

Uma Linguagem Pseudo-Natural para Construção de Interfaces Multimídia

Mirela Gico Casado

Ana Carolina Salgado

Departamento de Informática - UFPE
Av. Prof. Luiz Freire S/N
Cidade Universitária - Caixa Postal 7851 - Recife - PE
CEP 50739
e-mails: mgc@di.ufpe.br & acs@di.ufpe.br

Abstract

This work proposes an interface definition by users using for that a pseudo-natural language, PNAT. It includes the interpreter and its components : lexical analyzer, syntactical analyzer and executor. It is presented the interface system formalization to make uniform the interaction environment. The goal of the PNAT language is to facilitate the human-machine interaction since the interface manipulation becomes more pleasant and simple.

Key-words

linguagem natural, linguagem pseudo-natural, parsing, interação homem-computador

1 Introdução

Na maioria dos sistemas de banco de dados atuais, há uma crescente preocupação na melhoria da interface com os usuários. Dentre os vários tipos de interação homem-máquina existentes, menus, manipulação direta, linguagens de comando [Shné87], uma das formas mais humanas de manifestação externa da atividade mental é o uso da Linguagem Natural (sendo um tipo de linguagem de comando). É a maneira que exprime mais facilmente idéias e desejos do ser humano.

Na comunicação homem-máquina o uso de linguagem natural requer a consideração de vários aspectos. Um dos grandes problemas é o da implementação de alguma forma de interpretação da linguagem, já que na nossa língua, uma mesma frase pode ter vários sentidos, ou mesmo várias frases podem querer expressar a mesma idéia. Não há portanto ainda um sistema que se possa definir como um sistema dito *completo*. Na realidade o que se tem são *sistemas simplificados* [Arab89], que utilizam subconjuntos de Linguagem Natural. Para tentar facilitar o acesso dos usuários ao sistema é definida uma *Linguagem Pseudo-natural*, a *PNAT*.

Em sistemas de Banco de Dados uma forma de fornecer ao usuário acesso às informações armazenadas é permitir a estes usuários uma maneira mais natural de expressar suas consultas. A linguagem definida, *PNAT*, é usada não como uma linguagem de consulta às informações do Banco de Dados, como na maioria das propostas de utilização de Linguagens Naturais ([Luk 86], por exemplo), mas sim para construir o ambiente de consulta. Para tal são utilizados conceitos do Sistema de Interfaces definido em [Salg90], que trata da manipulação de objetos complexos, usando uma estrutura denominada *cenário*.

Na próxima seção são apresentados os principais conceitos relacionados à interface. Na seção 3, é dada uma visão geral sobre Processamento em Linguagem Natural, sua importância, usos e problemas relacionados. A seção 4, apresenta a definição da linguagem *PNAT*. Finalmente, na última seção é apresentada uma breve conclusão e algumas observações sobre o trabalho.

2 Especificação da Interface

A interface é representada por uma estrutura complexa, que denominamos *cenário*¹[Salg88], um novo conceito de interface de comunicação com o usuário, similar ao de “story” do sistema Storyboard da IBM [IBM 87] e ao de “card” da Hypercard da Apple [Appl88]. Toda interação com o usuário é realizada através de cenários que podem ser pré-definidos pelo sistema ou definidos pelo próprio usuário. Estes cenários são apresentados através de objetos complexos (objetos multimídia [Fium87] e a maneira como o usuário define estes cenários é o ponto de interesse deste trabalho.

Antes de definirmos formalmente o ambiente de interface, vamos definir as diferentes classes de objetos a ele relacionadas: *cenários, cenas, mensagens, regras e restrições*. Um cenário é um conjunto de cenas às quais são associadas mensagens. A idéia de base é a ativação de cenas representadas no espaço ao longo do tempo. Um cenário, sendo uma forma de diálogo, é portanto dinâmico e necessita considerar informações tanto de tempo para a interpretação das necessidades do usuário, como de espaço para a apresentação dos diálogos.

Definição 1 : Restrições de tempo e de espaço (*RTE*)

Uma restrição de tempo e de espaço é um par : $RTE = \langle D, P \rangle$

onde : *D* e *P* indicam respectivamente uma duração e uma posição no espaço de representação.

Definição 2 : Mensagem (*MSG*)

Uma mensagem é definida pela quádrupla : $MSG = \langle M, V, RTE_m, T \rangle$

onde *M* : método associado à mensagem,

V : valor resultante da execução do método,

RTE_m: restrições de tempo e de espaço associadas a mensagem : *D* e *P*

T_p : temporização em relação ao começo da interpretação da cena a qual a

¹O termo cenário é totalmente independente do conceito de cenário usado em aspectos específicos de Inteligência Artificial.

mensagem está associada. Esta noção permite definir a ordem de ativação das mensagens em uma cena.

Definição 3 : CENA

Uma cena é uma tripla : $CENA = \langle MSG, RTEc, R \rangle$

onde $RTEc$: são as restrições de tempo e espaço associadas à cena: D e P

R : é a representação da cena para o usuário.

Definição 4 : REGRA

Uma *REGRA* pode ser definida pela regra de produção:

$REGRA = \langle \text{Se } COND \text{ Então } MSG \rangle$

onde $COND$: Corresponde a uma expressão lógica que associa uma ou mais cenas e que, se for verdadeira, ativará a *MSG* relacionada.

Podemos agora definir formalmente o conceito de cenário.

Definição 5 : CENÁRIO

Um cenário é o par : $CENÁRIO = \langle cenas, regras \rangle$

onde $\{cenas\}$: é o conjunto de cenas que compõem o cenário;

$\{regras\}$: define o agrupamento das cenas a ser considerado no momento da interpretação do cenário.

A interpretação de uma cena pode gerar, por exemplo, a visualização de uma imagem ou seqüência de imagens em um monitor, a emissão de uma informação de áudio, a execução de um programa, a edição de um texto, ..., caracterizando assim um ambiente multimídia.

O nosso modelo de dados, pode ser representado como um objeto complexo com um conjunto de atributos [Bane88]. Uma classe pode ser definida como um domínio de um atributo. Este domínio, a menos que seja um domínio simples (D, P, M, V, \dots), consiste de um conjunto de atributos, onde cada domínio de cada atributo corresponde a uma nova classe e assim por diante. Se o domínio de um atributo é um conjunto de uma classe, o valor deste atributo para qualquer instância será um conjunto de objetos (Fig. 1).

3 Processamento em Linguagem Natural

Processamento em Linguagem Natural (PLN) é a área da Inteligência Artificial voltada à geração automática e compreensão da linguagem humana.

Há boas razões para considerar o uso destas linguagens em computadores. A primeira razão é que as informações a serem manipuladas normalmente estão em linguagem natural. A segunda razão é que as pessoas necessitam para se comunicar com as máquinas de linguagens mais "naturais", já que estas facilitam sua comunicação.

Infelizmente, embora o mapeamento entre a linguagem como um todo e o mundo considerado seja natural, o mapeamento entre um modelo e a parte correspondente da linguagem

não é tão natural. Sem dúvida, um dos maiores problemas no Processamento de Linguagem Natural é o da ambigüidade, isto é, a existência de várias formas de entendimento de uma mesma frase. Nos Sistemas de PLN o maior problema é a transformação de uma frase potencialmente ambígua em uma não ambígua que será utilizada pelo sistema. Esta transformação é conhecida como "Parsing".

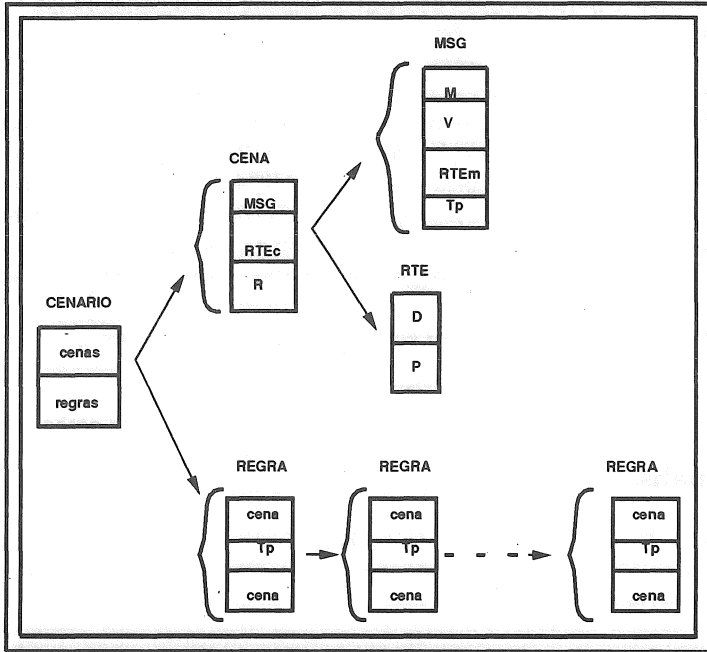


Fig 1 Classes de objetos da Interface

Existem cinco maneiras de se realizar o "parsing": Pattern-Matching [Weiz66], Baseada em Gramáticas [Hopc79], Semântica, Baseada em Conhecimento e Redes Neurais.

4 PNAT : uma Linguagem Pseudo-Natural

A PNAT é uma linguagem pseudo-natural que objetiva facilitar a interação do usuário com a máquina, permitindo que estes construam seus próprios ambientes de trabalho. Normalmente o que se tem é a possibilidade de acesso às consultas ao Banco de Dados utilizando LN.

Como qualquer linguagem, a PNAT, necessita ser interpretada para que possa ser executada. É necessária então a definição de um *interpretador*. São utilizadas para a sua

interpretação duas técnicas : “Parsing” baseado em gramática e Pattern-Matching. Este interpretador é composto de dois analisadores e um módulo executar.

Os analisadores são: analisador léxico e analisador sintático. Para a análise sintática, foi escolhida uma gramática livre de contexto ao invés de uma sensível ao contexto porque não foram consideradas restrições rígidas para concordâncias nominal e verbal. A análise não é feita rigidamente, isto é, tanto faz o usuário digitar *a interface* como *as interface*. Não é considerado este tipo de erro gramatical. Assume-se que existem palavras-chave que são os objetos e os comandos de manipulação da interface. Na gramática são consideradas algumas palavras sinônimas, para permitir ao usuário maior flexibilidade.

O módulo executor utiliza a técnica de Pattern-matching. Ele associa a cada palavra-chave encontrada, no conjunto gerado pelos analisadores na etapa anterior, ações que ele deverá realizar. Estas ações são as necessárias à manipulação da interface.

A interpretação é realizada sobre cada unidade elementar de comunicação que são as frases [Luft74]. A interpretação gera a transformação da frase do usuário em uma frase a ser entendida internamente pela máquina e que ao mesmo tempo traduza a mesma intenção da frase do usuário. Nota-se que como a preocupação neste trabalho se concentra na construção de interfaces, a gramática da linguagem se restringe apenas aos comandos necessários para construção, manipulação e remoção destas, desprezando as frases que se destinem a consultas.

São descritos a seguir os módulos componentes do Interpretador.

4.1 Analisador Léxico

O Analisador Léxico tem a função de receber a frase do usuário e transformá-la em uma lista de palavras, para que possa ser processada pelo Analisador Sintático. Uma frase acaba sempre com um ponto.

Em uma frase a unidade elementar é a *palavra*. Uma palavra é considerada como um conjunto de caracteres, que incluem letras maiúsculas e minúsculas, os números arábicos e todos os caracteres de pontuação (, . ? ! ” ...). Sabe-se que chegou ao fim de uma palavra quando encontra-se o caracter “ ” (branco). Existem na PNAT *palavras-chave* da linguagem e quando é realizada a Análise Léxica, é feita a uniformização e retirada de porções supérfluas das palavras-chave. O critério utilizado para a escolha do que seriam estas partes supérfluas das palavras adveio da percepção da menor porção da palavra, isto é, dos caracteres mínimos que identifiquem o objeto ou o comando mencionado.

Por exemplo, na frase : (*Construir Interface :ajuda*) o Analisador Léxico devolve então a seguinte lista de palavras : *CONST, INTER, :AJUDA*. Houve, por exemplo, a supressão na palavra *Construir* da porção *ruir*, e todas as palavras foram colocadas em maiúsculo integralmente. Tanto faz o usuário escrever : *Construir Interface :ajuda* ou *construiremos Interface ajuda* ou qualquer tempo verbal ou inflexão pois as porções “a mais” são desprezadas. Na verdade, só são analisadas as frases que visem a manipulação de interfaces. As demais não são tratadas.

4.2 Analisador Sintático

Este módulo é o responsável pela verificação se o formato (sintaxe) dos comandos é válido. A *sintaxe de uma linguagem* é o estudo das regras que determinam quais cadeias de palavras podem formar frases [Sava88].

Foi escolhida uma gramática Livre de Contexto para geração da PNAT. Este tipo de gramática é muito utilizado na análise de Linguagens de Programação pela sua simplicidade e eficiência dos analisadores obtidos. Por conseguinte também é muito usada na análise de Linguagens Pseudo-Naturais.

As palavras apresentadas em **negrito** na gramática (Fig 2), representam os símbolos terminais, isto é, as palavras-chave. As porções grifadas são suficientes para que as palavras sejam reconhecidas pelo Analisador Léxico. Todos os objetos que aparecem entre // (colchetes), são opcionais na emissão do comando. Se forem omitidos, o sistema se encarregará de questioná-los ao usuário. Nota-se que há também a presença de alguns sinônimos (ALTERAR E MODIFICAR, CONSTRUIR E FAZER, MOSTRAR E APRESENTAR, ...) com o intuito de permitir ao usuário mais opções no momento da interação.

Como visto na seção 2, a interface é composta de cenários e estes por sua vez são compostos de cenas. Quando um usuário requisita a construção de uma cenário, o sistema deve ir descendo até o nível mais baixo da *árvore de construção de interface*. A árvore de construção de interface (ACI) corresponde à árvore de hierarquia entre os objetos componentes de uma interface (Fig. 3). Descer até o nível mais baixo da árvore de hierarquia significa que o sistema interpela o usuário sobre os objetos do nível imediatamente inferior e continua descendo até que se chegue às folhas da árvore.

Nota-se que há sempre a ligação entre os objetos manipulados via identificadores que são colocados nos níveis inferiores da ACI, isto é, cada cena, por exemplo, possui na sua construção a identificação do cenário do qual faz parte. Deve-se notar também que alguns componentes dos objetos manipulados pela PNAT foram omitidos das produções da gramática. Um exemplo é a quantidade de cenários componentes de uma interface, as restrições associadas às cenas, as posições de representação das restrições de tempo e espaço. Todos estes elementos são requisitados ao usuário pelo sistema no momento da construção do objeto que o contém.

4.3 Executor

Este módulo é o encarregado de executar as frases que já foram verificadas pelos módulos de análise. Este módulo só é ativado se a frase conseguiu passar pela análise sintática e léxica

Neste ponto as porções supérfluas das palavras já foram retiradas e o que se tem é um conjunto de palavras-chave e de identificadores. Cada identificador vem precedido de dois pontos (:).

A cada palavra-chave está associado um conjunto de ações (rotinas) a serem realizadas. O método utilizado é o de Pattern-Matching.

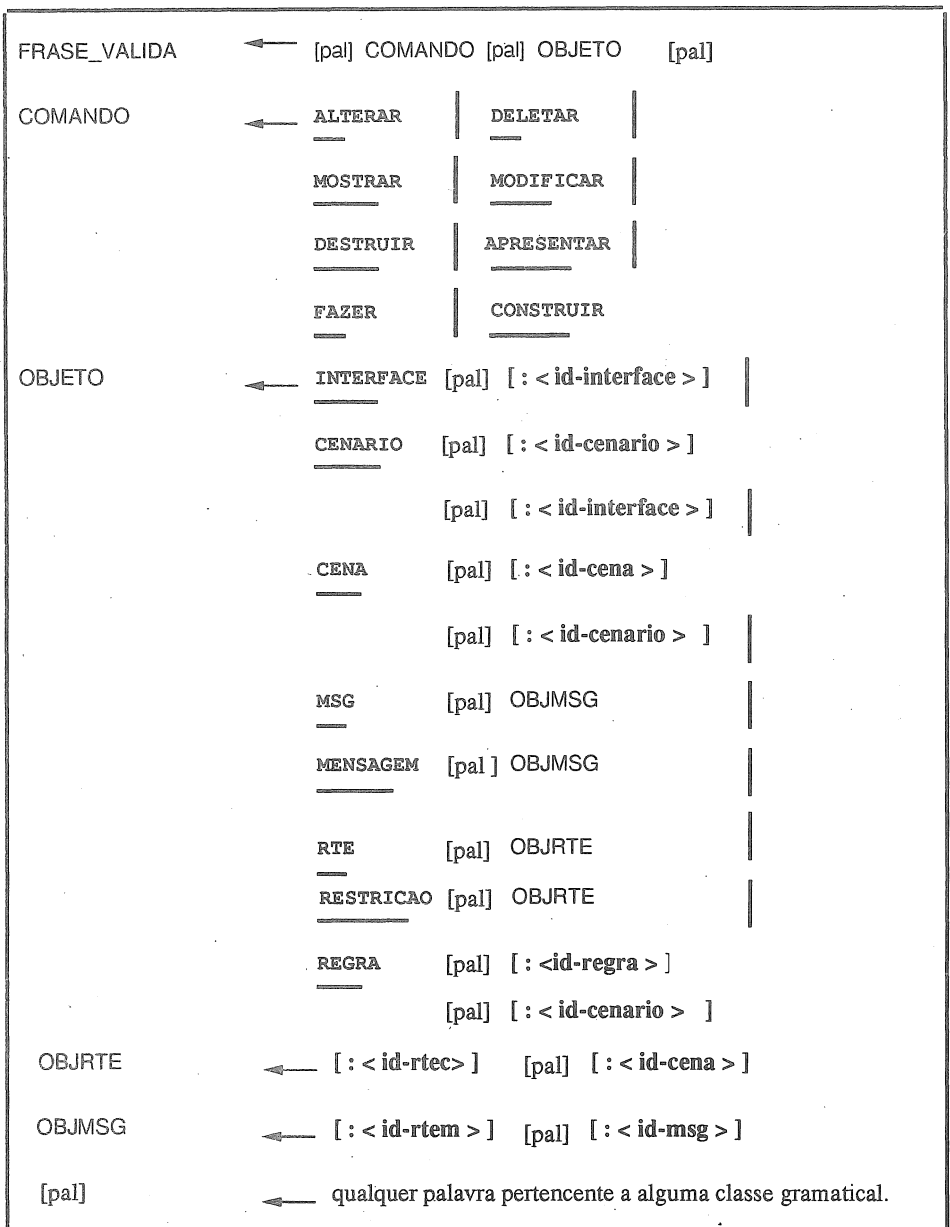


Fig 2 Gramática da PNAT

Para que os comandos sejam entendidos corretamente, é necessário apenas que alguma

parte ou partes dos comandos sejam escritos, porém a inclusão de palavras que não fazem parte da sintaxe da linguagem tais como artigos, adjetivos, conjunções preposições etc., ou mesmo a mudança do tempo verbal dos comandos, não influi no “entendimento” por parte do Executor, já que estas foram retiradas pelo Analisador Léxico.

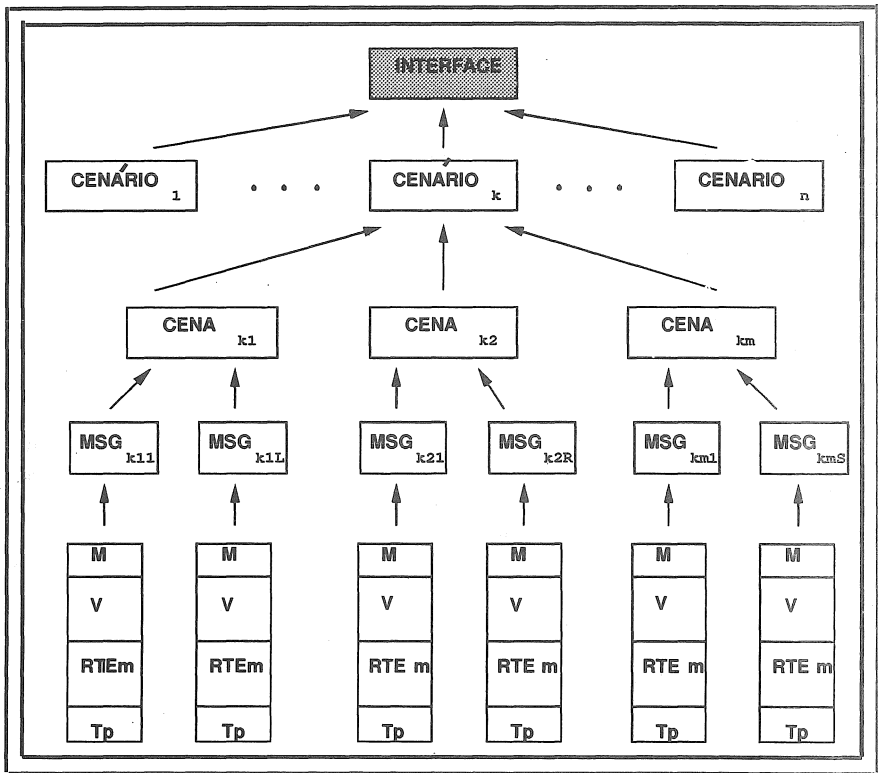


Fig 3 Árvore de Construção de Interface

As seguintes frases :

- *CONSTRUIR INTERFACE* :I1
- *CONSTRUA INTER* :I1
- *CONSTRUA UMA INTERFACE* :I1

teriam o mesmo efeito de execução isto é, seria contruída a interface II. O sistema se encarrega de questionar ao usuário sobre a quantidade de cenários associados a esta.

Para os comandos de alteração e remoção é suficiente que o usuário especifique qual objeto (Cenário, Cena, ...) que deseja alterar ou remover indicando ou não o seu identificador. Se o identificador não é indicado, este é questionado ao usuário. O usuário deve ao menos conhecer em qual interface está o objeto a ser alterado ou removido. Ao ser identificado, a estrutura do objeto é apresentada ao usuário que pode então ter acesso às informações que deseja alterar. Se o objeto a ser deletado não for uma folha na ACI, será solicitado ao usuário uma confirmação da remoção das estruturas abaixo deste objeto.

O comando MOSTRE proporciona ao usuário a visualização da ACI de uma dada interface a partir do objeto desejado, caso seja especificado o id-obj, ou de um determinado nível da ACI, caso o id-obj não seja especificado. No segundo caso, todos os objetos são apresentados um de cada vez (as sub-árvores). Vale frisar que o usuário deve especificar o identificador da interface que ele deseja visualizar.

5 Conclusão

Atualmente não existe ainda uma linguagem dita totalmente "natural" que atenda a todas as necessidades do usuário. O escopo de aplicação tem que ser restringido pela dificuldade que existe em expressar fielmente os objetivos dos usuários. Isto é consequência da inerente ambigüidade da linguagem natural. Na realidade o que existe atualmente são linguagens Pseudo-Naturais com objetivos próprios como em : [Jans90], [Oliv89], [Arab89], [Bier85], [Luk 86], [White90], entre outros trabalhos.

A linguagem Pseudo-Natural descrita facilita a interação com o usuário, permitindo-lhe construir as interfaces para as suas aplicações. O que a diferencia das demais linguagens Pseudo-Naturais, é que esta não é uma linguagem de consulta à base de dados, mas sim uma linguagem dedicada à construção de interfaces. Utiliza para tal os conceitos de cenários, cenas, regras, etc., que permitem a uniformização das interfaces a serem construídas.

Referências

- [Appl88] Hypercard User's Guide, Apple Computer, Inc., 1988.
- [Arab89] ARABE A. M. F. : "Comunicando-se com Bases de Dados Estáticas em Linguagem Natural : O protótipo de uma interface", 6 Simpósio Brasileiro em Inteligência Artificial, 1989.
- [Bane88] BANERJEE J., KIM W., KIM K.: "Queries in Object-Oriented Databases", IEEE, 1988.
- [Barn90] BARNETT J., KNIGHT K., MANI I., RICH E., "Knowledge and Natural Language Processing", Communications of the ACM, Vol. 33, No. 8, 1990.
- [Bier85] BIERMANN A.W., RODMAN R.D., RUBIN D.C., HEIDLAGE J.F., "Natural Language with Discrete Speech as a Mode for Human to Machine Communications", Communications of the ACM, Vol 28, No. 6, 1985.

- [Fium87] FIUME E., TSICHRITZIS D. : "Multimedia Objects", IEEE, 1987.
- [Harri78] HARRISON, M.A. : "Introduction to Formal Language Theory.", Addison-Wesley, 1978.
- [Hirs88] HIRSCHBERG J., BALLARD W.B., HINDLE D., "Natural Language Processing", AT & T Technical Journal, Vol 67, 1988.
- [Hopc79] HOPCROFT J. E., ULLMAN J. D., "Introduction to Automata Theory, Languages, and Computation", Addison-Wesley Publishing Company, 1979.
- [IBM 87] IBM Storyboard Plus, Manual de utilização do sistema, IBM Corporation, Junho 1987.
- [Jans90] JANSEN B., PROIX C., GROSZ G., ROLLAND C., "Using Object Concepts to Show the mapping between Knowledge in Quasi-Natural Language and Prolog", Institut Blaise Pascal-4, Place Jussieu, 1990.
- [Luft74] C.P. LUFT. : "Moderna Gramática Brasileira", Editora Globo, 1974.
- [Luk 86] LUK W. S., KLOSTER S., "ELFS : English Language From SQL", ACM TODS, Vol 11, No. 4, 1986.
- [Ober87] OBERMEIER K. K., "Natural-Language Processing", Byte, 1987.
- [Oliv89] OLIVEIRA C.A. : "A Sintaxe, a semântica e a pragmática : um enfoque integrado baseado no conhecimento lingüístico textual ", 6 Simpósio Brasileiro em Inteligência Artificial, 1989.
- [Salg88] SALGADO DE AGUIAR A.C. : "Contribution a un SGBD orienté objet (NICEBD) : traitement des données et des interfaces multimedia", Tese de doutorado, Universidade de Nice, Outubro 1988.
- [Salg90] SALGADO DE AGUIAR A.C., FONSECA D., CASADO M. G. : "Uniformização de Interfaces de comunicação em um Ambiente Multimídia ", 5 SBBB, 1990.
- [Sava88] SAVADOVSKY P. : "Introdução ao projeto de Interfaces em Linguagem Natural ", SID Informática, 1988.
- [Shne87] SHNEIDERMAN B., " Designing the User Interface : Strategies for Effective Human-Computer Interaction", Addison-Wesley Publishing Company, 1987.
- [Weiz66] Weizenbaum J., "ELIZA - A Computer Program For the Study of Natural Language Communication Between Man And Machine", Communications of the ACM, Vol 9, Num 1, 1966.
- [Whit90] WHITE G.M., "Natural Language Understanding and Speech Recognition", Communications of the ACM, Vol. 33, No. 8, 1990.