, F.A.C., - "Programação em Lógica", V Escola de Computação, B.H., 1986
- [18] - Kowalski, Robert, - "Logic for Problem Solving", Nilsson, Nilsson J., Editor, Imperial College - N.K 1981
- [19] - Lucas, Robert, - "Database Applications Using Prolog", Ellis Horwood Limited - Great Britain -1988
- [20] - Kowalski, R., - "Logic for Data Description", Logic and Database, Plenum Press, 1978
- [21] - Ullmano, J. D., - "Principles of Data Base Systems", Computer Science Press, New York, 1980
- [22] - Sá, José E. A., - "Conversão de Sistemas Convencionais para o Enfoque de Banco de Dados", Tese de Mestrado, PUC/RJ - 1982
- [23] - Hayes-Roth, Frederick; Waterman, Donald A.; Lenat, Douglas B., - "Building Expert Systems", Addison-Wesley Publishing Company, Inc. - 1983
- [24] - Stefik, M.; Aikins, J.; Balzer, R.; e outros, - "The Organization of Expert Systems, A Tutorial", Artificial Intelligence, North-Holland, 1982
- [25] - Keller, Robert, - "Tecnologia de Sistemas Especialistas - Desenvolvimento e Aplicação", Makron Books - 1991

- [26] - Walker, A., "Knowledge Systems and Prolog", Michael McCord, 1987
- [27] - Kowalski, R., - "Prolog as a Logic Programming Language", Department of Computing, Imperial College - NY 1981
- [28] - Aragon, D.; Castro, - "Um Sistema formal para representação de conhecimento", Anais do II Simpósio Brasileiro de Inteligência Artificial, 1985
- [29] - Nebendahl, D., "Expert Systems, Introduction to the Technology and Applications", Siemens, 1988
- [30] - Harmon, P. and King, D., - "Expert Systems - Artificial Intelligence in Business", New York : John Wiley & Sons Inc., 1985.
- [31] - Próspero Siqueira, Idmea Semeghini - CNPq/IP-USP; Velloso, Helmer Martin - UNICAMP - "Interação Homem/Máquina diante de Interface em Linguagem Natural com Português Escrito",
- [32] - Gane, Chris; Sarson, T., - "Análise Estruturada de Sistemas", Wesley, USA, 1975

ANEXO I

Formulários do MER :

DESCRIÇÃO DA ENTIDADE	
PROJETO : Interface Homem-Máquina em Linguagem Natural	
EQUIPE : Orientador : Prof. Germano Lambert Torres Coorientador : Prof. João Bosco Schumann Cunha Mestranda : Maria Silvana Peres Colaborador : Renato Ferreira Fernandes Júnior	
NOME : Termos da Língua	CÓDIGO : E01
DEFINIÇÃO : É o conjunto de palavras que auxiliam o usuário na complementação do evento-ocorrido	
ATRIBUTOS : substantivo, adjetivo, artigo, verbo	
IDENTIFICADORES : substantivo, adjetivo, artigo, verbo	

DESCRIÇÃO DA ENTIDADE

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :
 Morfologia Palavras

CÓDIGO :
 E02

DEFINIÇÃO :

É o conjunto de palavras que formam o vocabulário do interpretador. São chamadas por artigo, substantivo, adjetivo, verbo

ATRIBUTOS :

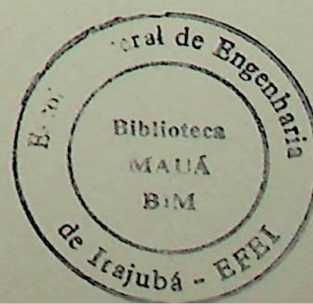
substantivo, artigo, adjetivo, verbo

IDENTIFICADORES :

substantivos, artigos, adjetivos, verbos

RESTRIÇÕES DE ACESSO :

Qualquer operador



DESCRIÇÃO DA ENTIDADE

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Regras da Língua

CÓDIGO :

E03

DEFINIÇÃO :

Composto pelas regras da língua-portuguesa

ATRIBUTOS :

tempo-modo, plural, sufixo

IDENTIFICADORES :

tempo-modo, plural, sufixo

DESCRIÇÃO DA ENTIDADE

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Artigos

CÓDIGO :

E04

DEFINIÇÃO :

Elemento que compõe o sujeito (ou o objeto)

ATRIBUTOS :

artigo (palavra)

IDENTIFICADORES :

artigo (palavras)

DESCRIÇÃO DA ENTIDADE

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :
Substantivos

CÓDIGO :
E05

DEFINIÇÃO :
Palavra que identifica o agente da ação.

ATRIBUTOS :
Proteção, chave-de-manobra, medida, alimentador

IDENTIFICADORES :
Proteção, chave-de-manobra, medida, alimentador

DESCRIÇÃO DA ENTIDADE

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :
Adjetivos

CÓDIGO :
E06

DEFINIÇÃO :
Palavra que exprime uma qualidade de um substantivo

ATRIBUTOS :
adjs (palavra)
circuito, número do relé

IDENTIFICADORES :
adjs (palavra)

DESCRIÇÃO DA ENTIDADE

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :
Verbos

CÓDIGO :
E07

DEFINIÇÃO :
Conjunto de palavras que corresponde a ação do sujeito.

ATRIBUTOS :
verb (radical, conjunção)

IDENTIFICADORES :
verb (radical, conjunção)

DESCRIÇÃO DA ENTIDADE

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Tipo termo técnico

CÓDIGO :

E08

DEFINIÇÃO :

São os identificadores dos tipos-de-eventos.

ATRIBUTOS :

proteção, chave-de-manobra, medida, alimentador

IDENTIFICADORES :

proteção, chave-de-manobra, medida, alimentador

DESCRIÇÃO DA ENTIDADE

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Ocorrência de termo técnico

CÓDIGO :

E09

DEFINIÇÃO :

São os identificadores dos componentes.

ATRIBUTOS :

adjs (palavras)

IDENTIFICADORES :

círculo, nro-relé, adjetivo

DESCRIÇÃO DA ENTIDADE

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Tipo Evento

CÓDIGO :

E10

DEFINIÇÃO :

É o identificador do evento ocorrido.

ATRIBUTOS :

ch-manobra, proteção, medida, eqpto

IDENTIFICADORES :

ch-manobra, proteção, medida, eqpto

DESCRIÇÃO DE AGREGAÇÃO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

PV (parâmetro valor)

DEFINIÇÃO :

Agrega tipo termo técnico a ocorrência de termo técnico.

ENTIDADES E RELACIONAMENTOS PARTICIPANTES :

Tipo termo técnico

Ocorrência de termo técnico

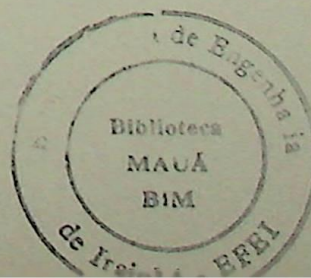
Topologia

ATRIBUTOS :

proteção, ch-manobra, medida, alimentador, circuito, nro relé, nro chave.

IDENTIFICADORES :

proteção, ch-manobra, medida, alimentador, circuito.



DESCRIÇÃO DE ATRIBUTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Adj

DEFINIÇÃO :

Se refere as palavras que formam o adjetivo

VINCULADOS A(S) ENTIDADE(S) :

VINCULADOS AO(S) RELACIONAMENTOS :

DOMÍNIO (Formato, Variação) :

String

ESTRUTURA :

Se agregado, quais os atributos agregados ?

☒ SIMPLES

☐ AGREGADO

OCORRÊNCIA :

☒ MONOVALORADO

☐ MULTIVALORADO

DESCRIÇÃO DE ATRIBUTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Artigo

DEFINIÇÃO :

Se refere as palavras que indicam o artigo.

VINCULADOS A(S) ENTIDADE(S) :

VINCULADOS AO(S) RELACIONAMENTOS :

DOMÍNIO (Formato, Variação) :

String

ESTRUTURA :

Se agregado, quais os atributos agregados ?

☒ SIMPLES

☐ AGREGADO

OCORRÊNCIA :

☒ MONOVALORADO

☐ MULTIVALORADO

DESCRIÇÃO DE ATRIBUTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Verbo

DEFINIÇÃO :

Se refere as palavras que compõem o verbo

VINCULADOS A(S) ENTIDADE(S) :

VINCULADOS AO(S) RELACIONAMENTOS :

DOMÍNIO (Formato, Variação) :

String

ESTRUTURA :

Se agregado, quais os atributos agregados ?

☒ SIMPLES

☐ AGREGADO

OCORRÊNCIA :



MONOVALORADO



MULTIVALORADO

DESCRIÇÃO DE ATRIBUTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Tempo-Modo

DEFINIÇÃO :

Se refere ao tempo e o modo do verbo.

VINCULADOS A(S) ENTIDADE(S) :

VINCULADOS AO(S) RELACIONAMENTOS :

DOMÍNIO (Formato, Variação) :

lista [string]

ESTRUTURA :

Se agregado, quais os atributos agregados ?

☒ SIMPLES

☐ AGREGADO

OCORRÊNCIA :



MONOVALORADO



MULTIVALORADO

DESCRIÇÃO DE ATRIBUTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Plural

DEFINIÇÃO :

Se refere a classificação do plural das palavras

VINCULADOS A(S) ENTIDADE(S) :

VINCULADOS AO(S) RELACIONAMENTOS :

DOMÍNIO (Formato, Variação) :

String

ESTRUTURA :

Se agregado, quais os atributos agregados ?

☒ SIMPLES

☐ AGREGADO

OCORRÊNCIA :

☒ MONOVALORADO

☐ MULTIVALORADO



DESCRIÇÃO DE ATRIBUTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Nro relé

DEFINIÇÃO :

Identifica um determinado relé em um ramo do circuito.

VINCULADOS A(S) ENTIDADE(S) :

Adjs

VINCULADOS AO(S) RELACIONAMENTOS :

Componentes

DOMÍNIO (Formato, Variação) :

Alfanumérico

ESTRUTURA :

Se agregado, quais os atributos agregados ?



SIMPLES



AGREGADO

OCORRÊNCIA :



MONOVALORADO



MULTIVALORADO

DESCRIÇÃO DE ATRIBUTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coordenador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :
 Chave-manobra

DEFINIÇÃO :
 Se refere as palavras que indicam uma chave dentro do sistema

VINCULADOS A(S) ENTIDADE(S) :
 Substantivo
 Adjs

VINCULADOS AO(S) RELACIONAMENTOS :
 Tipo-componente
 Componente

DOMÍNIO (Formato, Variação) :
 String

RESTRIÇÃO DE ACESSO :
 Disjuntor
 Chave

ESTRUTURA : ☒ SIMPLES ☐ AGREGADO	Se agregado, quais os atributos agregados ?
--	--

OCORRÊNCIA : ☒ MONOVALORADO ☐ MULTIVALORADO

DESCRIÇÃO DE ATRIBUTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coordenador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Conjunção

DEFINIÇÃO :

Indicativo da conjunção do verbo.

VINCULADOS A(S) ENTIDADE(S) :

Verbos

Tempo-Modo

VINCULADOS AO(S) RELACIONAMENTOS :

Conjunção

DOMÍNIO (Formato, Variação) :

ESTRUTURA :

☒ SIMPLES

☐ AGREGADO

Se agregado, quais os atributos agregados ?

OCORRÊNCIA :

☒ MONOVALORADO

☐ MULTIVALORADO

DESCRIÇÃO DE ATRIBUTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Radical

DEFINIÇÃO :

Corresponde ao radical de um verbo.

VINCULADOS A(S) ENTIDADE(S) :

Verbos

VINCULADOS AO(S) RELACIONAMENTOS :

Conjunção

DOMÍNIO (Formato, Variação) :

String

ESTRUTURA :

Se agregado, quais os atributos agregados ?

☒ SIMPLES

☐ AGREGADO

OCORRÊNCIA :

☒ MONOVALORADO

☐ MULTIVALORADO

DESCRIÇÃO DE ATRIBUTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coordenador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Nro Chave

DEFINIÇÃO :

Identifica o tipo de chave no circuito.

VINCULADOS A(S) ENTIDADE(S) :

Chaves

VINCULADOS AO(S) RELACIONAMENTOS :

DOMÍNIO (Formato, Variação) :

Númerico

ESTRUTURA :

Se agregado, quais os atributos agregados ?

☒ SIMPLES

☐ AGREGADO

OCORRÊNCIA :

☒ MONOVALORADO

☐ MULTIVALORADO

DESCRIÇÃO DE ATRIBUTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Medida

DEFINIÇÃO :

Se refere as palavras que indicam uma medida dentro do sistema.

VINCULADOS A(S) ENTIDADE(S) :

Substantivos

VINCULADOS AO(S) RELACIONAMENTOS :

Tipo-componente

DOMÍNIO (Formato, Variação) :

String

RESTRIÇÕES DE ACESSO :

Sobretensão, subtensão, sobrecorrente, subtensão, medida, etc

ESTRUTURA :



SIMPLES



AGREGADO

Se agregado, quais os atributos agregados ?

OCORRÊNCIA :



MONOVALORADO



MULTIVALORADO

DESCRIÇÃO DE ATRIBUTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coordenador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Substantivo

DEFINIÇÃO :

Se refere as palavras que indicam o substantivo

VINCULADOS A(S) ENTIDADE(S) :

VINCULADOS AO(S) RELACIONAMENTOS :

DOMÍNIO (Formato, Variação) :

String

ESTRUTURA :

Se agregado, quais os atributos agregados ?



SIMPLES



AGREGADO

OCORRÊNCIA :



MONOVALORADO



MULTIVALORADO

DESCRIÇÃO DE ATRIBUTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Sufixo

DEFINIÇÃO :

Se referencia a terminação das palavras

VINCULADOS A(S) ENTIDADE(S) :

VINCULADOS AO(S) RELACIONAMENTOS :

DOMÍNIO (Formato, Variação) :

String

ESTRUTURA :

Se agregado, quais os atributos agregados ?

☒ SIMPLES

☐ AGREGADO

OCORRÊNCIA :



MONOVALORADO



MULTIVALORADO

DESCRIÇÃO DE ATRIBUTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Proteção

DEFINIÇÃO :

Se refere as palavras que indicam uma proteção para o sistema

VINCULADOS A(S) ENTIDADE(S) :

Substantivos

VINCULADOS AO(S) RELACIONAMENTOS :

Tipo componente

DOMÍNIO (Formato, Variação) :

String

RESTRIÇÃO DE ACESSO :

Proteção, relé, relé de tempo, relé de temperatura, relé de distância, etc

ESTRUTURA :

☒ SIMPLES

☐ AGREGADO

Se agregado, quais os atributos agregados ?

OCORRÊNCIA :



MONOVALORADO



MULTIVALORADO

DESCRIÇÃO DE ATRIBUTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Eqpto

DEFINIÇÃO :

Se refere as palavras que indicam um equipamento dentro do sistema.

VINCULADOS A(S) ENTIDADE(S) :

Substantivos

Adjs

VINCULADOS AO(S) RELACIONAMENTOS :

Tipo-componente

Componente

DOMÍNIO (Formato, Variação) :

String

ESTRUTURA :

Se agregado, quais os atributos agregados ?

☒ SIMPLES

☐ AGREGADO

OCORRÊNCIA :

☒ MONOVALORADO

☐ MULTIVALORADO

DESCRIÇÃO DE ATRIBUTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Circuito

DEFINIÇÃO :

Identifica um determinado ramo do circuito.

VINCULADOS A(S) ENTIDADE(S) :

Adjs
 Medidas - E8 e E14
 Eqptos - E9 e E13
 Proteção - E11 e E16
 Chaves
 Ch-Manobra

VINCULADOS AO(S) RELACIONAMENTOS :

Componentes

DOMÍNIO (Formato, Variação) :

Circuito = N, Classe

onde : N - alfanumérico

Classe - alfabético

ESTRUTURA :

☒ SIMPLES

☐ AGREGADO

Se agregado, quais os atributos agregados ?

OCORRÊNCIA :



MONOVALORADO



MULTIVALORADO

DESCRIÇÃO DE ATRIBUTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Componente

DEFINIÇÃO :

Identifica o tipo de componente de um eqpto do circuito.

VINCULADOS A(S) ENTIDADE(S) :

Eqptos - E9

Eqptos - E13

VINCULADOS AO(S) RELACIONAMENTOS :

DOMÍNIO (Formato, Variação) :

String

ESTRUTURA :

☐ SIMPLES

☒ AGREGADO

Se agregado, quais os atributos agregados ?

transformador, alimentador, linha, banco capacitor

OCORRÊNCIA :



MONOVALORADO



MULTIVALORADO

DESCRIÇÃO DE HIERARQUIA

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :
 Morfologia

CÓDIGO :
 H01

DEFINIÇÃO :
 Se referencia a morfologia das palavras.

ENTIDADE DE MAIS ALTO NÍVEL :
 Morfologia Palavras

ENTIDADES DE MAIS BAIXO NÍVEL :
 Artigos, Substantivos, Adjs, Verbos

ATRIBUTOS QUE INDUZ(EM) A PARTIÇÃO :
 Tipos

HIERARQUIA :

☐ SUBCONJUNTO

☒ GENERALIZAÇÃO/ESPECIALIZAÇÃO

☒ TOTAL

☐ PARCIAL

DESCRIÇÃO DE HIERARQUIA

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Eventos

CÓDIGO :

H02

DEFINIÇÃO :

Se referencia a ocorrência de falta na subestação

ENTIDADE DE MAIS ALTO NÍVEL :

Tipo Evento

ENTIDADES DE MAIS BAIXO NÍVEL :

ATRIBUTOS QUE INDUZ(EM) A PARTIÇÃO :

HIERARQUIA :

☐

SUBCONJUNTO

☒

GENERALIZAÇÃO/ESPECIALIZAÇÃO

☒

TOTAL

☐

PARCIAL

DESCRIÇÃO DE HIERARQUIA

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :
 Topologia

CÓDIGO :
 H03

DEFINIÇÃO :
 Componentes da Subestação

ENTIDADE DE MAIS ALTO NÍVEL :
 Tipo termo técnico
 Ocorrência termo técnico

ENTIDADES DE MAIS BAIXO NÍVEL :
 PV (parâmetro valor)

ATRIBUTOS QUE INDUZ(EM) A PARTIÇÃO :

HIERARQUIA :

☐ SUBCONJUNTO

☒ GENERALIZAÇÃO/ESPECIALIZAÇÃO

☐ TOTAL

☒ PARCIAL

DESCRIÇÃO DE HIERARQUIA

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Regras

CÓDIGO :

H04

DEFINIÇÃO :

Constituído de todas as regras da língua.

ENTIDADE DE MAIS ALTO NÍVEL :

Regras da língua

ENTIDADES DE MAIS BAIXO NÍVEL :

ATRIBUTOS QUE INDUZ(EM) A PARTIÇÃO :

HIERARQUIA :

☐

SUBCONJUNTO

☒

GENERALIZAÇÃO/ESPECIALIZAÇÃO

☒

TOTAL

☐

PARCIAL

DESCRIÇÃO DE RELACIONAMENTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Termos Morfologia

CÓDIGO :

R01

DEFINIÇÃO :

Associa termos da língua a Morfologia Palavras

ATRIBUTOS :

artigos, substantivos, adjetivos, verbos

DIMENSÃO :	ENTIDADES PARTICIPANTES	PAPEL	CONNECTIVIDADE
	Termos da língua	possuído	1
	Morfologia Palavras	possui	N

DESCRIÇÃO DE RELACIONAMENTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :
 Regra Morfológica

CÓDIGO :
 R02

DEFINIÇÃO :
 Associa Regras da Língua a Morfologia Palavras.

ATRIBUTOS :
 tempo-modo, plural, sufixo

DIMENSÃO :	ENTIDADES PARTICIPANTES	PAPEL	CONNECTIVIDADE
	Regras da Língua	possuído	N
	Morfologia Palavras	possui	N

DESCRIÇÃO DE RELACIONAMENTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :
 Substantivo tp-modo

CÓDIGO :
 R03

DEFINIÇÃO :
 Associa o substantivo ao tipo termo técnico.

ATRIBUTOS :
 proteção, ch-manobra, ,medida, eqpto

DIMENSÃO :	ENTIDADES PARTICIPANTES	PAPEL	CONNECTIVIDADES
	Substantivos	possui	N
	Tipo termo técnico	possuído	1

DESCRIÇÃO DE RELACIONAMENTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :
 Adj Ident-Comp

CÓDIGO :
 R04

DEFINIÇÃO :
 Associa adjetivos a ocorrência de termo técnico.

ATRIBUTOS :
 proteção, chaves, eqptos, medidas

DIMENSÃO :	ENTIDADES PARTICIPANTES	PAPEL	CONECTIVIDADE
	Adjs	possui	N
	Ocorrência de termo técnico	possuído	1

DESCRIÇÃO DE RELACIONAMENTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Evento-Ocorrido

CÓDIGO :

R05

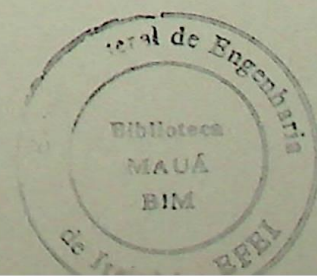
DEFINIÇÃO :

Associa ocorrência de termo técnico a tipo evento.

ATRIBUTOS :

circuito, nro relé, adj

DIMENSÃO :	ENTIDADES PARTICIPANTES	PAPEL	CONECTIVIDADE
	Ocorrência de termo técnico	possuído	1
	Tipo evento	possui	1



DESCRIÇÃO DE RELACIONAMENTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :
 Topologia

CÓDIGO :
 R06

DEFINIÇÃO :
 Associa PV a PV

ATRIBUTOS :
 proteção, ch-manobra, medida, alimentador, circuito, nro-relé.

DIMENSÃO :	ENTIDADES PARTICIPANTES	PAPEL	CONECTIVIDADE
	PV	possui	1
	PV	possuido	1

DESCRIÇÃO DE RELACIONAMENTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :
 Associações

CÓDIGO :
 R07

DEFINIÇÃO :
 Associa tipo termo técnico a tipo evento.

ATRIBUTOS :
 proteção, ch-manobra, medida, alimentador

DIMENSÃO :	ENTIDADES PARTICIPANTES	PAPEL	CONECTIVIDADE
	Tipo termo técnico	possuído	N
	Tipo evento	possui	N

DESCRIÇÃO DE RELACIONAMENTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :
 Associações

CÓDIGO :
 R08

DEFINIÇÃO :
 Associa verbos a tipo evento.

ATRIBUTOS :
 verbos

DIMENSÃO :	ENTIDADES PARTICIPANTES	PAPEL	CONNECTIVIDADE
	Verbos	possuído	N
	Tipo evento	possui	N

DESCRIÇÃO DE RELACIONAMENTO

PROJETO : Interface Homem-Máquina em Linguagem Natural

EQUIPE : Orientador : Prof. Germano Lambert Torres
 Coorientador : Prof. João Bosco Schumann Cunha
 Mestranda : Maria Silvana Peres
 Colaborador : Renato Ferreira Fernandes Júnior

NOME :

Correlação de Eventos

CÓDIGO :

R09

DEFINIÇÃO :

Associa tipo evento a tipo evento.

ATRIBUTOS :

ch-manobra, proteção, medida, eqpto, verbo

DIMENSÃO :	ENTIDADES PARTICIPANTES	PAPEL	CONECTIVIDADE
	Tipo evento	possuído	N
	Tipo evento	possui	N

ANEXO II

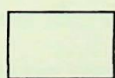
DIAGRAMA DE FLUXO DE DADOS

1.1 - INTRODUÇÃO

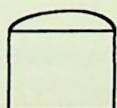
O D.F.D. é um modelo lógico do Sistema. O modelo não depende do hardware, do software, da estrutura de dados, ou da organização do arquivo. Como o diagrama é um quadro gráfico do sistema lógico, ele tende a ser de fácil entendimento mesmo para usuários não técnicos, e assim serve como excelente ferramenta de comunicação. O D.F.D. é um bom ponto de partida para o projeto de sistema.

1.2 - CONSTRUÇÃO DO DFD

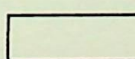
Símbolos básicos :



Origem ou destino dos dados :
representa a Entidade Externa.



Processo que transforma os dados



Depósito de Dados



Fluxo de Dados

OBS : . o processo não é necessariamente um programa;

. o depósito de dados não é o mesmo que um arquivo;

O D.F.D. é muito útil como ferramenta de comunicação. Se um D.F.D. contivesse dezenas de processos e depósitos de dados, este seria difícil de seguir, prejudicando a comunicação.

1.3 - UTILIZAÇÃO DO D.F.D.

O D.F.D. ajuda na organização de informações, a resumir informações, extrair detalhes_chaves, e considerar as relações entre esses detalhes.

O D.F.D. completo pode ser utilizado para explicar a compreensão de um problema, pode ser utilizado como auxílio a um projeto, no estágio de projeto detalhado e para testar o projeto físico inicial de um sistema.

Com a definição de cada elemento de dados no sistema, exploramos o que está ocorrendo dentro dos processos.

A medida que montamos o D.F.D., identificamos locais onde os dados foram mantidos de uma transação para a seguinte ou armazenamento permanente, porque descreviam um aspecto do mundo fora do Sistema.

Deve-se especificar os elementos de dados contidos em cada depósito de dados de forma semelhante a especificação de cada fluxo de dados. Como os dados só podem entrar em um depósito através de um fluxo e não podem sair a não ser que tenham entrado, o conteúdo de um depósito de dados pode ser lido a partir da especificação dos fluxos de dados que entram e saem. O conteúdo lógico de cada depósito de dados é mantido no dicionário de dados sob o nome de depósito de dados.

1.4 - COMO PROJETAR UM D.F.D.

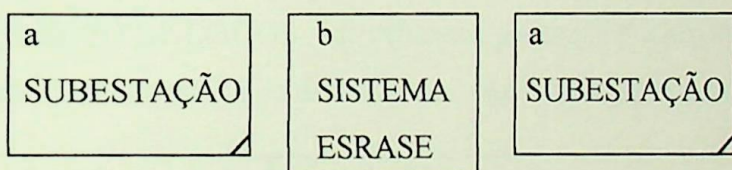
O D.F.D. de nível lógico é a ferramenta principal para entendimento e manipulação de um sistema de qualquer complexidade juntamente com o refinamento dessa notação para uso em análise.

1.4.1 - Convenções Simbólicas

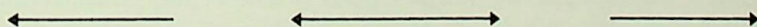
a) ENTIDADES EXTERNAS : São categorias lógicas de coisas ou pessoas que representam uma fonte ou destino para transação.

Como referência à entidade, esta pode ser identificada por uma letra minúscula no canto superior esquerdo. Para evitar cruzamento de linhas de fluxo de dados, a mesma entidade pode ser desenhada mais de uma vez no mesmo diagrama.

Quando tivermos de duplicar outra entidade, marcamos suas ocorrências com linhas inclinadas. Exemplo :



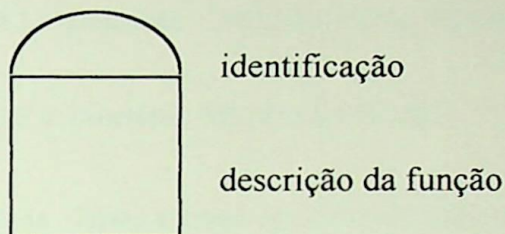
b) FLUXO DE DADOS : é simbolizado por meio de uma seta, de preferência horizontal e/ou vertical, com a ponta indicando a direção do fluxo.



Cada fluxo pode ser considerado como um tubo por onde passam pacotes de dados identificando os processos, entidades ou depósitos de dados, anotando-se uma descrição do conteúdo de cada fluxo ao longo de sua extensão.

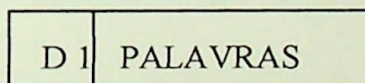
c) PROCESSO : é necessário descrever a função de cada processo e para facilitar a referência, fornecer uma identificação única para cada um.

A identificação pode ser um número. A partir do momento em que o processo recebe uma identificação, esta não deve ser modificada, exceto para desmembramentos ou agrupamentos, pois serve como referência para os fluxos de dados e para a decomposição do processo em níveis inferiores.

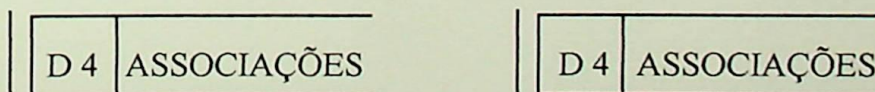


A descrição da função deve consistir num verbo ativo seguido de uma cláusula objeto. O verbo deve ser forte de modo a não deixar dúvidas.

d) DEPÓSITO DE DADOS : Identificados por um "D" e um número arbitrário, a esquerda. O nome deve ser escolhido de modo que seja bem significativo para o usuário. Exemplo :



Os depósitos de dados que são desenhados mais de uma vez no mesmo D.F.D., são identificados por linhas verticais adicionais à esquerda. Exemplo :



Quando um processo armazena dados, a seta do fluxo de dados aponta para o depósito de dados; quando o acesso a um depósito de dados é feito de forma a realizarmos

apenas leitura, é suficiente mostrarmos o grupo de elementos de dados recuperados, no fluxo de dados que está saindo.

Formulários que compõem o dicionário de dados :

| I | n | t | e | r | f | a | c | e | | E | S | R | A | S | E |

FLUXO DE DADOS

Origem_ref. : D 3 Descrição: EVENTOS OCORRIDOS

Destino_ref.: E.E.a Descrição: SISTEMA ESRASE

Descrição ampliada : Após a frase ser classificada e analisada, é gerado um conjunto de eventos que para o Sistema ESRASE representa o “cenário”, e, a partir daí, o ESRASE estará pronto para atuar.

| E | v | e | n | t | o | s | | e | m | | L | N |

FLUXO DE DADOS

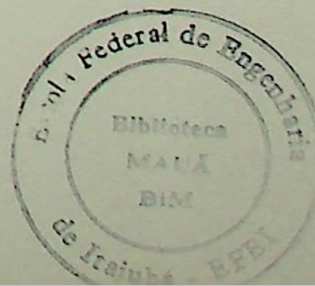
Origem_ref. : E.E.a Descrição: SUBESTAÇÃO

Destino_ref. : P 1 Descrição: Captar Sentenças em Linguagem Natural

Descrição ampliada : Corresponde a uma string de N caracteres, referente a algum acontecimento de falha na subestação, digitada pelo usuário para ser analisada.

Estruturas de Dados incluídas :

Frase



|C|o|n|j|u|n|t|o| |d|e| |T|o|k|e|n|s|

FLUXO DE DADOS

Origem_ref. : P 1 Descrição: Captar Sentenças em Linguagem Natural

Destino_ref. : P 2 Descrição: Análise Morfológica e Sintática

Descrição ampliada : Após ser elaborado uma frase pelo usuário, ela é dividida em partes, que são denominados Tokens, referentes a frase.

Estruturas de Dados incluídas :

Tokens

|S|e|n|t|e|n|ç|a| |e|m| |L|. |N|. |

FLUXO DE DADOS

Origem_ref. : P 2 Descrição: Análise Morfológica e Sintática

Destino_ref. : P 3 Descrição: Interpretação Significativa

Descrição ampliada : Formada por uma lista (em Prolog) devidamente classificada morfolologicamente e agrupada sintaticamente em grupos maiores.

Estruturas de Dados incluídas :

s-nominal-s ou

s-nominal-j ou

s-nominal-asj ou

s-nominal-as ou

s-nominal-sj ou

s-nominal-js

Formulário para registrar Fluxo de Dados :

|T|o|p|o|l|o|g|i|a| |d|a| |S|u|b|e|s|t|a|ç|ã|o| FLUXO DE DADOS

Origem_ref. : D 1 Descrição : TOPOLOGIA

Destino_ref. : P 3 Descrição : Interpretação Significativa

Descrição ampliada : Transporta os dados de TOPOLOGIA para que seja feita a verificação de que os componentes classificados fazem parte do D.B. do Sistema da Subestação.

Estruturas de Dados incluídas :

proteção ou
chave de manobra ou
equipamento ou
medida

| A | s | s | o | c | i | a | ç | ã | e | s |

FLUXO DE DADOS

Origem_ref. : D 4 Descrição : ASSOCIAÇÕES

Destino_ref. : P 3 Descrição : Interpretação Significativa

Descrição ampliada : Transporta o substantivo com o flag indicativo do respectivo evento para que seja feito a associação entre o substantivo, entrada via linguagem natural, com o evento.

Transporta o número do relé à partir do nome, ou, vice-versa.

Estruturas de Dados incluídas :

Associa ou
Associa-relé

| A | s | s | o | c | i | a | ç | ã | e | s |

FLUXO DE DADOS

Origem_ref. : D 4 Descrição : ASSOCIAÇÕES

Destino_ref. : P 4 Descrição : Correlação de Eventos

Descrição ampliada : Através do evento-medida, especifica o respectivo relé referente a esta linha, determinada pelo mesmo.

Estruturas de Dados incluídas :

Associa-estado-relé

| P | a | l | a | v | r | a | s |

FLUXO DE DADOS

Origem_ref.: D 2 Descrição: PALAVRAS

Destino_ref : P 2 Descrição : Análise Morfológica e Sintática

Descrição ampliada : Transporta um conjunto de palavras classificadas no D.B. para que seja feita uma associação com os Tokens classificados no processo P1, que formam a frase digitada pelo usuário.

Estruturas de Dados incluídas : Substantivo ou
Artigo ou
Adj ou
Radicais dos verbos ou
Tempo/modo dos verbos

|T|o|p|o|l|o|g|i|a| |d|a| |S|u|b|e|s|t|a|ç|ã|o|

FLUXO DE DADOS

Origem ref. : D 1 Descrição : TOPOLOGIA

Destino_ref : P 4 Descrição : Correlação de Eventos

Descrição ampliada : À partir do evento ocorrido o processo P4 busca informações referentes a relação do evento dentro da topologia.

Estruturas de Dados incluídas : não possui

|E|v|e|n|t|o| |O|c|o|r|r|i|d|o|

FLUXO DE DADOS

Origem_ref. : P 3 Descrição : Interpretação Significativa

Destino_ref. : D 3 Descrição : EVENTOS OCORRIDOS

Descrição ampliada : O evento escrito pelo usuário em Linguagem Natural, agora já está no formato do Sistema ESRASE.

Estruturas de Dados incluídas :

evento-proteção ou
evento-medida ou
evento-chave-manobra ou
evento-eqpto

|E|v|e|n|t|o| |O|c|o|r|r|i|d|o|

FLUXO DE DADOS

Origem_ref. : D 3 Descrição : EVENTOS OCORRIDOS

Destino_ref. : P 4 Descrição : Correlação de Eventos

Descrição ampliada : O evento escrito pelo usuário em Linguagem Natural, agora já está no formato do Sistema ESRASE.

Estruturas de Dados incluídas :

evento-proteção ou
evento-medida ou
evento-chave-manobra ou
evento-eqpto

| O | u | t | r | o | s | | E | v | e | n | t | o | s |

FLUXO DE DADOS

Origem_ref. : P 4 Descrição : Correlação de Eventos

Destino_ref. : D 3 Descrição : EVENTOS OCORRIDOS

Descrição ampliada : Acrescenta no D.B., Eventos Ocorridos, outros eventos que se relacionam a aqueles eventos ocorridos, como causa e efeitos.

Estruturas de Dados incluídas :

evento-proteção ou
evento-medida ou
evento-chave-manobra ou
evento-eqpto

| P | a | l | a | v | r | a | s |

FLUXO DE DADOS

Origem_ref. : P 5 Descrição : Editar D.B.

Destino_ref. : D 2 Descrição : PALAVRAS

Descrição ampliada : Transporta as palavras editadas pelo usuário para que sejam classificadas no D.B., e com isso sua respectiva associação com os Tokens

Estruturas de Dados incluídas :

Substantivo ou
Artigo ou
Adj ou
Radicais dos verbos ou
Tempo/modo dos verbos

Formulário para registrar Fluxo de Dados :

| A | s | s | o | c | i | a | ç | ã | o | e | s |

FLUXO DE DADOS

Origem_ref. : P 5 Descrição : Editar D.B.

Destino_ref. : D 4 Descrição : ASSOCIAÇÕES

Descrição ampliada : Transporta as palavras editadas pelo usuário para que sejam associadas ao flag indicativo.

Estruturas de Dados incluídas :

Associa

| P | a | l | a | v | r | a | s | | N | o | v | a | s | | e | | A | l | t | e | r | a | ç | ã | o | e | s |

FLUXO DE DADOS

Origem_ref. : EE.b Descrição : SISTEMA ESRASE

Destino_ref. : P 5 Descrição : Editar D.B.

Descrição ampliada : palavras digitadas pelos usuários para atualizações ou alterações do D.B.

Estruturas de Dados incluídas :

Tokens

Formulário para descrever depósito de dados :

| A | S | S | O | C | I | A | Ç | O | E | S |

DEPÓSITO DE DADOS

Ref. : D 4

Descrição : Armazena os dados correspondentes ao adj associa, referente à

Substantivo, Relé e Estado da linha.

Corresponde a agregação AG1 do DER do Sistema ESRASE.

Fluxos de Dados que entram :

- Associações (P 5 - D 4)

Fluxos de Dados que saem :

Conteúdo :

Associa : Substantivo (específico)

Associa : Relé - nome e número do relé (específico)

Associa : Estado - número do relé (específico)

| A | S | S | O | C | I | A | Ç | O | E | S |

DEPÓSITO DE DADOS

Ref. : D 4

Descrição : Armazena os dados correspondentes ao adj associa, referente à Substantivo, Relé e Estado da linha.

Corresponde a agregação AG1 do DER do Sistema ESRASE.

Fluxos de Dados que entram :

Fluxos de Dados que saem :

Associações (D4 - P 3)

Associações (D 4 - P 4)

Conteúdo :

Associa : Substantivo (específico)

Associa : Relé - nome e número do relé (específico)

Associa : Estado - número do relé (específico)

| T | O | P | O | L | O | G | I | A |

DEPOSITO DE DADOS

Ref. : D 1

Descrição : Armazena informações específicas da topologia de uma subestação.

Corresponde ao relacionamento R1 do DER do Sistema ESRASE.

Fluxos de Dados que entram :

Fluxos de Dados que saem :

Responsabilidade do Sistema
ESRASE

- Topologia da Subestação (D1 - P3)

- Topologia da Subestação (D1 - P4)

Conteúdo :

Conexão - circuitos e relações

Disjuntor - Sistema

Proteção - formado por circuito, número de relé (adj) e estado da chave.



|E|v|e|n|t|o|s| |O|c|o|r|r|i|d|o|s|

DEPOSITO DE DADOS

Ref. : D 3

Descrição : Armazena os adjs decorrentes da classificação das palavras e análise da frase escrita pelo usuário.

Corresponde a entidade E9 do DER do Sistema ESRASE.

Fluxos de Dados que entram :

- Evento Ocorrido (P3 - D3)
- Outros Eventos (P4 - D3)

Fluxos de Dados que saem :

- Evento Ocorrido (D3 - P4)
- Interface ESRASE (D3 - EE)

Conteúdo :

Adj : proteção
equipamento
chave de manobra
medida

|P|A|L|A|V|R|A|S|

DEPOSITO DE DADOS

Ref. : D 2

Descrição : Armazena dados correspondentes ao substantivo, adj, verbo e artigo, para que sejam classificadas as palavras na formação e análise das frases.

Corresponde a entidade E1 do DER do Sistema ESRASE.

Fluxos de Dados que entram :

Palavras (P5 - D2)

Fluxos de Dados que saem :

Palavras (D2 - P2)

Conteúdo :

Verbo, artigo, adj, substantivo (simples ou comoposto).

Adj : terminologia técnico (específico) e qualidade

Formulário para descrever processos :

C o r r e l a ç a o d e E v e n t o s		PROCESSO	Ref. : P 4
Descrição : À partir do evento descrito pelo usuário e através de uma análise do sistema (da subestação), determina-se as consequências referentes a este evento e tenta-se descobrir as causas que o ocasionaram.			
Entradas	Resumo Lógico	Saídas	
Associações (D4 - P4)	- Verifica quem está relacionado com o evento ocorrido	Outros Eventos (P4 - D3)	
Evento Ocorrido (D3 - P4)	- Efetiva a ocorrência destes eventos - Tenta descobrir a causa do evento ocorrido		
Topologia da Subestação (D1 - P4)			

Descrição : Pega a lista devidamente classificada e encontra (para o substantivo)

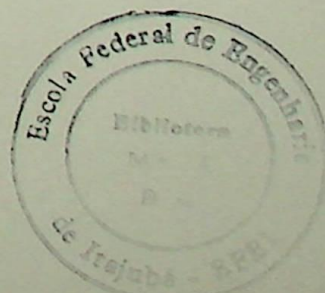
uma relação com um evento reconhecido pela máquina. À partir daí ele efetiva no

banco de dados o evento referente a frase escrita pelo usuário.

Entradas	Resumo Lógico	Saídas
- Sentença em L.N. (P2 - P3)	- Procura pelo substantivo e o relaciona com um evento conhecido	- Evento Ocorrido
- Topologia da Subestação (D1 - P3)	- Procura pelos outros elementos (adjs) que completam o evento. Se não encontrar, apresenta uma lista de possíveis componentes para a substituição	
- Associações (D4 - P3)	- Efetiva o evento no banco de dados	

Descrição : O Sistema analisa a frase digitada pelo usuário e divide-a em unidades menores ("Tokens"). Estes "Tokens" são classificados morfologicamente (ou ignorados). Após ser feita toda a classificação, passa-se a agrupar os "Tokens" em unidades maiores, classificando a frase sintaticamente.

Entradas	Resumo Lógico	Saídas
- Conjunto de Tokens (P1 - P2)	- Divisão da frase em "Tokens" - Classificação Morfológica	- Sentença em L.N. (P2 - P3)
- Palavras (D2 - P2)	- Classificação Sintática	



|C|a|p|t|a|r| |S|e|n|t|e|ç|a|s| |e|m| |L|i|n|g|u|a|g|e|m| |N|a|t|u|r|a|l|

PROCESSO

Ref. : P 1

Descrição : À partir de um evento ocorrido na subestação, o usuário descreve o evento para o computador utilizando-se do teclado, na forma da Linguagem Natural do usuário.

Entradas	Resumo Lógico	Saídas
- Eventos em L.N. (EEa - P1)	- Ocorre o evento na subestação - O usuário digita no computador a frase que descreve o evento - O computador “entende” (ou não) a frase descrita pelo usuário.	- Conjunto de Tokens (P1 - P2)

|E|d|i|t|a|r| |B|a|n|c|o| |d|e| |D|a|d|o|s|

PROCESSO

Ref. : P 5

Descrição : É constituído à partir de dados referentes a subestação, corresponde ao verbo, artigo, adj, substantivo e conjugação do verbo.

Entradas	Resumo Lógico	Saídas
- Palavras Novas e Alterações. (EEb - P5)	- O usuário poderá incluir novas palavras no D.B. - Após a inclusão de palavras, estas poderão ser deletadas e alteradas.	- Associações (P5 - D4) - Palavras (P5 - D2)

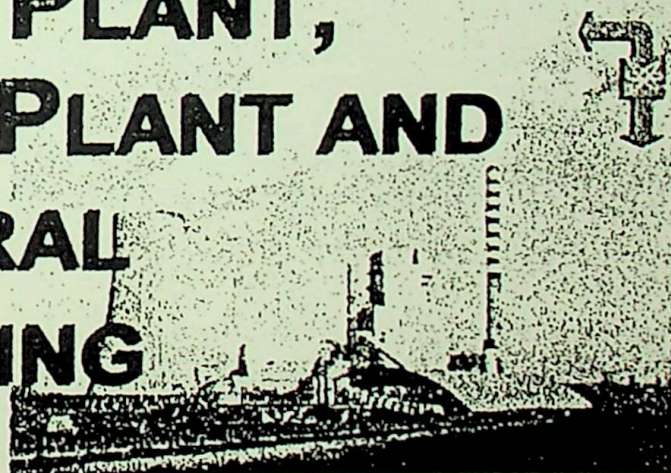
ANEXO III

SMiRT 13 Post Conference Seminar Nr. 13

Co-organizers:

N. F. de Almeida Neto, ELETROPAULO, Sao Paulo, Brazil
J. M. A. Anson, ELETROPAULO, Sao Paulo, Brazil
P. Bonissone, GEC Schenectady, USA
S. Fukuda, Tokyo MIT, Japan
S. Gehl, EPRI Palo Alto, USA
A. S. Jovanovic, MPA Stuttgart, Germany
A. C. Lucia, CEC JRC Ispra, Italy
S. Yoshimura, RACE, Univ. of Tokyo, Japan

APPLICATIONS OF INTELLIGENT SOFTWARE SYSTEMS IN POWER PLANT, PROCESS PLANT AND STRUCTURAL ENGINEERING



PROCEEDINGS

Volume II

A. S. Jovanovic (Editor)
MPA Stuttgart, August 1995

São Paulo, Brazil, August 21-23, 1995

Supported by:

CEC-JRC Ispra, Italy
ELETROPAULO, Sao Paulo, Brazil
EPRI, Palo Alto, USA
MPA Stuttgart, Germany

INTERFACE MAN/MACHINE FOR CONTROL CENTERS

Maria Silvana Peres¹
Germano Lambert Torres^{1,2}

Renato Ferreira Fernandes Jr.¹
João Bosco Schumann Cunha³

1. Escola Federal de Engenharia de Itajubá - Brazil

2. Universidade de Taubaté - Brazil

3. Instituto Nacional de Pesquisas Espaciais - Brazil

Summary : This article describes the Interface Man/Machine in common language that can be applied in leading the problem where the user employs Natural Language (N.L.) to describe an occurred event, and it has as main purpose to register the occurred event, as well as, a list of points which describes it we'll have the receiving of a sentence in common language, the Syntax and Morphological Analysis, the meaning interpretation, the Correlation of events and the Words Edition.

At first was developed a program to analyze the phrase structure, into smaller groups of tokens and after into larger groups, verifying the grammar that there was behind the text. The leading, which sets up the occurred event formed from adjective, verb, article and noun, provides events which form a list of points needed to the phrase interpretation, using, for it, a topology of the occurred event identification. It has been developed a program to interpretate the text already classified, and a program that, based on these information's could deduce other information, without being necessary the user intervention. Finally, all the modules have been joined, becoming a compact system with a friendly interface, turning totally transparent its use by the final user.

1 - INTRODUCTION

In the article show some advantages the user takes to use as expert system that can be applied to several different areas. To get an expert system is necessary to define what types of information will be necessary for knowledge base formation, so we be able to understand the occurred event and from this point, to define other events which have been produced from the occurred event. The objective of the interface Man\Machine in Natural Language (N.L.) is providing one more option to the final user to communicate with the machine[1][18].

An environment in Natural Language (N.L.), based on knowledge, allows the user to participate in the process of target-talk execution, explaining its intention. Using its daily life communication language. This form of adapted communication its the user own language, or, the Portuguese Language, in this case[4].

The result knowledge of the interactions with the user shall be learned by environment. This conception will make the environment capable to map a knowledge within the leading. The points are given by the user, and the system will require them in a "friendly" way explaining the need of same ones, based on stored knowledge.

The environment shall be used in several applications which require the procedures and points of information exchange. We may have an environment global view at the figure 1 [16]:

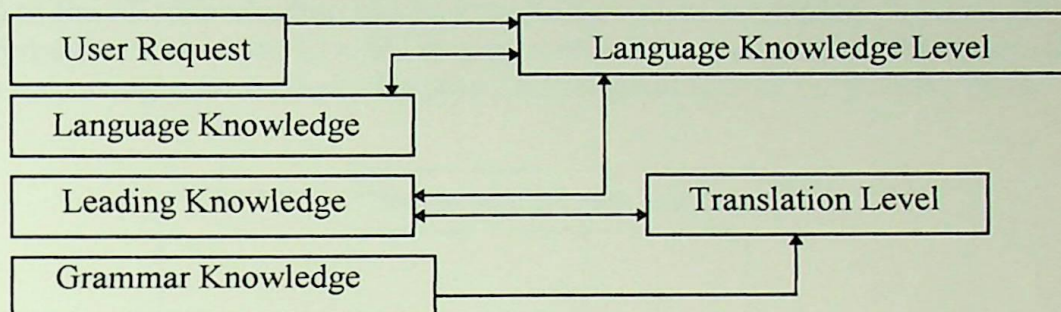


Figure 1 - Environment Global View.

We have used as an applicable example the Expert System for Substation Automatic Restoration (ESRASE), which has been developed as a helping tool in the operation of a substation, aiming a greater optimization of the functions, with the purpose of a greater in service reliability. The system is based on Expert System technology, that is, they're a system sub-group based on knowledge, since, they have a knowledge base very close to an expert one. It analyses the occurrence of certain defect at the substation, it establishes and carries out a Director Plan of Re-establishment Actions and the state variations of the substation components that are being held [6].

We're tried to assemble the main ideas of the interface since the text acquisition in an accessible language by the machine.

2 - ESRASE - EXPERT SYSTEM FOR SUBSTATION AUTOMATIC RESTORATION [3]

The basic characteristics of ESRASE are the speed, efficiency and reliability of restoration, considering the particulars and peculiarities of the intermediate responses of power substations and power systems to every one of the required switching operations. Its self-adaptability and its role as a decision-maker associated to the speed generated by automation, enable its efficiency and reliability to reach high levels.

ESRASE is driven, after the clearing of the fault, by the status of the switches, circuit breakers, protections and auxiliary relays and by the PS characteristics values shown both before and after the fault.

Language (N.L.). The user request are done in Natural Language (N.L.). It employs linguistic knowledge from Portuguese Language, so, it can be detected the user intention. This knowledge, given through rules, it'll be a standard for all applications, corresponding the words morphology.

It is important to emphasize that our objective it's not analyzing the phrase at the least details, but, interpreting the user intention. Thus we concern to do a simple knowledge base, however, not as much effective for our purpose. An example is presented below [13] :

- Rules of word plural:

TERMINATION	PLURAL
" a "	" as "
" o "	" os "
" r "	" res "
...	

- Rule of Verbs:

```
verb ( abrir, " ir " )
mode_tense ( " r ", " ir ", [ " ido ", " o ", " es ", " e ", " emos ", " is ", " em " ] )
```

4 - MORPHOLOGIC KNOWLEDGE

After being determined the leading which you want to work on, just match each leading element to the points that rule the system. There are in the system words (or language terms) that will be always present, it doesn't matter which expert system is for. These terms help the user to complete a fact, an action. There are words which are keys for the system and vary according to the expert system, these ones shall be classified correctly. And that's through these keys words the system will perform, through motor inference, to generate the occurred events, related by the user [2].

The morphology words are made from words classification like article, noun, adjective, and verb. An ADJ is an entity built by adjectives and other technical terms, named here occurency-type. An example is presented below:

a) For ESRASE System:

Substantive :	- protetion	Adj :	- 2K21-3	Verb:	- operate
	- switch		- 3K		- open
	- relay		- RD01		- close
			- opened		
			- closed		

b) For a Motor Control Center:

Substantive :	- motor	Adj :	- 21	Verb :	- increase
	- dc motor		- 23		- decrease
	- speed		- 22dc		- drop
	- voltage		- 2000		- start
	- frequency		- 1800		

5 - TECHNICAL KNOWLEDGE

Within a determined expert system there will be certain points which will be mentioned as keys in the system. These points are responsible for the component type and component identification that it'll develop the action [9].

For the system, the "component-type" it'll be always a noun, the "component" it'll be always an adjective and the "action" will be represented by the verb. There are determined expert system, where the verb does not have an important meaning in the event identification. In these cases, the identifier will be the noun.

Like for example, to determine the occurred fault events at a substation, where protection, relay, circuit breaker are the nouns, 2K4, 5K21-3 are the adjectives, the verb in such case will always mean a turn off action.

For example, in ESRASE Systems:

- Technical terms: - protection-event
 - switchnig-event
- Parameters: - circuit
 - relay
 - switch
- Values: - circuit - 1K
 - circuit - 6F
 - relay - 21
 - relay - 23

The figure 3 shows a generic system knowledge structure.

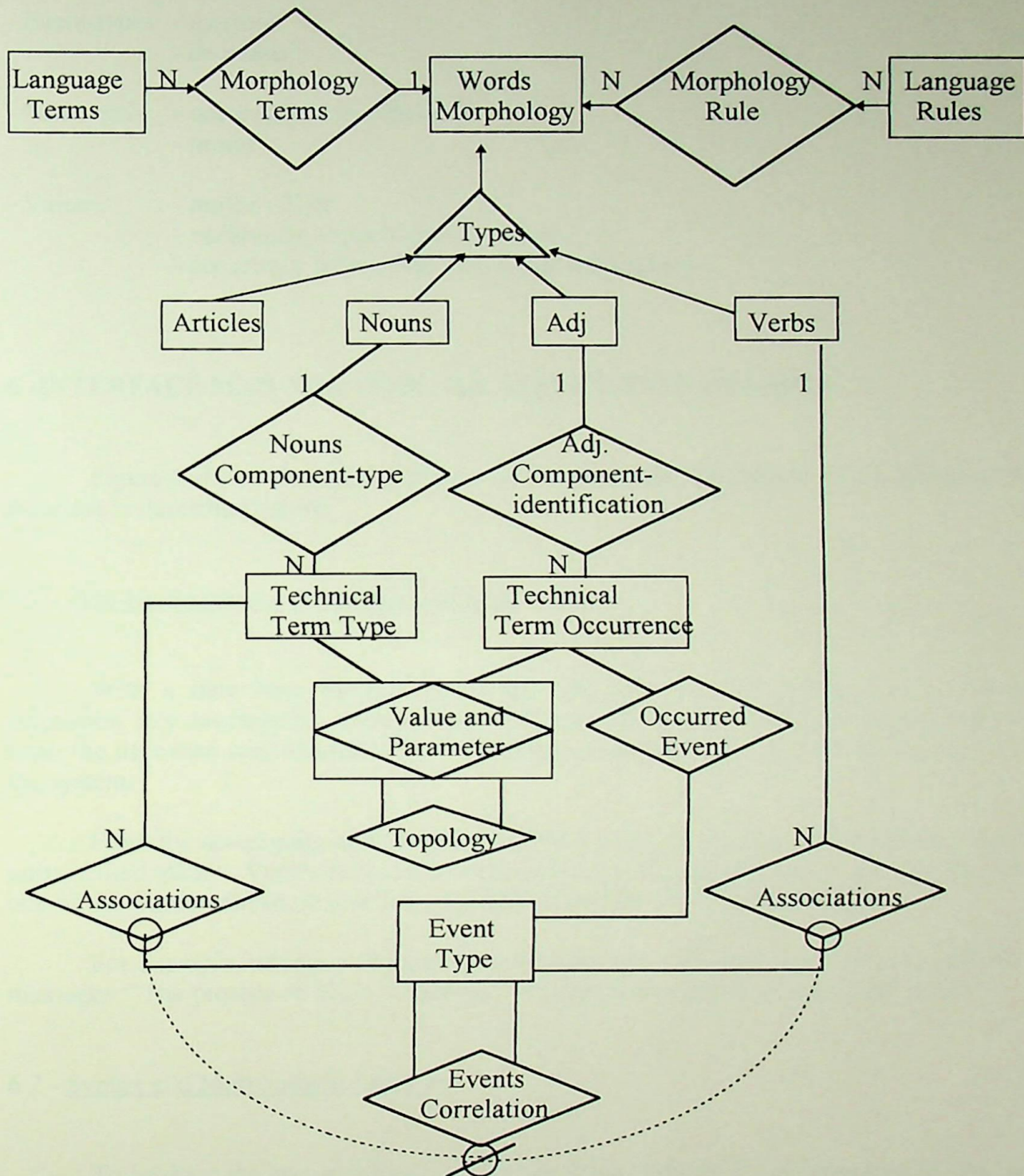


Figure 3 - Generic System Knowledge Structure.

Another possible message from the user is linking with the auxiliary service in a substation. For example: "The motor 21 speed increased." or "The armature current of motor 23 decreased to 3.5 (A).".

- Event-types: - increase
 - decrease
- Parameters: - occurrence_especification
 - motor
- Values: - motor - 21dc
 - occurrence_especification - speed
 - occurrence_especification - armature_current

6 -INTERFACE MAN-MACHINE - AN APPLICATION EXAMPLE

Figure 4 shows the main structures of the Interface Man/Machine. Each part of this structure is described below.

6.1 - Pick-Up Sentences in Natural Language

With a data base already assembled and the occurrence of an event from a substation, it's developed a sentence that describes the event in Natural Language (N.L.), from the user own way of expressing. The phrase shall refer to a certain event occurred in the system.

From the developing and the printing of the same, the phrase is divided into smaller units, called tokens. Verify each word consistency, or in other words, try to associate each one of the words with some one that might be within the ESRASE data base.

For example, after a contingency in a substation, the user can write the following messages: "The protection 3K21-3 opened." or "The feeder RD01 of circuit 1F is off."

6.2 - Syntax and Morphologic Analysis

To be done the phrase syntax and morphologic analysis, the system uses a data base called "WORDS", which correspond to a group of words and expressions that serve to describe a trouble within a specific system, Portuguese language terms and technical terms called facts base.

6.2.1 Morphologic Analysis

This function has as its main purpose to identify morphologically each phrase word, generating a list with all the words and their respective identifications. When the system receives the sentence, it saves this sentence and starts the process of division and classification.

- Sentence: "O relé de tempo do circuito 1F atuou."

In English: "The time relay of circuit 1F operated."

- Morphologic Analysis :

```
lista_morf = [ article (" O "), subst (" relé de tempo "), subst (" circuito "),  
              adj (" 1F "), verb (" atuou ") ]
```

6.2.2 Syntax Analysis

This function has as its main purpose to deduce a grammatical structure where the phrase can match. This process is done by the obtained list union from smaller blocks, identifying their syntax functions.

The following step would be the meaning interpretation of the phrase already being analyzed, where had been formed a morphologic classified list and syntaxily grouped in larger groups [11].

- Sentence: " O relé de tempo do circuito 1F atuou."

- Syntax Analysis :

```
sentence = [ s_nominal_as ( article (" O "), subst (" relé de tempo ") ),  
            s_nominal_sj ( subst (" circuito "), adj (" 1F ") ), verb (" atuou ") ].
```

6.3- Meaning Interpretation

The program uses a previous established knowledge base where through rules and facts determines the text terms which are important to these terms is associated an event from the system, and with the text adding information completes the fact.

Using the setence above, the program produces the follow statment:

```
protection_event (" 1 ", " F ", " 62 " ).
```


6.4 - Events Correlation

From the occurred event described by the user, already morphologic and syntaxily analyzed, it verifies its relations, its occurrences, it determines the referred consequences to this event and tries to discover the causes which has caused it.

After the data base occurrence has been set, the information passed to the user in Natural Language (N.L.) is changed into Prolog Language.

Right after the phrase being classified and analyzed, is generate a group of events. For exemple [11] :

- Events Correlation : switchching_event (" 1 ", " F ", " 4 ")

eqpt_event (" 1 ", " F ", alimentador (" RD01 "))

measure_event (" 1 ", " F ", 0, 0, 0, 0)

6.5 - Edit D.B.

It's made by words, formed by :

- Portuguese Language Terms : made by verbs, article, nouns and adjectives.
- Technical Terms : made by nouns and adjective.

As the system uses a specific data base, these data can be renew steadily, or changing been processed, or even deleted. It's made from printed data base and specific for an application . These data correspond to verb, article, adjective,noun.

Associated to this process are the "D.B. EVENTS TYPES", which store the edited words by the user and will be associated to an indication flag so they can be associated and classified to "tokens".

We'll may verify the Man/Machine Interface Communication into Natural Language "N.L." through the figure which is followed to the conclusion.

7 - ILLUSTRATIVE EXAMPLE :

- Occurency:

"The instant overcurrent relay of lines 3 and 12 from the low-voltage circuit opened."

- Sentence: "Protections 3F50 e 12F50 operated."
- Analysis: sentence = [s_nominal_sj (subst (" Protections "), adj (" 3F50 "),
s_nominal_s (" 12F50 "), verb (" atuaram ")].
- Events :
 - protection_event (" 3 ", " F ", " 50 ")
 - protection_event (" 12 ", " F ", " 50 ")
 - switching_event (" 3 ", " F ", " 4 ")
 - switching_event (" 12 ", " F ", " 4 ")
 - eqpt_event (" 3 ", " F ", alimentador (" RD03 "))
 - measure_event (" 3 ", " F ", 0, 0, 0, 0)
 - measure_event (" 12 ", " F ", 0, 0, 0, 0)

8 - REFERÊNCIAS BIBLIOGRÁFICAS :

- [1] Weizenbaum, J., "ELIZA - A Computer Program for the Study of Natural Language Communication between Man and Machine", Communication of the ACM, Vol. 9, Número 1, Jan.1966.
- [2] Savadovsky, P. - "Introdução ao Projeto de Interfaces em Linguagem Natural", Projeto Estra - SID Informática SA.
- [3] Ribeiro, G.M. and Lambert-Torres, G., "ESRASE - Expert System for Automatic Restoration of Substations",. III Symposium of Specialists of Electric Operational and Expansion Planning, Belo Horizonte, 1992.
- [4] Próspero Siqueira, I.S. and Velloso, H.M., "Interação Homem/Máquina diante de Interface em Linguagem Natural com Português Escrito", UNICAMP.
- [5] Date, C. J., "Introdução a Sistemas de Bancos de Dados", Editora Campus Ltda, 1991.
- [6] Lambert-Torres, G., Valiquette, B., Ribeiro, G.M., and Do, X.D., "ProControl - A Hybrid Expert System for Power System Protection and Control", II International Symposium on Expert System Applications in Power Systems (ESAP), Melbourne, Austrália, 1993.
- [7] Gane, C. and Sarson, T., "Análise Estruturada de Sistemas", Livros Técnicos Científicos e Editora Ltda, 1983.
- [8] Lambert-Torres, G., "Inteligência Artificial", Escola Federal de Engenharia de Itajubá (EFEI), 1992.
- [9] Harmon, P. and King, D., - "Expert Systems - Artificial Intelligence in Business", New York : John Wiley & Sons Inc., 1985.
- [10] Nebendahl, D., "Expert Systems, Introduction to the Technology and Applications", Siemens, 1988
- [11] Li, D., "A PROLOG Database System", Heriot-Watt University, Edinburgh, UK, 1984.
- [12] Stefik, M., Aikins, J., Balzer, R., *et alii.*, "The Organization of Expert Systems, A Tutorial", Artificial Intelligence, North-Holland, 1982.
- [13] Levine, R.I., Drang, D.E. and Edelson, B., "A Comprehensive Guide to AI and Expert Systems", McGraw Hill, 1986.

- [14] Hayes-Roth, F., Waterman, D.A. and Lenat, D.B., "Building Expert Systems", Addison-Wesley Publishing Company, 1983.
- [15] Kowalski, R., "Logic for Data Description", Logic and Database, Plenum Press, 1978.
- [16] Bittencourt, G., "JAI I - Tópicos em Representação de Conhecimento", INPE - São José dos Campos - S.P.
- [17] Brachman, R.J., Fikes, R.E. and Levesque, H.J., "Krypton : A Functional Approach to Knowledge Representation", IEEE, October 1983.
- [18] Rumpel, D. and Krost, G., "Natural Language Interface and Database Issues in Applying Expert Systems to Power Systems", IEEE, Vol. 80, número 5, May 1992

- [14] Hayes-Roth, F., Waterman, D.A. and Lenat, D.B., "Building Expert Systems", Addison-Wesley Publishing Company, 1983.
- [15] Kowalski, R., "Logic for Data Description", Logic and Database, Plenum Press, 1978.
- [16] Bittencourt, G., "JAI I - Tópicos em Representação de Conhecimento", INPE - São José dos Campos - S.P.
- [17] Brachman, R.J., Fikes, R.E. and Levesque, H.J., "Krypton : A Functional Approach to Knowledge Representation", IEEE, October 1983.
- [18] Rumpel, D. and Krost, G., "Natural Language Interface and Database Issues in Applying Expert Systems to Power Systems", IEEE, Vol. 80, número 5, May 1992

DATA	07/10/1996
PROC.	
PED.	PPG
LIV.	
RS	Reação

INV. N.º 10	
BIM - EFEI	
DATA	Conteúdo

EFEI - BIBLIOTECA MAUÁ

8200846



NÃO DANIFIQUE ESTA ETIQUETA