



**PUC**  
**CAMPINAS**  
PONTIFÍCIA UNIVERSIDADE CATÓLICA

**CENTRO DE CIÊNCIAS EXATAS, AMBIENTAIS  
E DE TECNOLOGIAS**

**PROGRAMA DE PÓS-GRADUAÇÃO *STRICTO-SENSU***

**PAULO EDUARDO DOMINGUES**

**DESENVOLVIMENTO DE MÉTODO PARA  
CONSULTA EM LINGUAGEM NATURAL DE  
COMPONENTES DE *SOFTWARE***

**PUC CAMPINAS**

**2007**

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS**

**GRÃO-CHANCELER**

Dom Bruno Gamberini

**MAGNÍFICO REITOR**

Prof. Pe. Wilson Denadai

**VICE-REITOR**

Prof<sup>a</sup>. Dra. Ângela de Mendonça Engelbrecht

**PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO**

Prof<sup>a</sup>. Dra. Vera Engler Cury

**DIRETOR DO CENTRO DE CIÊNCIAS EXATAS, AMBIENTAIS E DE  
TECNOLOGIAS**

Prof. Dr. Orandi Mina Falsarella

**COORDENADORES DO PROGRAMA DE**

**PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**CURSO DE MESTRADO PROFISSIONAL EM TELECOMUNICAÇÕES**

**ÁREA DE CONCENTRAÇÃO: GESTÃO DE REDES DE TELECOMUNICAÇÕES**

Prof. Dr. Omar Carvalho Branquinho

Prof. Sérgio Roberto Pereira

**PAULO EDUARDO DOMINGUES**

**DESENVOLVIMENTO DE MÉTODO PARA  
CONSULTA EM LINGUAGEM NATURAL DE  
COMPONENTES DE *SOFTWARE***

Dissertação apresentada como exigência para obtenção do Título de Mestre em Engenharia Elétrica, ao Programa de Pós-Graduação na área de concentração Gestão de Redes e Serviços, Pontifícia Universidade Católica de Campinas.

Orientador: Prof. Dr. Joinvile Batista Junior

**PUC CAMPINAS**

**2007**

Ficha Catalográfica  
Elaborada pelo Sistema de Bibliotecas e  
Informação – SBI – PUC-Campinas

t004.6  
D671d Domingues, Paulo Eduardo.  
Desenvolvimento de método para consulta em linguagem natural de componentes de software / Paulo Eduardo Domingues. - Campinas: PUC-Campinas, 2007.  
154p.

Orientador: Joinville Batista Júnior.  
Dissertação (mestrado) – Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologias, Pós-Graduação em Engenharia Elétrica. Inclui anexos e bibliografia.

1. Interfaces de usuário (Sistema de computador). 2. Linguagem de programação (Computadores). 3. Sistemas de recuperação da informação. 4. Software.  
5. Telecomunicações. I. Batista Júnior, Joinville. II. Pontifícia Universidade Católica de Campinas. Centro de Ciências Exatas, Ambientais e de Tecnologias. Pós-Graduação em Engenharia Elétrica. III. Título.

22.ed.CDD – t004.6

## **BANCA EXAMINADORA**

Presidente e Orientador: Prof. Dr. Joinvile Batista Junior

1° Examinador: Prof. Dr. Marcelo Luis Francisco Abbade

2° Examinador: Prof. Dr. Marcio Luiz de Andrade Neto

Campinas, 28 de Setembro de 2007.

## **AGRADECIMENTOS**

A Deus que ouviu o meu pedido e me conduziu durante todos os passos desta dissertação.

A providência que tudo proveu, a minha esposa, Márcia, pela compreensão ao tempo dedicado aos estudos, pelo apoio e incentivos irrestritos.

Ao Prof. Dr. Joinvile Batista Junior,  
Orientador e mestre sempre atento e aplicado na minha formação do mestrado profissional pela Faculdade de Engenharia Elétrica da Pontifícia Universidade Católica de Campinas.

Aos familiares e amigos pela torcida e estímulo.

"A mente que se abre a uma nova idéia jamais voltará ao seu tamanho original".

Albert Einstein  
(1879 – 1955)

## RESUMO

DOMINGUES, Paulo Eduardo. Desenvolvimento de método para consulta em linguagem natural de componentes de *software*. 2007. 154f. Dissertação (Mestrado em Gestão de Redes de Telecomunicações) – Pós-Graduação em Engenharia Elétrica, Centro de Ciências Exatas, Ambientais e de Tecnologias, Pontifícia Universidade Católica de Campinas, Campinas, 2007.

O desenvolvimento baseado em componentes permite criar componentes inter-operáveis, com interfaces bem definidas, reduzindo a complexidade no desenvolvimento de software. Neste cenário, a biblioteca de componentes de software exerce um papel importante em um ambiente corporativo, suportando a documentação, especificação, armazenamento e recuperação de componentes. Dentro das organizações, uma biblioteca de componentes fornece uma infra-estrutura para o gerenciamento do ciclo de vida dos componentes. Este trabalho propõe o armazenamento e a recuperação de componentes de software com a utilização de uma interface em linguagem natural. É descrito um método para gerar uma forma de representação, a ser armazenada na biblioteca, para os textos que descrevem as características dos componentes que integram a biblioteca. O texto da consulta gerada pelo usuário também é representado de forma semelhante para permitir a comparação entre as descrições dos componentes da biblioteca e a questão do usuário. Adicionalmente, é apresentado o método para determinar a semelhança entre partes das representações do texto das características com o texto das consultas, de forma a retornar como resultado a indicação em ordem decrescente de prioridade os componentes que melhor atendem a consulta do usuário.

Termos de Indexação: linguagem natural, extração de relações, biblioteca de componentes, reuso de *software*.



## ABSTRACT

DOMINGUES, Paulo Eduardo. *Development of Method for Natural Language Research of Software Components*. Campinas, 2007. 154 pages. Master's Thesis – Department of Electrical Engineering (Telecommunications), Pontifícia Universidade Católica de Campinas. Campinas, 2007.

*The development based on components allows to create inter-operable components, with well defined interfaces, reducing the complexity in the software development. In this scene, the library of software components plays an important role in corporate level, supporting documentation, specification, storage and recovery of components. Inside organizations, a components library supplies infrastructure for components lifecycle management. This work considers the storage and the recovery of components of software with the use of an interface in natural language. A method to generate a representation form is described, to be stored in the library, for the texts that describe the characteristics of the components that live in the library. The text of the research generated for the user also is represented of similar form to allow the comparison between the descriptions of the components of the library and the question of the user. Additionally the method is presented to determine the similarity between parts of the representations of the text of the characteristics with the text of the research, of form to return as resulted in sequence decreasing indication from priority the components that better take care of the research of the user.*

*Index terms: natural language, relation extraction, components library, software reuse.*

# LISTA DE FIGURAS

Figura 1 - Descrição Padrão <i>Bridge</i> .....	22
Figura 2 - Reuso de Componentes de <i>Software</i> : Grafo de Similaridade Conceitual .....	28
Figura 3 - Representação análise Gramatical .....	44
Figura 4 - Exemplo pessoa-afiliação gerado pelo <i>shallow parse</i> .....	45
Figura 5 - Árvore de dependência para a frase: <i>Troops advanced near Tikrit</i> .....	46
Figura 6 - Exemplo Simplificado de uma proposta de representação do “Corpus” .....	50
Figura 7 - Representação da Árvore Sintática de um texto.....	58
Figura 8 - Legenda da Notação dos Nós do Grafo FGR.....	65
Figura 9 - Legenda da Notação dos Arcos do Grafo FGR.....	66
Figura 10 - Legenda da Notação dos <i>Labels</i> do Grafo FGR.....	66
Figura 11 - FGR da Faceta do propósito do padrão de projeto “ <i>Chain of Responsibility</i> ”.....	68
Figura 12 - FGR da Faceta do propósito do padrão de projeto “ <i>Memento</i> ”.....	68
Figura 13 - FGR original da faceta propósito do padrão de projeto “ <i>Abstract Factory</i> ”.....	70
Figura 14 - FGR reduzida da faceta do propósito do padrão de projeto “ <i>Abstract Factory</i> ”.....	71
Figura 15 - Algoritmo de <i>Matching</i> das FGRs - <i>Composite</i> .....	83
Figura 16 - Algoritmo de <i>Matching</i> das FGRs – <i>Abstract Factory</i> .....	84
Figura 17 - Algoritmo de <i>Matching</i> das FGRs – <i>Abstract Factory</i> .....	85
Figura 18 - Algoritmo de <i>Matching</i> das FGRs – <i>Abstract Bridge</i> .....	86
Figura 19 - Relações e modificadores mais freqüentes.....	87

## LISTA DE ABREVIATURAS E SIGLAS

ACE	=	Extração de Conteúdo Automática
AS	=	Árvore Sintática
FGR	=	Forma Gráfica de Representação
MUC	=	<i>Message Understanding Conferences</i>
NR	=	Núcleo da Relação
REES	=	<i>Relation and Event Extraction System</i>
TE	=	Elemento do <i>Template</i>
TR	=	Relação do <i>Template</i>
TREC	=	<i>Text Retrieval Conference</i>
T-REX	=	<i>Trainable Relation Extraction</i>

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>15</b>
1.1	Objetivo do Trabalho.....	15
1.2	Representação e Busca em Biblioteca de Componentes.....	18
1.3	Utilização de Componentes de Padrões de Projeto.....	19
1.4	Consultas em Linguagem Natural.....	23
1.5	Justificativa do Trabalho.....	23
1.6	Descrição Sucinta dos Capítulos da Dissertação.....	24
<b>2</b>	<b>CONTEXTUALIZAÇÃO DO PROBLEMA</b> .....	<b>26</b>
2.1	Representação e Busca de Componentes de <i>Software</i> .....	26
2.2	Resposta a Consultas.....	29
2.2.1	Resposta a Consultas e suas Áreas Correlatas.....	29
2.2.2	Características da Área de Resposta a Consultas.....	31
2.3	Principais Abordagens de Resposta a Consultas.....	33
2.4	Atividades Típicas na Abordagem Baseada em Conhecimento.....	34
2.4.1	Análise da Questão.....	34
2.4.2	Pré-processamento da Coleção de Documentos.....	38
2.4.3	Seleção de Candidatos de Documentos de Resposta.....	38
2.4.4	Análise de Candidatos de Documentos de Resposta.....	39
2.4.5	Extração de Resposta.....	40
2.4.6	Geração de Resposta.....	41
2.5	Extração de Relações.....	41
2.5.1	<i>Menor Caminho de Núcleos Dependentes para a Extração da Relação</i> .....	42
2.5.2	<i>Método de Núcleo para a Extração da Relação</i> .....	43

2.5.3	<i>Núcleo de Árvore Dependente para a Extração da Relação</i> .....	45
2.5.4	REES: <i>Sistema de Extração de Relações e Eventos em Larga Escala</i> .....	46
2.5.5	TREX: <i>Framework Flexível para Extração da Relação</i> .....	49
<b>2.6</b>	<b>Contextualização do Método Proposto .....</b>	<b>50</b>
<b>2.7</b>	<b>Conceitos e Ferramenta Utilizados para a Representação e Busca dos Padrões de Projeto.....</b>	<b>54</b>
2.7.1	Lematização .....	55
2.7.2	Co-referência.....	56
2.7.3	Ontologia.....	56
2.7.4	Ferramenta de Rotulação Sintática e Semântica .....	57
<b>3</b>	<b>DETALHAMENTO DO MÉTODO PROPOSTO.....</b>	<b>59</b>
<b>3.1</b>	<b>A Escolha da Faceta Propósito para o Experimento da Dissertação</b>	<b>59</b>
<b>3.2</b>	<b>Critérios para a Concepção do Método e Análise de suas Conseqüências .....</b>	<b>60</b>
<b>3.3</b>	<b>Definição de uma Solução para Atender os Critérios Definidos.....</b>	<b>61</b>
<b>3.4</b>	<b>Atividades do Método Proposto.....</b>	<b>63</b>
3.4.1	Rearranjo da Redação dos Propósitos para Gerar uma Árvore Sintática (AS) Base para a Geração da Forma Gráfica de Representação (FGR).....	64
3.4.2	Notação da Forma Gráfica de Representação (FGR) .....	64
3.4.3	FGR dos Propósitos a partir da AS Base .....	69
3.4.4	Reduções Possíveis de Automação na FGR .....	69
3.4.5	Sugestões de Redação para Bibliotecário e Usuário da Biblioteca..	71
<b>4</b>	<b>AValiação E Análise DO Método Proposto.....</b>	<b>80</b>
<b>4.1</b>	<b>Rastreabilidade das Avaliações .....</b>	<b>80</b>
<b>4.2</b>	<b>Resultados Obtidos.....</b>	<b>86</b>

<b>5</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS.....</b>	<b>90</b>
<b>6</b>	<b>REFERÊNCIAS .....</b>	<b>93</b>
<b>7</b>	<b>ANEXOS .....</b>	<b>96</b>
7.1	Anexo A – AS Base para gerar FGRs dos Propósitos. ....	96
7.2	Anexo B – FGRs dos Propósitos dos Padrões de Projeto. ....	112
7.3	Anexo C – Reduções Automatizáveis das FGRs dos Propósitos. ...	126
7.4	Anexo D – Análise das Alterações de Redação para gerar Sugestões para os Usuários. ....	133
7.5	Anexo E – Notação Léxica utilizada pelo Parser Anexo A – AS Base para gerar FGRs dos Propósitos. ....	152
7.6	Anexo F - Notação Sintática utilizada pelo Parser. ....	153

# 1 INTRODUÇÃO

## 1.1 Objetivo do Trabalho

Atualmente existem corporações nas quais várias unidades desenvolvem *software* em ambientes isolados promovendo uma baixa sinergia entre a utilização dos componentes gerados entre estas unidades. Uma alternativa de reuso em ambiente corporativo é a utilização de uma biblioteca corporativa de componentes que permite criar visibilidade entre os componentes gerados pelos projetos da organização. Uma biblioteca de componentes fornece uma infra-estrutura para o gerenciamento do ciclo de vida dos componentes da organização. A biblioteca é uma estrutura que permite documentar, especificar, armazenar e utilizar os componentes promovendo o reuso.

Uma biblioteca de componentes estimula o reuso de componentes do repositório, permitindo à equipe de desenvolvimento de *software* encontrar soluções nos componentes da biblioteca para problemas enfrentados no desenvolvimento do projeto. Além disso, novos componentes podem ser concebidos, contribuindo para a população do repositório.

Os padrões de projetos são soluções reutilizáveis para os problemas encontrados pelos desenvolvedores de *software* orientado a objetos. Adicionalmente, estes padrões contribuem para melhorar a documentação e a manutenção de sistemas ao fornecer uma especificação explícita de interações de classes e objetos e o seu objetivo subjacente. Padrões de projetos refletem uma experiência consolidada na obtenção de soluções reusáveis para o desenvolvimento de *software*. Em geral, um padrão tem quatro elementos essenciais: o nome do padrão, o problema, a solução e as conseqüências. O nome do padrão é uma referência usada para descrever um problema de projeto, suas soluções e conseqüências em uma ou duas palavras. O problema descreve quando aplicar o padrão. Ele explica o problema e seu contexto. A solução descreve os elementos que compõem o projeto, seus relacionamentos, suas responsabilidades e colaborações.

As conseqüências são os resultados e análises das vantagens e desvantagens do padrão, que incluem o seu impacto sobre a flexibilidade, a extensibilidade ou a portabilidade de um sistema.

Assim, os desenvolvedores de *software* orientado a objetos poderiam encontrar facilmente um componente com padrões de projeto, que forneça uma solução para um problema encontrado no desenvolvimento do projeto na biblioteca. É importante ressaltar que como a biblioteca poderia conter uma diversidade de componentes com vários tipos de padrões, se faz necessário ter uma boa classificação. Desta forma os componentes de *software* podem ser documentados, armazenados e posteriormente reusados no desenvolvimento de outros *softwares*.

No mercado de telecomunicações, são essenciais a criação de padrões, recomendações e modelos de referência que permitam garantir a interoperabilidade entre sistemas na criação da infra-estrutura para fornecimento de serviços, bem como a aderência a processos operacionais e seus fluxos e interfaces. O <sup>1</sup>*Network Management Forum* (atual *Telemanagement Forum* ou TMFORUM), tem capitaneado esforços na busca de padrões da indústria, permitindo uma evolução rápida e eficaz no preenchimento das lacunas entre os fornecedores de sistemas, operadoras e outros provedores de serviços de telecomunicações. Como resultado destes esforços, foi definido o *enhanced Telecom Operations Map* (eTOM), um abrangente modelo de processos de negócios para operadoras e provedores de serviços, alinhado com as necessidades operacionais e com a evolução dos conceitos de ferramentas de *software* voltadas para esse meio. No entanto, a evolução do mercado de telecomunicações está também intimamente ligada com os avanços na tecnologia da informação (TI) e com o advento de novos modelos de governança. No ambiente de TI, foi desenvolvido o *IT Infrastructure Library* (ITIL), que é um conjunto de melhores práticas para o gerenciamento eficaz do ambiente. O ITIL foi criado pela *Central Computer and Telecommunication Agency* (CCTA). Recentemente, a análise conjunta e comparativa das diferentes abordagens (ex.:

---

<sup>1</sup> O *Network Management Forum* ou TMFORUM foi fundado em 1988 com o objetivo de acelerar a disponibilização de soluções interoperáveis de gerenciamento. Desde então, tem se adaptado à contínua evolução tecnológica e aos cenários de negócios, patrocinando alguns dos mais importantes eventos na área de Sistemas de Suporte a Operações, tendo seus padrões sido adotados por outras entidades do mundo (ex.: ITU-T).



eTOM-ITIL *Application Note*) fornece um indício de convergência. Juntos, os modelos sugeridos por <sup>2</sup>ITU-T, TMFORUM e CCTA complementam-se, permitindo um entendimento uniformizado e a viabilização de plataformas mais flexíveis e com capacidade de evoluir na mesma velocidade que as novas tecnologias, os novos serviços e as novas necessidades dos clientes (LEONEL; MAUSBACH, 2006).

O mercado de telecomunicações tem se tornado mais próximo da área de TI em uma época onde os processos são cada vez mais mantidos e gerenciados por *software*. A convergência destas duas áreas tem proporcionado maior flexibilidade e agilidade aos operadores de telecomunicações permitindo oferecer seus serviços com maior competitividade e com um grau mais elevado de personalização. A utilização de componentes na área de telecomunicações implica na aquisição e no desenvolvimento de sistemas modulares, flexíveis e principalmente com arquitetura prevista para a evolução. O objetivo da utilização de componentes é a definição de componentes interoperáveis, com interfaces bem definidas, reduzindo a complexidade no desenvolvimento, assim como os custos, através da reutilização de componentes exaustivamente testados. O desenvolvimento destes componentes deve atender premissas de benefícios em qualidade, produtividade e manutenibilidade dentro da indústria de *software* e da área de telecomunicações.

Neste cenário, a biblioteca de componentes é um elemento de grande sinergia no âmbito corporativo, aumentando a oferta, intercâmbio e a maturidade de componentes reutilizáveis.

Este trabalho define um método para armazenar e buscar componentes em uma biblioteca de componentes de *software*. O método foi concebido para suportar uma interface em linguagem natural, tanto para as descrições dos

---

<sup>2</sup> A União Internacional de Telecomunicações (ITU) é o mais antigo órgão de padronização, criado em 1865 para gerenciar as primeiras redes de telegrafia. Nesses muitos anos, acompanhou o nascimento da telefonia de voz, a evolução das comunicações por rádio, a transmissão de sinais de TV, o lançamento dos primeiros satélites e, mais recentemente, a convergência tecnológica e o início da era da informação como um fórum neutro para confronto e negociações entre indústrias, governo e sociedade.

componentes, a serem armazenadas na biblioteca, quanto para as consultas, realizadas pelos usuários da biblioteca.

## 1.2 Representação e Busca em Biblioteca de Componentes

As principais atividades para gerar e manter uma biblioteca corporativa de componentes de *software* são:

a) Criação de Componentes: a concepção dos componentes que participam da arquitetura do projeto em elaboração;

b) Aceitação de Componentes: critérios para geração de componentes no repositório, documentação, conformidade com requisitos e versionamento;

c) Representação e Recuperação de Componentes: definir uma forma de representação de componentes na biblioteca, associada a uma posterior recuperação de componentes para reuso.

A utilização de componentes possibilita a composição de *software*, a partir de componentes parametrizados e testados. Independente do tamanho da equipe se faz necessária a utilização de ferramentas para simplificar o cadastramento e a recuperação de componentes da biblioteca. A padronização na geração, cadastramento, versionamento e recuperação de componentes na biblioteca, é o fator fundamental para assegurar qualidade e a confiabilidade na reutilização efetiva de componentes no âmbito corporativo.

O foco deste trabalho é o armazenamento e a recuperação de componentes de *software* em uma biblioteca corporativa. Neste contexto, é importante caracterizar os seguintes papéis: o do bibliotecário e o do usuário da biblioteca.

O bibliotecário é o responsável pela aceitação dos componentes na biblioteca, garantido a qualidade da documentação e dos componentes inseridos, facilitando assim a sua recuperação e o seu entendimento para a utilização em outros projetos. O bibliotecário tem também a responsabilidade de submeter os

componentes na biblioteca e revisar a descrição dos componentes inseridos. A revisão da descrição do componente facilita o entendimento do usuário na escolha do componente desejado.

O usuário da biblioteca submete consultas à biblioteca, com o objetivo de obter componentes de *software* que atendam, de forma completa ou parcial, aos seus requisitos de projeto.

### **1.3 Utilização de Componentes de Padrões de Projeto**

Muitas soluções são criadas a partir de problemas semelhantes encontrados anteriormente. Muitos padrões são reconhecidos e utilizados por comunidades de diversas áreas, como arquitetura, economia e ciências da computação. Padrões de projetos, concebidos como soluções de projetos para a Engenharia de *Software* no modelo Orientado a Objetos, têm tido ampla aceitação na comunidade de engenharia de *software*.

A padronização na descrição dos padrões de projeto é fundamental para sua utilização efetiva em uma comunidade de usuários. As facetas que compõem a descrição de um padrão de projeto (BUSCHMANN, 1996), são enumeradas a seguir:

1. Nome do Padrão
2. Intenção e Objetivo
3. Também conhecido como (outro nome para o padrão)
4. Motivação
5. Aplicabilidade
6. Estrutura
7. Participantes (classes ou objetos que participam do padrão e suas responsabilidades)
8. Colaborações (como os participantes colaboram para executar suas responsabilidades)
9. Consequências
10. Implementação (que armadilhas ou técnicas se precisa conhecer para implementar)

11. Exemplos de código (blocos de códigos ilustrando como implementar)
12. Uso conhecido
13. Padrões relacionados

O nível de formalidade para expressar as diferentes partes em uma lista pode variar, mas a estrutura pode ser a mesma. Os padrões de projeto são normalmente descritos usando a *Unified Modeling Language* (UML).<sup>3</sup>

Padrões de projeto foram concebidos no contexto de arquiteturas orientadas a objetos (GAMMA, 1995). De fato, uma das maneiras de medir a qualidade de um sistema orientado a objetos é avaliar se os desenvolvedores trataram de forma adequada as colaborações comuns entre os objetos. A atenção com estes mecanismos durante o desenvolvimento de um sistema conduz à geração de uma arquitetura mais compacta, mais simples e compreensível que aquelas produzidas quando estes padrões são ignorados.

A representação de componentes em uma biblioteca corporativa de componentes de *software* para posterior recuperação e reuso, contribui de forma bastante efetiva para alcançar níveis crescentes de maturidade de *software* no ambiente corporativo.

Como soluções reutilizáveis para os problemas freqüentemente encontrados no desenvolvimento de *software* orientado a objetos, os padrões de projetos foram escolhidos como objeto de estudo para a definição de um método de descrição e busca em linguagem natural. O motivo determinante para a escolha, foi a disponibilidade de textos de boa qualidade na literatura. A documentação, em linguagem natural, dos padrões de projetos é muito utilizada pelas empresas e profissionais de desenvolvimento de *software* orientados a objetos.

---

<sup>3</sup> *Unified Modeling Language* (UML) é uma linguagem de modelagem não proprietária de terceira geração. A UML não é um método de desenvolvimento, o que significa que ela não diz o que fazer primeiro e em seguida ou como desenhar seu sistema, mas ela lhe auxilia a visualizar seu desenho e a comunicação entre objetos. Basicamente, a UML permite que desenvolvedores visualizem os produtos de seu trabalho em diagramas padronizados. Junto com uma notação gráfica, a UML também especifica significados, isto é, semântica. É uma notação independente de processos, embora o *Rational Unified Process* (RUP) tenha sido especificamente desenvolvido utilizando a UML.

Para o contexto deste trabalho, foram consideradas as seguintes facetas dos padrões de projeto:

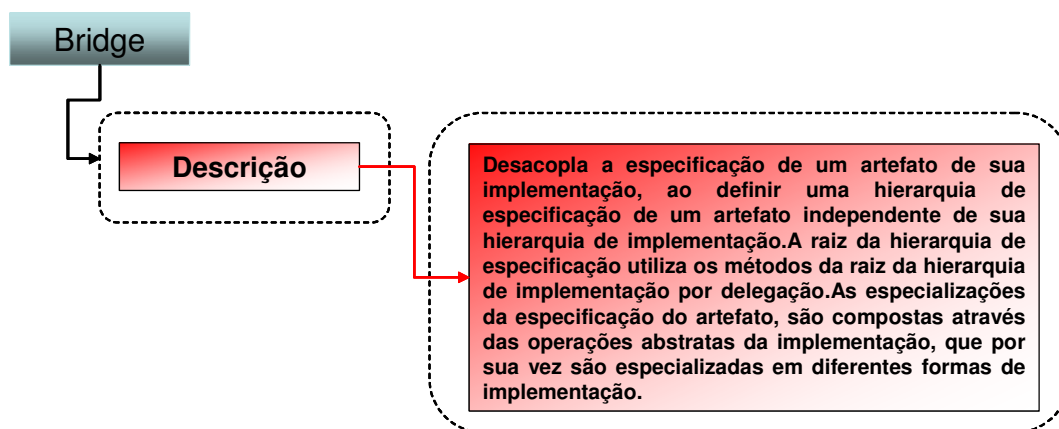
- **Intenção (Propósito):** O que faz? Que problema particular de projeto ele trata?
- **Aplicabilidade:** Onde o padrão pode ser aplicado?
- **Conseqüências:** Como o padrão suporta a realização de seus objetivos? Quais seus custos e benefícios?

**QUADRO 1 - Itens de Descrição do Padrão de Projeto *Bridge*.**

<p><b>Padrão de Projeto : Bridge</b></p> <p><b>Intenção</b></p> <ul style="list-style-type: none"> <li>• Desacopla uma abstração de sua implementação de tal modo que ambos possam variar independentemente.</li> </ul> <p><b>Aplicabilidade</b></p> <p>Este padrão é usado quando:</p> <ul style="list-style-type: none"> <li>• se quer evitar uma ligação permanente entre uma abstração e sua implementação. O que ocorre, por exemplo, quando a implementação deve ser selecionada ou trocada em tempo de execução;</li> <li>• tanto a abstração quanto sua implementação devem ser extensíveis por especialização. Neste caso, o pattern Bridge deixa você combinar estas diferentes abstrações e implementações e estendê-las independentemente;</li> <li>• mudanças na implementação de uma abstração não devem ter impacto nos clientes, isto é, seu código não deve ter que ser recompilado;</li> <li>• quando se quer esconder completamente a implementação dos clientes, evitando que a representação de uma classe seja feita através da interface dessa classe.</li> <li>• há um tipo de hierarquia de classes onde ocorre generalizações aninhadas, indicando a necessidade de se dividir um objeto em duas partes;</li> <li>• se quer compartilhar uma implementação entre vários objetos, e este fato deva estar oculto para os clientes.</li> </ul> <p><b>Conseqüências</b></p> <p>Este pattern tem os seguintes benefícios e implicações:</p> <ul style="list-style-type: none"> <li>• Desacopla a interface da implementação</li> <li>• Melhora a extensibilidade</li> <li>• Oculta detalhes de implementação do cliente</li> </ul>
---

O QUADRO 1 apresenta as facetas, de interesse para esta proposta, para o padrão de projeto **Bridge** (GAMMA, 1995).

Para que o usuário possa verificar, de forma complementar, se os padrões de projeto recuperados atendem à sua expectativa, é definida nesta proposta uma nova faceta, denominada simplesmente de “Descrição”. Esta faceta foi criada para sintetizar o entendimento das representações gráficas apresentadas na literatura dos padrões de projetos. A FIG. 1 ilustra este item para o padrão de projeto *Bridge*.



**FIGURA 1 - Descrição Padrão *Bridge*.**

A busca é realizada a partir de questões do usuário focando em um problema a ser resolvido. No processo de busca as questões do usuário são mapeadas nas facetas existentes na biblioteca. Este mapeamento é feito da seguinte forma:

- a) o objetivo, a ser alcançado com a solução do problema, é mapeado na faceta Propósito;
- b) a descrição do problema é mapeada na faceta Aplicabilidade;
- c) as restrições, que o usuário impõe à aplicação do componente para solucionar o problema, são mapeadas na faceta Conseqüência.

Após o resultado da busca, a faceta de Descrição é apresentada ao usuário para que ele possa verificar se o componente atende seus interesses.

#### **1.4 Consultas em Linguagem Natural**

Este trabalho define uma interface em linguagem natural, tanto para a representação de descrições das facetas dos padrões de projeto na biblioteca, quanto para a formulação de consultas à biblioteca de componentes.

O primeiro motivo é que a linguagem natural é a forma mais amigável para interação com seres humanos. O segundo motivo é que as pesquisas em processamento de linguagem natural têm proporcionado resultados muito promissores, conforme apresentado no capítulo seguinte.

#### **1.5 Justificativa do Trabalho**

O novo ambiente convergente é uma realidade nas telecomunicações (OSÓRIO, 2005). Seus problemas e desafios estão na agenda de agentes do mercado, operadores, fornecedores de infra-estrutura, produtores de conteúdo, reguladores e governo. A arquitetura-alvo deve ser independente da rede de acesso e os dispositivos, sejam eles móveis ou fixos (PDAs, telefones fixos, celulares, televisão, *handhelds*, PCs etc), devem permitir uma plataforma de serviços aberta e independente que possibilite a implementação e entrega rápida de novos serviços, assim como sua interoperabilidade. Junto com as redes de próxima geração, as operadoras precisam desenvolver capacitações de próxima geração para viabilizar a entrega dos novos serviços e experiências ao cliente. A camada de sistemas deve suportar a evolução das outras camadas de rede e, para isso, deve também estar orientada a serviços, independentes de rede e com interfaces abertas (OSÓRIO, 2005).

O resultado gerado pelo método desenvolvido nesta dissertação contribui para o desenvolvimento de serviços de telecomunicações baseados em componentes de *software*. O método proposto suporta a representação e busca de

componentes de *software*, no contexto de uma biblioteca corporativa, contribuindo para aumentar o reuso de *software* na construção de novos serviços de telecomunicações.

## 1.6 Descrição Sucinta dos Capítulos da Dissertação

O capítulo 2 descreve resumidamente o estágio alcançado por trabalhos em áreas correlatas: recuperação e busca de componentes de *software*, recuperação de informação, extração de informação e resposta a consultas. Adicionalmente, descreve os conceitos que serão utilizados no desenvolvimento deste trabalho, bem como a ferramenta de análise sintática e semântica que será utilizada no desenvolvimento do processo proposto.

O capítulo 3 detalha a concepção do método proposto. É definida uma notação para uma forma de representação gráfica da estrutura gramatical dos textos que descrevem os propósitos dos padrões de projeto. É gerada a forma gráfica de representação, utilizando a notação proposta, para os 23 padrões de projeto, propostos originalmente na literatura. É gerado um guia de sugestões para auxiliar a redação da descrição dos padrões de projeto, geradas pelos bibliotecários da biblioteca corporativa, e para auxiliar a elaboração das questões, geradas pelos usuários da biblioteca. É definido um mapeamento, entre a estrutura gramatical gerada por um *parser* externo e a representação gráfica dos textos. Finalmente, é definido um algoritmo de *matching*, entre as formas de representações gráficas das descrições dos propósitos dos padrões de projeto, e das questões dos usuários.

O capítulo 4 trata da validação do método proposto. Para questões, extraídas de fonte externa (ROCHA, 2006), que endereçam os objetivos dos problemas a serem solucionados pela utilização de padrões de projeto, são definidas as formas de representação gráfica. Então, o algoritmo de *matching* é aplicado para cada questão de usuário, em relação aos três padrões de projeto mais promissores. Os resultados obtidos são apresentados identificando os padrões de projeto recomendados para resolver um problema proposto.



No capítulo 5 os resultados obtidos são então utilizados para uma análise das forças e fraquezas do método proposto. Adicionalmente, são indicados trabalhos futuros para complementar o trabalho da dissertação.

No "Anexo A" são apresentados às árvores sintáticas geradas automaticamente pelo parser (analisador sintático) através de um texto escrito em linguagem natural. Neste anexo estão contidos a frase original, as tentativas de reescrita do texto com o objetivo de readequação da árvore sintática e a solução adotada. As árvores sintáticas que representam adequadamente a semântica descrita pelos textos não sofrem modificações.

No "Anexo B" são apresentadas as Formas Gráficas de Representação geradas através da Árvore Sintática Base que se encontra no "Anexo A". Cada uma destas Formas Gráficas de Representação será utilizada pelo algoritmo de *matching*.

No "Anexo C" encontram-se as reduções automatizáveis das Formas Gráficas de Representação dos Propósitos. Para cada Forma Gráfica de Representação, de cada propósito de cada padrão de projeto, são enumeradas as reduções passíveis de automação. Os padrões de projeto, para os quais não se aplica nenhuma redução, foram omitidos. A análise das reduções resultou em algumas heurísticas de redução que fazem parte do refinamento das Formas Gráficas de Representação.

No "Anexo D", são analisadas as alterações de redação, descritas no "Anexo A". O objetivo desta análise é o de determinar as sugestões de redação para o bibliotecário e para o usuário da biblioteca.

No "Anexo E" encontra-se a notação léxica utilizada pelo parser. Esta notação auxilia no entendimento da representação em forma de árvore sintática base referenciada no "Anexo A".

No "Anexo F" encontra-se notação sintática utilizada pelo Parser. Assim como o "Anexo E", esta notação auxilia no entendimento da representação em forma de árvore sintática base referida no "Anexo A".

## 2 CONTEXTUALIZAÇÃO DO PROBLEMA

Neste capítulo é descrito o estágio alcançado por trabalhos em áreas correlatas. Estas áreas são a recuperação e busca de componentes de *software*, recuperação de informação, extração de informação e resposta a consultas. Ainda neste capítulo são descritos os conceitos que serão utilizados no desenvolvimento deste trabalho, bem como a ferramenta de análise sintática e semântica que será utilizada no desenvolvimento do método proposto.

### 2.1 Representação e Busca de Componentes de *Software*

Uma das condições necessárias para que a abordagem de reutilização de *software* seja realizada é a que está relacionada ao processo de recuperação de componentes. Esta busca de componentes objetiva a classificação de possíveis soluções para que determinada necessidade seja atendida, a partir de um conjunto de características de referência. O esquema facetado, vem sendo utilizado em inúmeros trabalhos e relaciona um componente às facetadas (XAVIER, 2001). Pode-se entender como facetada o termo que melhor descreva cada um dos conceitos em um título ou domínio. Segundo Prieto-Diaz (1991) uma facetada é um tipo de descritor que auxilia a identificação de um componente. O sistema de classificação facetada depende da automação de um sistema de recuperação que busca e recupera um componente de acordo com uma especificação facetada qualquer.

Prieto-Diaz e Freeman (1987) foram os primeiros a sugerir a possibilidade de se adotar a classificação facetada para o armazenamento e recuperação de artefatos de *software* reusável. O esquema de classificação facetada se baseia na definição de facetadas e termos para classificar e recuperar os artefatos de uma biblioteca.

O Quadro 2 ilustra seis facetadas, e seus respectivos termos, no contexto do domínio de reuso de componentes de *software*.

**QUADRO 2- Reuso de componentes de software: facetas e termos**

Funções	Objetos	Mídia	Tipo de Sistema	Área Funcional	Configuração
adicionar	argumentos	vetor	assembler	Contas a Pagar	Anuncios
incluir	vetores	buffer	geração de código	Contas a Receber	Dispositivo de reparo
fechar	retrocesso	cartão	otimização de código	Análise Estrutural	Dispositivo de restauração
comparar	brancos	disco	compilador	Auditoria	Associação
complementar	buffers	file	Gerenciamento BD	Controle de processo	Auto reparo
comprimir	caracteres	teclado	expression evaluator	Faturamento	barbearia
criar	descriptos	linha	ponteiro de arquivo	Contabilidade	Estação de Transmissão
decodificar	digitos	lista	BD hierarquico	Orçamento	central de cabeamento
deletar	diretorios	mouse	BD hibrido	Planejamento da Capacidade	negociante de carro
dividir	expressões	impressora	interpretador	CAD	catálogo de vendas
avaliar	arquivo	tela	analizador léxico	Contabilidade de custo	circulação
trocar	instruções	sensor	editor de linha	controle de custo	anuncios classificados

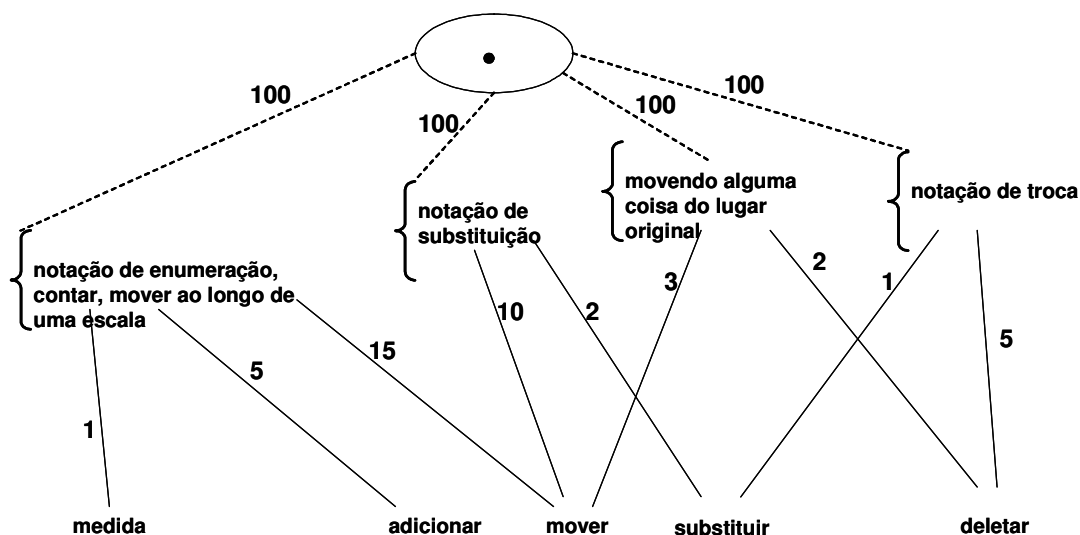
**FONTE – Prieto-Diaz; Freeman (1987)**

No esquema facetado, o usuário define a consulta para o componente que procura através da escolha de uma tupla (conjunto de termos) contendo um termo para cada faceta. Cada componente adquirido é representado nos termos de um vetor de descrição. O tamanho deste vetor corresponde ao número de facetas associadas ao domínio de aplicação. A definição de um elemento neste domínio é representada pelo vetor, que assume para cada uma de suas posições o valor correspondente a um termo associado à faceta vinculada à respectiva posição. Tais vetores de classificação são armazenados em uma tabela relacional, onde as colunas representam as facetas e as linhas denotam descrições dos componentes. A tabela de classificação pode posteriormente ser procurada para recuperar as descrições do componente. A pesquisa poderia ser feita utilizando uma linguagem de consulta relacional, tal como o SQL, por exemplo.

Na situação mais geral de busca, ocorre um casamento parcial entre os termos da tupla da consulta e das tuplas dos componentes armazenados na biblioteca. Portanto, para algumas das facetas da tupla de consulta não é possível encontrar um artefato na biblioteca indexado pelo termo solicitado na consulta para a respectiva faceta.

Cada faceta pode conter também uma métrica que avalia a distância conceitual entre seus termos. Com essa métrica pode-se determinar a distância conceitual entre artefatos, facilitando o processo de agrupamento de artefatos por similaridade. No mecanismo de busca, essa métrica avaliará a distância conceitual

entre a pergunta e os artefatos e seus atributos em cada uma das facetas. A distância conceitual, entre um termo e sua faceta, define o quanto os conceitos do respectivo termo e faceta estão próximos conceitualmente. Desta forma, é possível calcular a distância conceitual entre dois termos distintos de uma dada faceta, a partir da soma das distâncias conceituais entre cada um dos termos em relação à faceta.



**FIGURA 2 - Reuso de componentes de *software*: grafo de similaridade conceitual**

**FONTE – Prieto-Diaz; Freeman (1987)**

Na FIG. 2 são descritos as facetas, os termos e o grau de similaridade. A distância conceitual fornece uma forma de medir a similaridade entre os termos das facetas. A similaridade é calculada medindo o caminho entre os termos. Os pesos estão diretamente relacionados a distância conceitual.

A partir dos grafos de distância conceitual, definidos para cada faceta, é possível recuperar em ordem decrescente de similaridade, os artefatos que mais atendem a tupla de termos que descreve o artefato procurado na consulta do usuário.

Para evitar múltiplas classificações devido aos sinônimos das palavras, facetas e termos são escolhidos a partir de um vocabulário controlado suportado por um dicionário de sinônimos.

O foco principal do esquema facetado é a facilidade da classificação, a simplicidade da representação e armazenamento das descrições (PRIETO-DIAS, 1991). Sua principal deficiência é o alto custo de manutenção dos repositórios, realizada de forma manual (MILI, 1997).

## **2.2 Resposta a Consultas**

Resposta a Consultas trata de questões factuais, relacionadas a fatos ou eventos, onde as questões são utilizadas para substituir as *queries* baseadas em palavras-chave e as respostas no lugar de documentos.

No método proposto, as questões utilizadas pelos usuários endereçam um problema a ser solucionado, o que acarreta complexidade em relação à resposta de questões factuais que ocorre na área de resposta a consultas, pois não existe controle sobre a diferença entre os textos que descrevem os padrões e as questões que descrevem os problemas a serem solucionados por esses padrões.

Considerando que o método proposto destina-se a suportar uma interface que responde questões abertas de usuários, é relevante tecer um paralelo do método proposto com abordagens que endereçam respostas a consultas.

### **2.2.1 Resposta a Consultas e suas Áreas Correlatas**

A área de Resposta a Consultas utiliza técnicas para a recuperação de documentos candidatos e para extração de respostas. Estas técnicas são oriundas das áreas de Recuperação de Informação e das áreas de Extração de Informação. Através destas técnicas as questões são analisadas e expandidas. Os documentos são recuperados e as respostas extraídas dos documentos candidatos são ordenadas de acordo com a relevância da questão.

Para a Recuperação de Informação as *queries* não têm formato de questões e normalmente são reduzidas a um conjunto de palavras (*bag of words*) após a extração de seus radicais (*stemming*). Os usuários precisam extrair as respostas dos documentos utilizando ferramentas de busca (*search engines*) que exploram estatísticas como a ocorrência de palavras, estrutura e formato. Exemplo: *Google* faz análise de *link*.

Em um cenário típico de recuperação de informação, os usuários submetem *queries* correspondentes às suas necessidades e o sistema retorna uma lista volumosa de documentos. Os usuários então precisam achar a informação solicitada nos documentos encontrados.

A Extração de Informação identifica elementos semânticos em um corpo de texto no qual são utilizados *templates* pré-definidos a partir de textos em linguagem natural.

Cada *template* é composto por um conjunto de elementos designados por campos (RILOFF, 1994), o equivalente aos campos de uma tabela em uma base de dados. O Quadro 3 apresenta um exemplo do domínio da Extração de Informação, cujo alvo de um sistema deste gênero é analisar uma série de documentos referente ao domínio “terrorismo”. O objetivo é extrair os elementos mais relevantes que compõe o domínio “terrorismo”. Assim esses elementos poderiam ser: vítima, bomba, perpetrador, etc.

**QUADRO 3 - Exemplo de *Template*.**

**Template: Terrorismo**

Vítima

Local

Bomba

Perpetrador

Assassinato

Explosão

**Fim Template**

A captura de informação é realizada sobre os papéis chaves em eventos de interesse. Por exemplo, em um evento relacionado com um ato terrorista, papéis chaves são: perpetrador, vítima, local, etc. O preenchimento destes *templates* a partir de grandes volumes de texto resulta em informação disponível em uma base de dados estruturada, podendo ser utilizada para *queries* na base, mineração de dados e geração de resumos.

Os padrões de extração utilizados pelas técnicas de extração de informação são estáticos, ou seja, utilizam padrões pré-definidos e não padrões gerados automaticamente a partir das consultas. Na área de resposta a consultas podem ser utilizados padrões dinâmicos de extração de respostas, gerados a partir das questões dos usuários.

### **2.2.2 Características da Área de Resposta a Consultas**

Resposta a Consultas pode ser utilizada em Domínios Abertos onde os usuários submetem questões em linguagem natural, como: “Qual é o maior vulcão da Europa?” e um sistema retorna uma lista de respostas curtas, tais como: “Sobre o Monte Etna, o mais alto vulcão da Europa, se encontra a maravilhosa cidade ...”.

Os tipos de aplicações de Resposta a Consultas são divididos em tipo de informação acessada, domínio de pesquisa, tipo de domínio e multi-modal. No tipo de informação acessada encontram-se os dados estruturados (base de dados), dados semi-estruturados (*web*: HTML, XML), ou texto livre. O domínio de pesquisa abrange a *web*, coleções específicas de documentos (como por exemplo: as coleções das <sup>4</sup>TRECs), ou um único texto (avaliação de compreensão de texto). O tipo de domínio pode ser independente (domínio aberto ou específico, por exemplo: sistemas de ajuda). O tipo de aplicação multi-modal refere-se a imagens anotadas, texto de fala, etc.

---

<sup>4</sup> As *Text Retrieval Conferences* (TRECS) tiveram início em 1992, cujo propósito era avançar o estado da arte em recuperação de textos (*text retrieval*) através do provimento da infra-estrutura necessária para a avaliação em grande escala das metodologias de recuperação de textos.

As questões podem ser classificadas de acordo com o tipo de resposta ou de acordo com o tipo de solicitação (KANDO; SHIKAWA, 2004). As questões de acordo com o tipo de resposta estão relacionadas a fatos ou eventos factuais (exemplo: "Quem é o presidente do Brasil?"), questões de opiniões ou sínteses sobre um determinado assunto (exemplo: "Qual é a atitude do autor em relação ao racismo?"). As questões de acordo com o tipo de solicitação referem-se às questões sim/não (exemplo: "O resultado do plebiscito foi a favor do desarmamento?"), e questões que identificam um comando (exemplo: "Enumere as capitais brasileiras litorâneas.>").

As questões associadas a "Porque" e "Como" requerem determinar relações que caracterizam causas ou procedimentos. As questões associadas a "Qual" podem não denotar muita semântica, dificultando a identificação da entidade de interesse da questão. Exemplo: "Qual foi o mais premiado dos filmes que concorreram ao Oscar?". É mais difícil determinar que a entidade de interesse é: "filme".

Os tipos de Respostas são classificados como: Qualidade das Respostas e Construção das Respostas (KANDO, SHIKAWA, 2004). A Qualidade da Resposta abrange as respostas longas com justificativas, respostas curtas (ex: frases) e resposta exatas (entidades solicitadas). A Construção das Respostas contempla a extração de passagens dos documentos originais, a geração a partir de várias sentenças ou de vários documentos e a extração de resumos.

Os critérios para julgar a resposta são relevância, correção, concisão, completude, simplicidade e justificativa (KANDO; SHIKAWA, 2004). A relevância refere-se à resposta relacionada à questão. A correção, a uma resposta que está factualmente correta. Na concisão, a resposta não deve conter informação desnecessária. A completude está relacionada a não apresentar somente parte da resposta. E, finalmente a justificativa que deve ser composta de texto suficiente para que o leitor avalie se a resposta é adequada à questão.

A resposta pode ser avaliada como correta (resposta apropriada), não exata (resposta incompleta), sem texto que a justifique (resposta sem justificativa nos textos do domínio de pesquisa) e errada (resposta inapropriada).



## 2.3 Principais Abordagens de Resposta a Consultas

As principais abordagens de Resposta a Consultas são baseadas em Conhecimento, na *Web* e em Padrões.

A abordagem Baseada em Conhecimento utiliza uma metodologia orientada pela lingüística e a maioria dos sistemas usa uma rede léxica (geralmente a *WordNet* (2006)). A metodologia orientada pela lingüística permite determinar o tipo de resposta solicitado pela questão e recuperar pequenos trechos de documentos. É possível também encontrar entidades combinando a categoria do tipo de resposta com os extratos (*snippets*) obtidos dos documentos. Uma rede léxica permite localizar o tipo de resposta e verificar que a resposta candidata é do tipo correta. Através da rede léxica é possível expandir palavras por meio de seus sinônimos ou termos mais genéricos. A rede léxica permite obter vários sentidos possíveis de uma palavra, e remover a ambigüidade através do contexto.

A abordagem baseada na *web* utiliza técnicas estatísticas simples para filtrar as respostas adequadas às questões. A redundância de dados proporcionada pela *web* pode ser utilizada como corpus auxiliar (LIN, 2002). As respostas são extraídas utilizando padrões simples (*surface patterns*) e expressões regulares são aplicadas sobre passagens para extrair respostas (ZHANG; LEE, 2002). O acesso aos dados semi-estruturados (HTML, XML), e estruturados (bases de dados específicas: *Internet Movie Database*, *CIA World Factbook*) da *Web* são realizados através de interfaces (*wrappers*) (KATZ, 2002).

A abordagem baseada em padrões pesquisa por padrões pré-definidos de expressões textuais que caracterizam respostas possíveis para um dado tipo de questão. A presença de tais padrões nos *strings* candidatos a respostas pode ser uma evidência de uma resposta correta. A abordagem baseada em padrões utiliza categorias detalhadas de tipos de questões. Para cada especialização da questão são definidos vários tipos de padrões que caracterizam a especialização. Por exemplo, para questão “*Who*” são definidos várias especializações: “*Who-Author*”, “*Who-Famous*”, etc. Para múltiplos extratos de documentos candidatos, é verificada a presença dos padrões associados às categorias detalhadas para cada tipo de questão.

## 2.4 Atividades Típicas na Abordagem Baseada em Conhecimento

As atividades típicas da Resposta a Consultas são realizadas na seguinte sequência:

1. Análise da Questão
2. Pré-processamento da Coleção de Documentos
3. Seleção de Documentos de Resposta Candidatos
4. Análise de Documentos de Resposta Candidatos
5. Extração de Resposta
6. Geração de Resposta

### 2.4.1 Análise da Questão

A entrada da análise da questão é uma questão expressa em linguagem natural. Entretanto existem algumas técnicas para realizar esta análise.

A técnica de *tokenization* consiste em dividir o texto em uma seqüência de símbolos, em que cada símbolo é considerado uma palavra. Para tanto, remove a pontuação e caracteres especiais. Números podem ou não ser incluídos.

O *Part-of-speech* é o processo da marcação da posição das palavras em um texto correspondendo a uma parte particular do discurso, baseada em sua definição e em seu contexto. O *POS-tagging* permite a remoção de ambigüidades léxicas de cada palavra, considerando sua posição na frase.

Os passos para a análise da questão são a quebra da questão em palavras (*Tokenization*), a remoção de ambigüidades léxicas de cada palavra, considerando sua posição na frase (*PartOfSpeech-tagging* ou simplesmente *POS-tagging*), o reconhecimento de entidades compostas por mais de uma palavra (exemplo: engenharia de *software*, governo federal, etc.), a análise sintática gerando uma estrutura de representação da frase (sujeito, verbo, objeto, sintagmas nominais, sintagmas verbais, etc), os tipos de respostas (pessoa, organização, local, data,

medida, período de tempo, etc), o foco da resposta (exemplo: Quem é o presidente do Brasil?), a extração de palavras-chave da questão (exemplo: Quem foi o inventor do avião?), a remoção de ambigüidade de palavra-chave a partir do contexto da questão (estrela: astro com luz – atriz de cinema – topologia de rede) e a expansão das palavras-chave (exemplo: inventor - sinônimo: descobridor).

Em linguagem natural são muito frequentes situações em que certas sequências de palavras têm um significado diferente daquele que seria inferível a partir dos significados das partes. O processamento de linguagem natural necessita normalmente do pré-agrupamento das palavras em termos compostos. O reconhecimento de um termo composto não se limita evidentemente a uma simples comparação de “strings”, já que as próprias palavras que compõem o termo composto podem ser flexionadas. Em alguns casos é importante restringir os tipos de flexões que podem ser feitas a partir das palavras, dar a possibilidade de indicar características para o termo definido (de outro modo a sua utilidade em gramáticas seria muito limitada), permitir a herança de características de algumas das palavras que a constituem.

Há problemas no armazenamento dos termos compostos devido a ambigüidades léxicas dos termos constituintes. O reconhecimento do termo composto levanta também problemas, porque uma palavra pode ter mais que um radical diferente o que obriga a seguir simultaneamente vários caminhos diferentes na árvore de termos compostos.

Aos termos compostos são associadas categorias gramaticais e respectivas semânticas, não sendo, em grande parte dos casos possível atribuir uma categoria atômica como adjetivo ou nome comum, correspondendo-lhes por vezes categorias gramaticais de nível superior. No caso do tratamento semântico a situação é ainda mais grave, já que (por definição) o significado do termo composto nem sempre é dedutível a partir da semântica das palavras que o constituem, logo a única hipótese de tratamento semântico correto desses casos é ter a indicação explícita da semântica do termo composto.

Análise sintática é uma técnica empregada no estudo da estrutura sintática de uma língua. Ela é útil quando se pretende descrever as estruturas

sintáticas possíveis ou aceitáveis da língua ou decompor o texto em unidades sintáticas com o objetivo de compreender a maneira pela qual os elementos sintáticos são organizados na sentença. A análise sintática permite gerar uma estrutura de representação da frase (sujeito, verbo, objeto, sintagmas nominais, sintagmas verbais, etc).

A extração das palavras-chave da questão é realizada para identificar os termos relevantes para que seja possível a realização do processo de *matching* da questão com as sentenças candidatas.

A expansão das palavras-chave proporciona um aumento do vocabulário influenciando no processo de *matching*. Este vocabulário deve ser controlado, pois pode apresentar um alto custo de manutenção.

A remoção de ambigüidade de palavras-chave a partir do contexto da questão refere-se a exclusão dos termos relevantes que têm o mesmo sentido no contexto da questão.

A entrada poder ser uma questão em linguagem natural ou limitada a um subconjunto da linguagem natural (limitada em termos de vocabulário e sintaxe). O sistema pode suportar um diálogo com o usuário para refinar o contexto da questão.

A saída poderá ser um vetor de termos ponderados, pré-processados por um algoritmo de extração de radicais (*stemming*), isto se o mecanismo para a seleção de documentos for um sistema de Recuperação de Informação.

A extração de respostas dos documentos é realizada através da identificação do tipo semântico da entidade requerida pela questão do usuário (Por exemplo: pessoa, local, data, companhia, etc.), da identificação das palavras-chave da questão a ser utilizada no processo de *matching* da questão com as sentenças candidatas e da identificação de relações sintáticas ou semânticas para auxiliar o processo de *matching*.

A identificação do tipo da questão é realizada através da palavra-chave da questão. Por exemplo, “quando“ indica a data ou tempo, “onde” indica uma localização e “quem” indica uma pessoa.

Vários pesquisadores construíram hierarquias do tipo da questão com base no tipo de resposta esperado para auxiliar na identificação do tipo de questão do usuário (MOLDOVAN, 2000), (SRIHARI; LI, 2000), (HOVY, 2001). Alguns utilizaram a *WordNet*<sup>5</sup> para estender o tipo de resposta possível (HARABAGIU, 2001).

A identificação das palavras-chave da questão é feita com a extração das palavras-chave da questão e com a expansão das palavras. Os sinônimos e variantes morfológicas podem ser utilizados para expandir as palavras, assim como técnicas de expansão de *queries*, tais como expandir através das mais promissoras passagens recuperadas de uma enciclopédia a partir das palavras-chave da questão (ITTYCHERIAH; FRANZ; RATNAPARKHI, 2001).

A identificação das relações sintáticas ou semânticas compreende a análise sintática da questão com níveis variados de sofisticação, utilizados como parâmetros na Extração de Resposta. As representações semânticas capturam as dependências entre os termos (HARABAGIU, 2001) e as relações gramaticais referem-se ao sujeito-verbo-objeto (SCOTT; GAIZAUSKAS, 2001).

Uma vez que o tipo da entidade procurada foi identificado, o restante do processo de análise da questão se resume a identificar restrições adicionais que as entidades que casam com o tipo identificado devem satisfazer. Esse processo pode ser tão simples quanto extrair palavras-chave do restante da pergunta e usá-las no casamento da pergunta com as sentenças que contêm as respostas candidatas. Esse conjunto de palavras-chave também pode ser expandido através do uso de sinônimos. As restrições podem ser excludentes (ou seja, se o fragmento não satisfizer a restrição ele é eliminado da lista de respostas candidatas) ou pode servir para apenas atribuir um peso à resposta candidata, o que servirá para ordenar a lista de possíveis respostas que é retornada.

---

<sup>5</sup> A rede *WordNet* é, na verdade, uma base relacional, em que unidades lexicais do inglês, pertencentes às categorias dos substantivos, verbos, adjetivos e advérbios, estão organizadas em termos de conjuntos de sinônimos (isto é, os *synsets*), os quais expressam conceitos lexicalizados. Tais conjuntos relacionam-se entre si em função das cinco relações de sentido: antonímia, hiponímia, meronímia, acarretamento e causa. Assim, se forma uma espécie de rede a partir do significado da

## 2.4.2 Pré-processamento da Coleção de Documentos

No Pré-processamento da Coleção de Documentos, as questões que precisam ser respondidas em tempo real com base em grandes bases de textos (*gigabytes* ou *terabytes*), devem ser pré-processadas ou utilizar um sistema externo de busca (*search engine*). Este pré-processamento pode ir além da indexação de palavras-chave para rotular entidades associadas a nomes de pessoas, organizações, locais, quantidades, valores monetários, expressões de tempo, etc (*named entities*). A rotulação de termos é realizada de acordo com uma classificação definida a priori (PRAGER, 2001) e as expressões de relação são do tipo ternária (sujeito-relação-objeto) (KATZ, 1997).

## 2.4.3 Seleção de Candidatos de Documentos de Resposta

Escolhida uma forma de se pré-processar os documentos, algumas decisões precisam ser tomadas. Para a seleção de documentos de respostas, a maioria dos sistemas tem utilizado *search engines* convencionais, mas existem alguns pontos que permanecem para serem tratados. Devem-se determinar quantos dos documentos serão retornados e analisados, tratando-se de uma ferramenta de pesquisa que retorna documentos ordenados por relevância à questão. Se a ferramenta de pesquisa utilizada permitir a recuperação de trechos de documentos (ao invés dos documentos inteiros), é necessária ainda a determinação de alguns parâmetros, como o tamanho dos trechos (por exemplo: 200 caracteres), percentual de palavras-chave presentes na passagem e valorização da palavra que corresponde ao foco da questão. A utilização de um segmentador de texto baseado em tópicos para extrair trechos mais curtos dos documentos.

#### 2.4.4 Análise de Candidatos de Documentos de Resposta

Este passo não é necessário se a coleção de documentos já foi pré-processada. Sendo assim, pode ocorrer que sistema não realize esta análise.

Tipicamente os sistemas identificam pelo menos as entidades (*named entities*) definidas nas *Message Understanding Conferences (MUCs)* e realizam a quebra de sentenças (*sentence splitting*), rotulação léxica com remoção de ambigüidades (*POS tagging*), e categorização de grupos sintáticos (substantivos, verbos, frases preposicionais, etc). As entidades definidas nas *MUCs* podem ser representadas pelas classes: empresas, pessoas, locais, etc. Os sistemas podem ser estendidos para incluir outras classes (produtos, endereços, medidas) ou ainda subclasses (cidades, estados, países). A identificação de entidades compostas é realizada por mais de uma palavra (*multwords*).

Alguns sistemas vão além, realizam uma análise sintática mais completa, derivando uma estrutura sintática e mapeando em um conjunto de restrições relacionais expressa em uma linguagem lógica ou usando termos relacionais entre os termos selecionados da sentença original.

Os sistemas existentes podem fazer uso da análise sintática superficial para identificar nomes compostos e suas variantes, e reindexar e reordenar os documentos antes do *matching* contra a representação da questão (FERRET, 2001). Existem outras alternativas, como derivação de representação de dependência de sentenças e mapeamento em lógica de primeira ordem (da mesma forma com as questões) (HARABAGIU, 2001) ou a derivação de representação de sentença diretamente anotada com representação de papel semântico (HOVY, E. et al., 2001). Além destas, os sistemas podem empregar a análise sintática para mapear documentos candidatos em uma forma lógica ou quase lógica (MOLLA; BERRI; HESS, 1998; SCOTT; GAIZAUSKAS, 2001; ZAJAC, 2001).

### 2.4.5 Extração de Resposta

Os sistemas típicos têm mapeado a questão em um tipo de resposta esperado e com um conjunto de restrições ou anotações semânticas derivadas de tipos de questões. A representação da questão e dos textos de respostas candidatas são comparadas, e um conjunto de resposta candidata é produzido.

No processo de *matching* uma unidade de texto (que pode ser sentença) deve conter um *string* que combine com o tipo semântico da resposta esperada. A generalização (*subsumption*) pode ser utilizada, com o auxílio de hierarquia de termos mais gerais da *WordNet*, por exemplo.

Uma vez que o tipo esperado foi encontrado na unidade de texto candidata, outras restrições podem ser aplicadas. Estas restrições podem ser utilizadas para excluir o candidato, ou para definir sua ordenação entre os demais. No processo de *matching* os sistemas podem variar por tipo de restrição utilizada, e como as restrições são satisfeitas e ponderadas.

Para um parágrafo de resposta candidato que apresenta o tipo de resposta procurada para a questão é estabelecida uma janela de resposta em torno do candidato e várias características (*features*) quantitativas são utilizadas em uma heurística numérica ponderada para computar a ordenação da resposta candidata (MOLDOVAN, 2000). Um exemplo de uma característica quantitativa é a sobreposição de palavras entre a questão e a janela de resposta. Uma extensão a esta abordagem é a utilização de um algoritmo de aprendizado de máquinas (*machine learning*) para otimizar os pesos em uma função de *score* linear, que combina as características caracterizando a janela de resposta (HARABAGIU, 2001).

Invertendo a ordem (SRIHARI; LI, 2000), outras restrições (não do tipo da questão) são aplicadas para ordenar sentenças em segmentos de textos candidatos. Só então o tipo de resposta esperado é usado como filtro para extrair a parte apropriada das sentenças selecionadas. Para ordenar sentenças são utilizadas características tais como:

- a) Quantas palavras da questão são achadas na sentença;



b) A ordem das palavras-chave na sentença é comparada com a ordem na questão;

c) Se o verbo chave e uma variante combinam.

A combinação do tipo de resposta esperado com as medidas de comparação baseadas em palavras em uma única função de *score*, é aplicada as janelas de sentenças que se movem sobre os documentos candidatos a respostas (ITTYCHERIAH; FRANZ; RATNAPARKHI, 2001).

Os sistemas que utilizam representações mais ricas, tais como formas lógicas ou textos anotados com informação semântica e gramatical, podem utilizar restrições adicionais, expressas nestas representações, para o processo de *matching*. Estas restrições devem ser tratadas com preferências e não como mandatórias, para sacrificar muita cobertura (*recall*) em favor da precisão (*precision*). Os sistemas que utilizam estas representações mais ricas (HOVY, 2001), (SCOTT; GAIZAUSKAS, 2001), utilizam também uma estratégia de recuperação (*fallback*) quando estas restrições não são aplicáveis.

#### **2.4.6 Geração de Resposta**

Os sistemas avaliados nas *Text Retrieval Conferences (TREC)* têm retornado *strings* de 50 ou de 250 caracteres (PRAGER, 2001). Esta resposta não é adequada por vários motivos. Primeiro, o *string* pode não ter uma boa estrutura gramatical, ser de difícil leitura e apresentar falta de contexto (anáforas pendentes). Segundo, os usuários podem querer uma resposta com mais evidência e mais contexto. Terceiro, as respostas mais complexas podem demandar um diálogo com o usuário para prosseguir.

### **2.5 Extração de Relações**

Considerando que as relações encontradas entre os conceitos de um dado período de um texto são a base para a geração da Forma Gráfica de Representação do método proposto, é relevante revisar propostas na literatura que

focam a extração de relações para utilização em extração de informações a partir de textos em linguagem natural.

### 2.5.1 *Menor Caminho de Núcleos Dependentes para a Extração da Relação*

Em Bunescu e Mooney (2005), a extração da relação é tipicamente capturada através do menor caminho entre as duas entidades em um gráfico de dependência. Exemplo: "*Protesters seized several pumping stations, holding 127 Shell workers hostage.*"

O objetivo é exclusivamente a extração das relações entre tipos pré-definidos de entidades em um corpus de Extração de Conteúdo Automatizada (ACE).

O corpus utilizado contém co-referência, nomes de entidades e relações. As entidades podem ser do tipo *person, organization, facility, location e geo-political*. Existem 5 tipos gerais de relações: *role, part, located, near e social*. A relação do tipo *role* liga pessoas a uma organização a qual pertencem ou fornecem algum serviço. A relação do tipo *part* indica o subconjunto de relacionamentos, tais como um estado a uma nação ou uma subsidiária para a sua matriz. A relação do tipo *located* indica a posição de uma pessoa ou de uma organização em alguma posição. A relação do tipo *near* indica a proximidade de uma posição a outra. A relação do tipo *social* indica as relações profissionais, familiar ou pessoal entre duas pessoas.

A relação deve ser extraída somente entre as entidades mencionadas na mesma sentença. Os argumentos são conectados a seus predicados alvos diretamente através de uma arco que aponta para o predicado ("*troops* → *raided*"), ou indiretamente através de uma parte da preposição ou infinitivo ("*warning* ← *to* ← *stop*"). Exemplo: "*Troops recently have raided churches, warning ministers to stop preaching.*" Outros tipos de dependências dizem respeito aos relacionamentos modificador-núcleo presente em adjetivo-substantivo ("*several* → *stations*"), substantivo-substantivo ("*pumping* → *stations*"), ou construções advérbio-verbo ("*recently* → *raided*").

Um *parser*, *Context Free Grammar* (CFG) pode ser utilizado para extrair as dependências como predicado-argumento (exemplo: “*troops* → *raided*”) ou núcleo modificador (exemplo: “*pumping* → *stations*”). É utilizado um sufixo adicional “(-)” para os verbos e substantivos que ocorrem ao longo de um caminho de dependência. Este sufixo adicional indica itens com polaridade negativa. Exemplo: a frase “Ele nunca foi para Paris” está associada com o seguinte caminho “Ele → foi(-) ←para ←Paris.

### 2.5.2 Método de Núcleo para a Extração da Relação

Zelenko (2003), tem como objetivo extrair relações de um texto em linguagem natural. Utiliza um exemplo de treino (corpus) para as experiências que compreende 200 diferentes artigos de notícias e publicações (*Associated Press, Wall Street Journal, Washington Post, Los Angeles Times, Philadelphia Inquirer*). Os padrões para identificar as relações são determinados através de um conjunto de relações já extraídas manualmente.

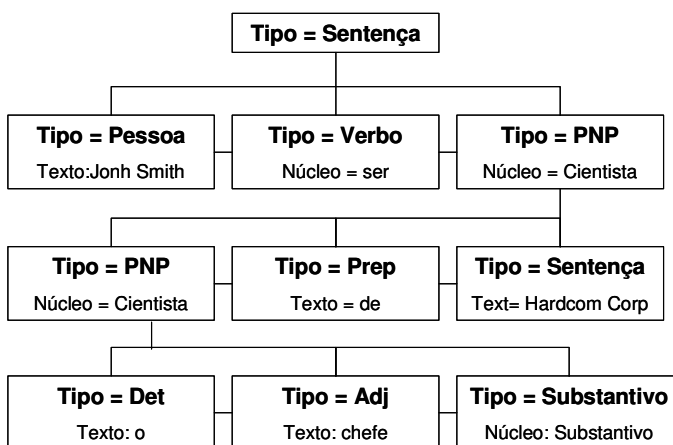
É utilizado para a extração da relação e das entidades um <sup>6</sup>*shallow parsing* (*parse* parcial) construído manualmente por um sistema de extração de informação. O *parse* gera uma representação de uma sentença e identifica nomes, substantivos e um conjunto de *part of speech* no texto. São também classificados substantivos e nomes referentes a pessoas, organizações e localizações produzindo assim as entidades. A análise gramatical é convertida em forma de árvore dentro de alguns exemplos para a relação pessoa-afiliação. Este tipo de relação faz uma associação entre uma pessoa e uma organização. O método extrai do texto relações pessoa-afiliação e organização-localização.

A FIG. 3 mostra uma representação da análise gramatical produzida pelo *shallow parsing* da sentença “*John Smith is th chief scientist of th Hardcom*

---

<sup>6</sup> *Shallow parsing* é uma técnica do processamento da linguagem natural que tenta fornecer de forma automática o entendimento da estrutura de uma sentença, mas sem analisar toda sua estrutura gramatical (interna) dentro de uma árvore gramatical. É chamado de análise parcial onde o texto é dividido em um série de palavras que juntas constituem uma unidade gramatical (exemplo, substantivo, verbo , preposição) (DERSHOWITZ, 2006).

*Corporation* . Os tipos “PNP”, “Det”, “Adj” and “Prp” denotam “Sintagma Nominal” (Personal Noun Phrase), “Determinador”, “Adjetivo” e “Preposição” , respectivamente.



**FIGURA 3 - Representação análise gramatical**

**FONTE - ZELENKO (2003).**

A análise gramatical é convertida em forma de árvore dentro de alguns exemplos para a relação pessoa-afiliação. Este tipo de relação faz uma associação entre uma pessoa e uma organização. O método extrai do texto relações pessoa-afiliação e organização-localização. É criado um exemplo para a relação pessoa-afiliação através dos nós pessoa e organização na árvore gramatical com atributos para os nós, especificando o papel que um nó desempenha em uma relação pessoa-afiliação. Veja a FIG. 4.

A pessoa e a organização receberão os papéis de membro e afiliação, respectivamente. O resto dos nós não irá receber nenhum papel refletindo que eles não participam da relação. Cada nó pode ter diferentes números de atributos. Os atributos são nomeados, e cada nó tem necessariamente atributos com nomes de “*Tipo*” e “*Papel*”.

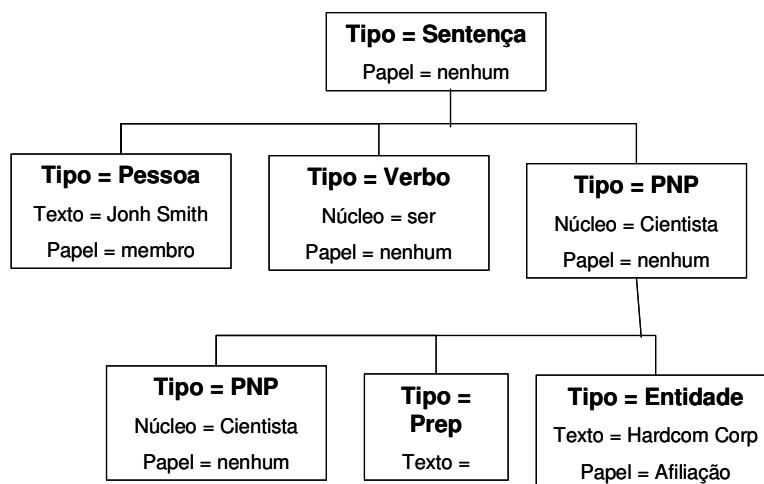


FIGURA 4 - Exemplo pessoa-afiliação gerado pelo *shallow parse*.

FONTE - ZELENGO (2003).

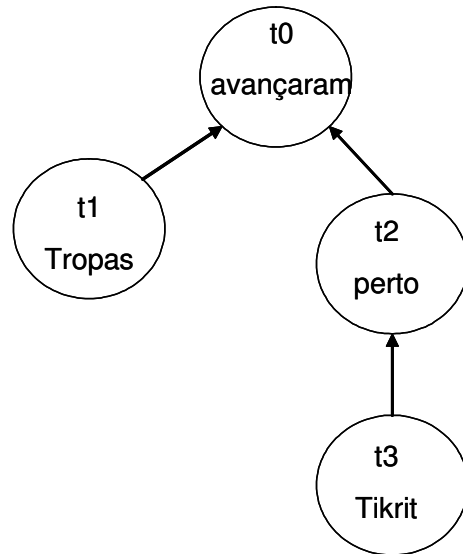
### 2.5.3 Núcleo de Árvore Dependente para a Extração da Relação

Culotta e Sorensen (2003) é uma extensão de Zelenko (2003) que tem como objetivo estimar a similaridade entre as dependências de árvores de sentenças. Uma árvore de dependência é uma representação que denota relações gramaticais entre palavras em uma sentença. A tarefa é identificar e classificar relações e entidades no texto. O <sup>7</sup>SVM é utilizado com o núcleo das árvores para detectar as relações entre as entidades em um corpus. É feita uma conversão da árvore sintática/gramatical para árvore de dependência. Um conjunto de regras mapeia e analisa a árvore núcleo para uma árvore de dependência. Por exemplo: sujeitos são dependentes de seus verbos e os adjetivos são dependentes dos substantivos que modificam. Não foi considerado o relacionamento de rótulos

---

<sup>7</sup> *Support Vector Machine (SVM)* é uma técnica para resolver problemas de reconhecimento de padrões. No SVM, os padrões de entrada são transformados para um vetor de características de alta dimensionalidade, cujo objetivo é separar as características linearmente no espaço. Uma vez que o espaço adequado de características é definido, o SVM seleciona o hiperplano particular, chamado de hiperplano de margem máxima (MMH), o qual corresponde à maior distância de seus padrões no conjunto de treinamento (BARANOSKI, 2005).

(títulos) (exemplo: "objeto", "sujeito"); ao invés disto foi somente usado a estrutura da dependência.



**FIGURA 5 -** Árvore de dependência para a frase: **As tropas avançaram perto de Tikrit.**

**FONTE -** CULOTTA; SORENSEN (2003).

Os métodos de núcleo são as técnicas que calculam uma função de núcleo entre instâncias (exemplos) de dados onde a função pode ser calculada através de uma medida de similaridade. Dado um conjunto de instâncias rotuladas, o método de núcleo determina o rótulo comparando cada instância aos exemplos de treino (corpus) utilizando esta função de núcleo. A tarefa da função de núcleo é encontrar as relações similares. É definido uma árvore núcleo sobre as árvores de dependência e este núcleo é incorporado como um SVM para extrair relações dos documentos. A FIG. 5 apresenta uma árvore de dependência para a frase: "Troops advanced near Tikrit" ("As tropas avançaram perto de Tikrit").

#### **2.5.4 REES: Sistema de Extração de Relações e Eventos em Larga Escala**

REES (AONE; RAMOS-SANTACRUZ, 2000) faz uso de *templates* onde as "Relações" cobrem o que está no MUC-7 e são chamadas de Elementos do *Template* e Relações do *Template* (TEs e TRs), sendo que existem 39 tipos de

relações. Enquanto os TEs extraem somente entidades no singular o REES extrai entidades no singular e no plural. Possui um rotulador que reconhece o nome de pessoas, organizações, lugares e artefatos (somente veículos). Atualmente o REES trabalha somente com eventos baseados em verbos. Foi desenvolvida uma ontologia das relações e dos eventos a serem extraídos. Esta ontologia representa uma variedade de domínios: político, financeiro, negócio, forças armadas e eventos e relações do cotidiano.

REES faz o tratamento da co-referência baseado em regra que resolve: substantivo de frases, de organização, pessoa e tipos de localização; Pronomes de pessoas no singular: “ele” e “ela”.

O usuário pode executar o REES através de uma interface gráfica chamada “*TemplateTool*”. O QUADRO 4 mostra um exemplo de *template* para a relação pessoa-afiliação.

**QUADRO 4 - *Template* da relação Pessoa-Afiliação para “Frank Ashley, a spokesman for Occidental Petroleum Corp.”**

<p>&lt;Pessoa_Afiliação_AP8802230207-54&gt;:=          Tipo: Pessoa_Afiliação          Pessoa: [TE para “Frank Ashley”]          Org: [TE para “Occidental Petroleum”]</p>
--

**FONTE - AONE; RAMOS-SANTACRUZ (2000).**

Como primeiro passo na construção de um sistema de extração de evento de relação em larga escala foi desenvolvida uma ontologia das relações e eventos a serem extraídos. O QUADRO 5 apresenta as ontologias das relações e o QUADRO 6 a ontologia dos eventos.

QUADRO 5 - Ontologia das Relações

Relações	
Local-Relações	Artefato-Relações
Local-Nome & Apelido	Artefato-Nome & Apelido
Local-Tipo	Artefato-Tipo
Local-Subtipo	Artefato-Subtipo
Place-Descriptor	Artefato-Descriptor
Local-Pais	Artefato-Criador
	Artefato-Proprietário
Organization-Relations	Person-Relations
Org-Nome&Apelido	Pessoa-Nome & Apelido
Org-Descriptor	Pessoa-Tipo
Org-FoundationDate	Pessoa-Subtipo
Org-Nacionalidade	Pessoa-Descriptor
Org-TicketSimbol	Person-Honorific
Org-Localização	Pessoa-Idade
Org-ParentOrg	Pessoa-NumeroTelefonico
Org-Proprietário	Pessoa-Nacionalidade
Org-Founder	Pessoa-Afiliação
Org-StockMarket	Person-Sibling
	Person-Esposo
	Person-Pais
	Person-Avós

FONTE - AONE; RAMOS-SANTACRUZ (2000).

QUADRO 6 - Ontologia dos eventos

Eventos	
Veículo	Transação
Departamento de Veículo	Comprar Artefato
Entrada de Veículo	Vender Artefato
Lancamento Nave Espacial	Importar Artefato
	Exportar Artefato
Mudança Pessoal	Negócios
Despedido	Iniciar Negócio
Termino do contrato	Encerrar Negócio
Promovido	Adquirir companhia
	Vendar companhia
	Juntar companhia

FONTE - AONE; RAMOS-SANTACRUZ (2000).

Os eventos são extraídos junto com seus eventos participantes, por exemplo "Who did what to whom when and where?". Para o evento "COMPRA", por exemplo, o REES extrai o comprador, o artefato, o vendedor, o tempo e a localização do evento "COMPRA". Atualmente REES cobre 61 tipos de eventos. Alguns deles são mostrados no QUADRO 6.



### 2.5.5 TREX: *Framework* Flexível para Extração da Relação

Em T-REX (IRIA, 2004) é apresentado um *framework* de extração de relação que utiliza um algoritmo da extração da entidade baseado em *Support Vector Machine* e combina a estrutura de um corpus de treino com uma ontologia. A atividade da extração da relação (REX), também é conhecida como a extração de um evento ou *template* que estabelece relações entre as entidades classificadas. É estudado o uso de uma representação gráfica para a extração da relação. Esta representação gráfica oferece vantagens como permitir expressar de uma maneira uniforme as relações entre os sub-gráficos. Por exemplo, relações de co-referência, relações gramaticais, relações relacionadas ao formato HTML, e mesmo as anotações das relações no texto. Adicionalmente permite a prototipagem rápida de novos algoritmos uma vez que todas as características podem ser capturadas em uma representação e tratadas uniformemente.

A FIG. 6 mostra a representação de um corpus para algumas informações (dados) do corpus. O objetivo do algoritmo de extração da relação é a identificação de algumas características na representação da relação entre entidades. Bordas com setas representa uma relação usual “*isa*” (“é um”), as bordas tracejadas a relação “*instance\_of*” (“composto\_de”). A representação do corpus está a esquerda, enquanto que a direita é definida a ontologia que define a estrutura da representação pretendida pelo corpus. A sentença original é “*Profits soared at Boeing Co., easily topping forecasts on Wall Street, as their CEO Alan Mulally announced first quarter results.*” (“Lucros da Boeing CO., cobrindo previsões em *Wall Street*, como seu CEO Alan Mulally anunciou”). O modelo de dados é instanciado para um corpus particular com referência ao modelo da ontologia que fornece a estrutura para a representação pretendida. Ele compreende as entidades específicas como “*companhia*” e “*pessoa*” como também as entidades sintáticas/gramaticais tais como “*frase*” e “*palavra*”. Além das entidades, as relações binárias podem ser modeladas como mostrado pelas relações “*próximo*” e “*adjacente*”. O objetivo do algoritmo de extração da relação é a inferência de algumas características na representação da relação “*tem\_CEO*” entre a companhia “Boeing” CO.” e a pessoa “Alan Mulaly”.

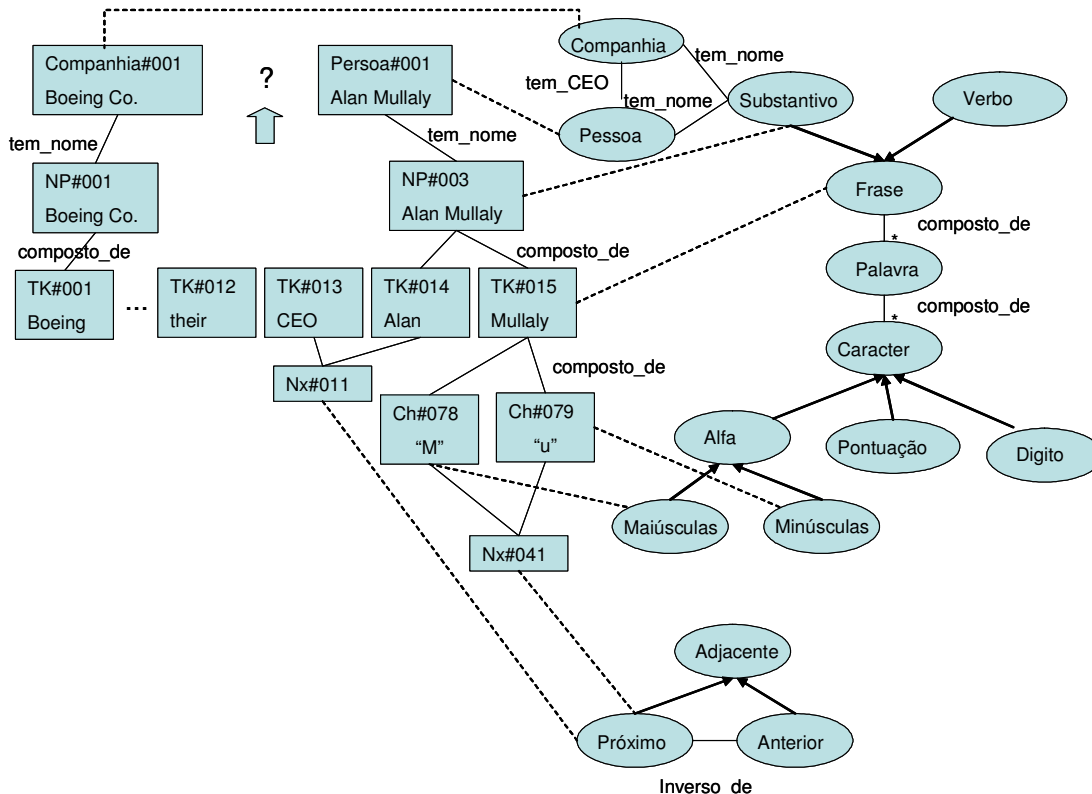


FIGURA 6 - Exemplo Simplificado de uma proposta de representação do "Corpus".

FONTE - IRIA (2004).

## 2.6 Contextualização do Método Proposto

Nesta seção são apresentadas as principais diferenças e similaridades do método proposto com as propostas encontradas na literatura.

A recuperação de informação utiliza *queries* e não questões. As *queries* não têm formato de questões e normalmente são reduzidas a um conjunto de palavras após a extração de seus radicais. A extração de informação captura elementos relevantes de eventos onde são utilizados *templates* pré-definidos a partir de textos em linguagem natural. A captura de informação é realizada sobre os papéis chaves em eventos de interesse.

Resposta a Consultas utiliza-se de técnicas para a recuperação de documentos candidatos e para extração de respostas. As questões são utilizadas para substituir as *queries* baseadas em palavras-chave e as respostas no lugar de documentos. Resposta a Consultas trata de questões factuais, relacionadas a fatos ou eventos, onde o problema escolhido demanda questões complexas.

Para o método proposto, a indicação do padrão de projeto mais adequado para um determinado problema é a resposta à questão. Com base em algumas características do padrão de projeto e na gramática da língua portuguesa são criadas algumas regras ou heurísticas para a extração dos termos relevantes. Estas regras levam em consideração não só as palavras-chave mas também o relacionamento entre elas. As questões são realizadas pelos usuários em linguagem natural. As heurísticas de extração são aplicadas às questões e o casamento entre os termos é realizado, indicando os padrões mais adequados.

O foco principal da classificação facetada é a facilidade da classificação do artefato, simplesmente da representação e do armazenamento dos descritores do artefato, da uniformidade de atributos da classificação e da facilidade da automatização. A principal deficiência da classificação facetada adotada em muitos sistemas existentes, é o alto custo de manutenção dos repositórios e grafos de distância conceitual, que é feita predominantemente de forma manual. A utilização da linguagem natural é mais simples e transparente para o usuário ao uso de facetas. Basta que o usuário escreva a questão em linguagem corrente.

O modelo proposto se baseia na definição de uma representação que captura a sintaxe e a semântica de padrões de projetos armazenados na biblioteca corporativa com o objetivo de suportar a indicação de padrões de projetos que atendam um dado problema descrito pelo usuário. A consulta de padrões de projeto em uma biblioteca corporativa deve indicar, em ordem de prioridade, os padrões mais adequados à questão do usuário.

O esquema proposto permite a construção de uma ontologia específica, pois o domínio escolhido é muito restrito se comparado a Resposta a Consultas onde podem ser utilizadas em Domínios Abertos. Esta ontologia permite suportar a expansão de termos das questões dos usuários. Em um processo usual de resposta

a questões, baseado em uma grande quantidade de informação textual, como a fornecida pela *Web* por exemplo, não seria viável a determinação da similaridade entre conceitos da questão e dos textos disponíveis para busca.

A abordagem baseada na *Web* utiliza técnicas estatísticas simples para filtrar as respostas adequadas às questões. A redundância de dados proporcionada pela *Web* pode ser utilizada como corpus auxiliar (LIN, 2002). O modelo proposto não faz uso de um corpus treinado ou auxiliar para extrair as respostas ou padrões de projetos adequados. No método proposto as frases dos usuários ou questões são analisadas por meio de um analisador léxico e sintático. Os termos relevantes e seus relacionamentos são extraídos com base nas heurísticas de representação e na estrutura de árvore fornecida pelo analisador sintático (*parser*). O casamento entre os termos é realizado, indicando os padrões mais adequados. Diferentemente do modelo proposto na abordagem baseada na *Web* as respostas são extraídas utilizando padrões simples (*surface patterns*) e expressões regulares são aplicadas sobre passagens para extrair respostas (ZHANG; LEE, 2002).

Assim como a abordagem baseada em conhecimento, o método proposto faz uso de uma rede léxica. Os termos relevantes são expandidos por meio de seus sinônimos ou termos mais genéricos e é possível obter vários sentidos possíveis de uma palavra e remover as ambigüidades através do contexto.

O método proposto utiliza-se de padrões de forma similar à abordagem baseada em padrões. A diferença está na forma como os padrões são criados. O modelo proposto utiliza os padrões da gramática da língua portuguesa (utilizada no Brasil) que são rotulados através de um analisador sintático e léxico (*parser - tagger*). As regras de representação e de busca são criadas a partir da gramática de língua portuguesa e da estrutura de árvore fornecida pelo *parser*. Não são definidos padrões que caracterizam a especialização ou generalização.

No que se refere a análise da questão o método proposto utiliza quebra da questão em palavras (*tokenization*). Neste processo toda a estrutura gramatical é extraída da questão através de uma análise sintática. Esta análise sintática é realizada com o auxílio de um analisador sintático desenvolvido por (BICK, 2000) onde é gerada uma estrutura de representação da frase (sujeito, verbo, objeto,

sintagmas nominais, sintagmas verbais, etc). No modelo proposto as palavras são identificadas através de objetos, eventos e entidades. Os relacionamentos entre estas palavras são também identificados no modelo proposto normalmente pelos verbos.

A utilização da linguagem natural dificulta o processo de *matching* devido às ambigüidades da língua. A remoção de ambigüidade da palavra-chave a partir do contexto da questão e dos padrões de projetos é por intermédio da expansão da palavra-chave. Se a palavra-chave pode ser expandida por intermédio de um sinônimo ou uma derivação dentro do contexto dos padrões de projetos, esta palavra é utilizada. Se para a palavra-chave não for cabível uma expansão esta palavra é descartada. Esta remoção pode ser realizada de forma manual no momento da identificação da palavra-chave.

A técnica de *stemming* é utilizada para extrair a origem morfológica das palavras. Converte palavras no plural para singular e formas verbais para o infinitivo, entre outras mudanças. Em mecanismos de busca genéricos, o *stemming* tem problemas devido a dependência do idioma e a ambigüidade das palavras que possibilitam a geração de mais de um radical. O método proposto faz uso da conversão das palavras no plural para singular e das formas verbais para o infinitivo.

Para a extração das respostas, tanto a identificação das palavras-chave quanto a identificação de relações sintáticas é abordada pelo método proposto. No processo de *matching* proposto os termos são comparados de forma direta. O casamento entre a frase inserida pelo usuário e a representação na biblioteca, é realizado somente se os termos relevantes entre as duas frases forem iguais. Não é considerada a sequência em que os termos estão relacionados. Isto significa que o casamento é feito através de padrões similares. Como este tipo de casamento é muito restritivo e gera problemas de vocabulário, os termos são expandidos através das similaridades que cria uma relação entre os termos relevantes expandindo o vocabulário.

Diferentemente do modelo proposto, Culotta e Sorensen (2003) utiliza uma representação que denota relações gramaticais entre palavras em uma sentença e estima as similaridades entre as dependências de árvores de sentenças.

REES (AONE; RAMOS-SANTACRUZ, 2000) faz uso de *templates* e trabalha com evento baseado no verbo para a extração das relações. Bunescu e Mooney (2005) utilizam tipos pré-definidos de entidades e relações. A extração da relação é tipicamente capturada através do menor caminho entre as duas entidades em um gráfico de dependência. Zelenko (2003) utiliza um *shallow parsing* (parse raso ou parcial) para produzir as entidades e trabalha com relações pré-definidas (exemplo: pessoa-organização) onde a análise gramatical é convertida em forma de árvore gramatical. T-Rex (IRIA, 2004) combina a estrutura de um corpus de treino com uma ontologia. Utiliza uma representação gráfica para a extração da relação onde as características podem ser capturadas em uma representação e tratadas uniformemente.

Observa-se ainda que, as propostas analisadas nesta dissertação têm uma característica em comum. Todas utilizam um corpus de treino, o que difere do método proposto.

O modelo proposto faz uso de uma representação gráfica para as relações assim como em T-Rex (IRIA, 2004). A diferença está na forma como esta representação é gerada. No modelo proposto é utilizado um analisador sintático fornecido por (BICK, 2000) onde toda a estrutura gramatical é analisada.

Assim como em Bunescu e Mooney (2005), o modelo proposto trabalha com polaridade negativa, mas não faz uso de um sufixo. No método proposto a polaridade negativa é representada como por exemplo, pelas palavras “não” ou “nunca” que são rotuladas pelo analisador sintático e representadas em uma Forma Gráfica de Representação (FGR). O modelo proposto identifica ainda um outro tipo de polaridade negativa. A palavra “sem” que tem a conotação de “não” é convertida em um processo de redução da Forma Gráfica de Representação.

## **2.7 Conceitos e Ferramenta Utilizados para a Representação e Busca dos Padrões de Projeto**

Esta seção apresenta alguns conceitos que serão utilizados no processo de descrição e busca dos padrões de projeto:

- a) Lematização;
- b) Co-referência;
- c) Ontologia;
- d) Grau de similaridade entre conceitos;
- e) Fator de *matching* entre textos.

Esta seção descreve também uma ferramenta externa (BICK, 2000) que será utilizada para gerar a estrutura de representação das frases dos textos dos padrões de projeto na biblioteca.

Adicionalmente, a conceituação de período e de oração é apresentada a seguir, para utilização no decorrer do texto.

Período é uma oração ou uma reunião de orações que formam um sentido completo. Um período em um texto é delimitado por símbolos gramaticais tais como: ponto final, ponto de exclamação ou interrogação.

Uma oração é composta de um predicado e opcionalmente de um sujeito. Uma oração pode ser absoluta, se for a única oração do período. Uma oração pode ser coordenada quando vem ligada a outra oração de mesma natureza em um dado período. Uma oração pode ser subordinada, quando seu sentido depende de uma outra oração no mesmo período.

### **2.7.1 Lematização**

A lematização é o processo de extração do lema de uma palavra. O lema é a forma canônica de todas as formas que representam um mesmo significado. Esta forma canônica é muito utilizada para determinar o singular de palavras no plural, bem como o infinitivo associado à conjugação de verbos.

Um conceito relacionado à lematização é o conceito de *stemming* que obtém a parte principal de uma palavra (radical), extraindo seu afixos. Observe, que os verbos regulares têm lema e radical coincidentes, enquanto os irregulares não.

A abrangência do conceito de lematização utilizada neste trabalho é restrita aos seguintes casos:

- a) infinitivo das conjugações de verbos regulares e irregulares;
- b) forma única para palavras com variantes de gênero, número e grau.

### 2.7.2 Co-referência

A co-referência é a característica de certos itens lingüísticos que ao invés de serem interpretados por seus traços semânticos ou por um “conhecimento identificador”, fazem referência a outro item a ser encontrado no próprio texto para a sua interpretação. Observe, por exemplo o seguinte texto: “O presidente desembarcou esta manhã em São Paulo. Ele assinou um importante decreto na presença do governador.” Neste texto, o termo “Ele” se refere a “O presidente”.

Os elementos de co-referência são palavras que se referem a uma mesma realidade lingüística, garantindo a manutenção e a progressão das idéias em um texto (exemplos: sinônimos, pronomes, expressões definidas).

O tratamento de co-referência é utilizado pelo método proposto, como um método externo, porque a complexidade de solução justifica um trabalho a parte.

### 2.7.3 Ontologia

Uma ontologia relaciona conceitos, de um dado domínio do conhecimento, através de relações do tipo: abstração/especialização, agregação e similaridade. A ontologia pode ser representada como uma rede semântica e, portanto, visualizada como um grafo no qual os nós são os conceitos e os arcos são as relações. Uma ontologia pode ter nós associados a palavras compostas, como por exemplo: “engenharia de *software*”.

A construção de uma ontologia, para o domínio de padrões de projeto, não é tratada neste trabalho. O trabalho pressupõe que termos de um texto podem ser ampliados por outros termos, similares ou mais genéricos, obtidos a partir de



uma ontologia externa, construída para padrões de projeto. Ampliar um termo significa, por exemplo, considerar o termo A equivalente ao termo B, se existir na ontologia que suporta o processo de *matching* entre textos, um termo B similar ou mais genérico associado ao termo A.

#### 2.7.4 Ferramenta de Rotulação Sintática e Semântica

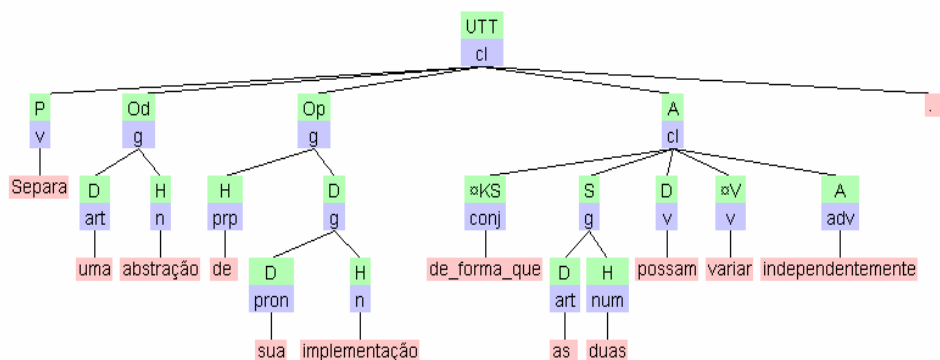
A estrutura de representação dos textos dos Padrões de Projetos que são armazenados na biblioteca, para responder as consultas formuladas pelos usuários utiliza uma ferramenta externa de rotulação sintática e léxica denominada “Palavras”.

O VISL (*Visual Interactive Syntax Learning*) é um sistema eletrônico de ensino para a gramática portuguesa, que tem como base um analisador léxico e sintático (*tagger-parser*) para Português (BICK, 2000). O sistema se apóia em um léxico de 50.000 lemas e milhares de regras gramaticais para fornecer uma análise completa, tanto morfológica como sintática de um texto qualquer. O autor do *parser* informa que esta ferramenta alcança um nível de correção de 99% em termos de morfologia e de 97 a 98% em termos de sintaxe. O *parser* “Palavras”, será denominado a partir deste ponto simplesmente por *parser*.

O *parser* gera uma representação, em forma de árvore para um período de um texto, ou para trechos de um período delimitados por ponto e vírgula ou por dois pontos. Esta representação, denominada Árvore Sintática (AS), será referenciada ao longo do texto por sua sigla AS.

A FIG. 7 ilustra um exemplo de geração de uma AS para o texto da faceta “Propósito” do padrão de projeto “*Bridge*”.

- **Padrão:** *Bridge*
- **Propósito:** Separa uma abstração de sua implementação de forma que as duas possam variar independentemente.



**FIGURA 7 - Representação da Árvore Sintática de um texto.**

**FONTE - BICK (2000).**

Na representação da AS, cada nó da árvore é rotulado pela sua caracterização sintática em verde e léxica em azul. Para a análise sintática são utilizados, por exemplo, os rótulos: Od (Objeto Direto), A (Complemento Adverbial), S (Sujeito), P (Predicado Verbal), etc. Para a análise léxica são utilizados, por exemplo, os rótulos: prp (Preposição), adv (Advérbio), art (Artigo), conj (Conjunção), etc.

O Anexo E “Notação Léxica utilizada pelo *Parser*” e o Anexo F “Notação Sintática utilizada pelo *Parser*” apresentam a notação léxica e sintática utilizada na AS do *parser*.

### **3 DETALHAMENTO DO MÉTODO PROPOSTO**

Neste capítulo é detalhada a concepção do método proposto. É definida uma notação para uma forma de representação gráfica da estrutura gramatical dos textos que descrevem os propósitos dos padrões de projeto. Um guia de sugestões é elaborado para auxiliar a redação da descrição dos padrões de projeto, geradas pelos bibliotecários da biblioteca corporativa, e para auxiliar a elaboração das questões, geradas pelos usuários da biblioteca.

Ainda neste capítulo é gerada uma forma gráfica de representação, utilizando a notação proposta a partir de um mapeamento automático, entre a estrutura gramatical gerada por um parser externo e a representação gráfica dos textos. São identificados refinamentos para as formas gráficas de representação gerados automaticamente, possibilitando a remoção de termos desnecessários.

Finalmente, é definido um algoritmo de matching, entre as formas de representações gráficas das descrições dos propósitos dos padrões de projeto, e das questões dos usuários.

#### **3.1 A Escolha da Faceta Propósito para o Experimento da Dissertação**

Das quatro facetas selecionadas para descrever cada padrão de projeto na biblioteca (ver seção 1.3), a faceta “Propósito” dos padrões de projeto foi a escolhida para o experimento deste trabalho pelas seguintes razões: foi possível achar na literatura questões associadas a esta faceta; o material encontrado apresenta uma boa diversidade para a análise; e utilizar o processo manual para trabalhar também nos textos das outras duas facetas seria muito trabalhoso.

A alternativa encontrada, para obter as questões necessárias para o experimento do trabalho, foi buscar na literatura questões associadas às facetas escolhidas. Esta busca foi bem sucedida em relação à faceta “Propósito”. Utilizar questões geradas pelos autores do trabalho, não seria adequado para validar o trabalho, porque não configuraria um grupo de questões isento, podendo ser tendencioso em relação à solução proposta.

O material encontrado para análise, proporcionado pelas facetas “Propósito” dos 23 padrões de projeto propostos na literatura (GAMMA, 1995) e de questões associadas a estes padrões de projeto, apresenta uma diversidade suficiente para a geração da primeira versão do método proposto.

O experimento foi realizado manualmente, pois ainda não foram construídas ferramentas para automatizar o método proposto. Desta forma, a ampliação do experimento, para considerar o texto das outras duas facetas, fica postergada para um trabalho futuro; se possível, respaldado por uma ferramenta construída para automatizar o método proposto.

Definido que o experimento deste trabalho será baseado na faceta “Propósito” dos padrões de projeto, o tipo de questões do usuário de interesse para o experimento também fica determinado. As questões relacionadas com o objetivo do problema a ser solucionado são as questões que endereçam a faceta “Propósito” (ver seção 1.3). Portanto, o foco do experimento deste trabalho é a definição de um método que permita determinar, em ordem de prioridade, quais os padrões de projeto cujo propósito melhor atende às questões do usuário que endereçam o objetivo do problema a ser solucionado.

### **3.2 Critérios para a Concepção do Método e Análise de suas Conseqüências**

Dois critérios norteiam a concepção do método proposto: a utilização de uma abordagem que explore a estrutura gramatical dos textos utilizados para descrever a faceta escolhida e as questões dos usuários; e a definição de um método com alto grau de automação.

Observe que o critério de explorar a estrutura gramatical dos textos, em vez de utilizar palavras isoladas, aumenta consideravelmente o grau de dificuldade de se atingir um processo totalmente automatizado. Esta dificuldade decorre principalmente dos seguintes fatores:

a) a linguagem natural é ambígua, o que dificulta a obtenção da estrutura gramatical dos textos pela utilização de um analisador sintático (*parser*);

b) *parsers* têm suas limitações na captura da estrutura gramatical de um texto, diferentemente do ser humano que utiliza informações do domínio de aplicações para reconhecer a estrutura gramatical de um texto;

c) o bibliotecário pode formular mal os textos que representam as facetas dos padrões de projetos; da mesma forma o usuário da biblioteca em relação às questões.

A ambigüidade da gramática pode dificultar, por exemplo, que um *parser* reconheça se uma dada preposição está iniciando um objeto indireto ou faz parte de um sintagma nominal (trechos de palavras que podem ocupar o lugar de um sujeito, de um objeto, etc). A determinação da co-referência (a que entidade um dado pronome se refere) também pode ser ambígua, até para um ser humano.

O ser humano utiliza o seu conhecimento do domínio de aplicação para determinar qual a valência do verbo (transitivo direto, indireto, intransitivo) mais adequada para o contexto. O uso de uma ontologia permite ampliar o léxico utilizado (utilizando o termo original, acrescido de seus termos similares ou mais abstratos), mas é pouco efetivo para determinar a valência do verbo adequado para o contexto.

Contudo a questão mais crítica é a má formulação das descrições das facetas dos padrões de projetos e das questões de usuários. A formulação dos textos pode estar inadequada pela utilização de termos ou de trechos de texto irrelevantes para o objetivo da descrição, ou até por uma captura equivocada da definição da faceta ou da questão do usuário. Obviamente, textos mal formulados comprometem irremediavelmente a qualidade da utilização da biblioteca corporativa de componentes.

### **3.3 Definição de uma Solução para Atender os Critérios Definidos**

A solução adotada para tratar os problemas expostos na seção anterior se baseia na definição de um guia com sugestões para uma formulação de textos, considerando o uso de uma interface em linguagem natural e interagindo com o bibliotecário e com o usuário da biblioteca, para realimentar de forma amigável a formulação dos textos das facetas e das questões.

O bibliotecário e o usuário da biblioteca corporativa são pessoas de uma organização, que compõem um grupo mais seletivo de pessoas, em relação, por exemplo, ao público heterogêneo que acessa um site de uma dada empresa. A proposta de utilização de um guia de sugestões de formulação de textos para facetas e para questões faz sentido para este público seletivo. Este guia deve não só orientar a captura da essência da definição da faceta e das questões, bem como facilitar o trabalho do *parser*, sem desconsiderar o uso de uma interface em linguagem natural.

A geração automática de uma representação gráfica para os períodos dos textos de facetas e questões, a partir da árvore sintática gerada pelo *parser*, permite aos usuários uma realimentação a respeito do texto formulado. O usuário pode alterar o texto até que a representação gráfica correspondente, gerada automaticamente a partir da estrutura sintática gerada pelo *parser*, atenda o objetivo dos usuários. Desta forma, são atingidos dois objetivos:

- a) a captura adequada da essência da definição da faceta do padrão de projeto, ou da questão do usuário;
- b) o reconhecimento correto, pelo *parser*, da estrutura gramatical dos períodos dos textos das facetas e das questões.

A conclusão, até o presente momento, é que a solução de orientar através de um guia e de suportar através de uma visualização gráfica, a geração da descrição das facetas e das questões é uma excelente solução de compromisso. Esta alternativa é plenamente justificada porque auxilia de forma amigável uma complexa atividade do domínio humano, que é a geração de textos concisos e com capacidade de expressão adequada, sem desconsiderar o uso de uma interface em linguagem natural. Adicionalmente, esta alternativa viabiliza a utilização do *parser*, e a automação completa de todo o processo restante de armazenamento das descrições das facetas na biblioteca e de busca dos padrões mais adequados para atender a uma dada questão.

### 3.4 Atividades do Método Proposto

A concepção do método, para representação de facetas de padrões de projeto em uma biblioteca corporativa e busca dos padrões mais adequados para atender questões de usuário, é realizada através de um processo manual auxiliado pela utilização do *parser* palavras (analisador sintático) (BICK, 2000). No entanto, a concepção do método proposto, viabiliza a construção uma ferramenta para sua automação, cuja utilização facilitaria substancialmente a utilização do método proposto e atuaria com um catalisador para a sua evolução.

A concepção do método, para atender a solução proposta na seção anterior, se baseia nas seguintes atividades:

a) rearranjo da redação da faceta “Propósito”, dos 23 padrões de projeto, para gerar uma Árvore Sintática (AS) Base para a geração da Forma Gráfica de Representação (FGR);

b) definição da notação da FGR;

c) geração da FGR da faceta “Propósito” para os 23 padrões de projeto, a partir da AS Base;

d) análise das reduções na FGR passíveis de automação;

e) análise das sugestões de redação para o bibliotecário e usuário da biblioteca;

f) captura do mapeamento automático da AS Base para a FGR correspondente;

g) definição do algoritmo de *matching* entre as FGRs da faceta “Propósito” e das questões que focam no objetivo do problema a ser solucionado.

### 3.4.1 Rearranjo da Redação dos Propósitos para Gerar uma Árvore Sintática (AS) Base para a Geração da Forma Gráfica de Representação (FGR)

O Anexo A “AS Base para gerar FGRs dos Propósitos” captura a AS da redação original, a AS de redações intermediárias, a AS Base da redação da solução e as observações analisando as alterações necessárias.

Na AS da redação original, o texto original é baseado no texto proposto da literatura (GAMMA, 1995). As AS de redações intermediárias caracterizam as tentativas de alteração de redação para atingir a AS Base. A AS Base corresponde à redação da solução, cujo mapeamento para a FGR pode ser automatizado. São anotadas observações, com a análise sucinta das alterações necessárias para obter a AS Base a partir da redação da solução. Estas observações são utilizadas para definir as sugestões de redação dos textos dos propósitos e das questões.

### 3.4.2 Notação da Forma Gráfica de Representação (FGR)

A notação para os nós do grafo da Forma Gráfica de Representação (FGR) é a seguinte:

- a) Núcleo da Relação: Elipse;
- b) Entidade: Retângulo;
- c) Entidades referenciadas: Retângulo com *label* En (n inteiro) à esquerda;
- d) Co-referente: círculo com *label* equivalente ao *label* da entidade referenciada;
- e) Sintagma: Entidade ligada a outra Entidade.

Observe que a denominação “Núcleo da Relação”, que captura ao verbo da oração, permite a distinção com a denominação de “Relação”, que captura o Núcleo da Relação (verbo) e as Entidades associadas (objeto direto, objeto indireto, complemento).



A FIG. 8 apresenta a legenda correspondente à notação para os nós do grafo da FGR.

Legenda- Notação para Nós

Representação	Descrição
	Entidade
	Núcleo da Relação
	Entidade Referenciada
	Co-referente
	Sintagma

**FIGURA 8 - Legenda da notação dos nós do grafo FGR.**

A notação para os arcos do grafo da FGR é a seguinte:

- a) arcos podem ter um *label* associado;
- b) arcos podem ter múltipla origem ou múltiplos destinos e neste caso podem ser rotulados com E (AND) ou com OU (OR);
- c) arcos que indicam subordinação têm uma flecha cuja origem é o elemento principal e cujo destino é o elemento subordinado;
- d) arcos que indicam coordenação têm flechas no dois sentidos;
- e) arcos que associam uma Entidade (sujeito) com o Núcleo da Relação (Verbo) têm um pequeno círculo do lado do Núcleo da Relação.



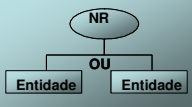
Os arcos que indicam subordinação são utilizados para indicar a subordinação entre duas orações ou ligar Entidade Principal com Entidade Subordinada (Sintagma).

A FIG. 9 apresenta a legenda correspondente à notação para os arcos do grafo da FGR.

No caso de Sintagma, o arco pode opcionalmente ter um *label* associado decorrente de um verbo no participio atuando como qualificador, como ocorre no

exemplo: “Interface desejada pelo Cliente”, no qual o arco entre “Interface” e “Cliente” é rotulado como “desejada”.

Legenda- Notação para Arcos

Representação	Descrição
↓	Subordinação
↕	Coordenação
	Entidade-Núcleo Relação
	Múltiplas Origens (AND)
	Múltiplos Destinos (OR)
	Associação

**FIGURA 9 - Legenda da notação dos arcos do grafo FGR.**

A notação para os *labels* adicionais é a seguinte:

- a) Cardinalidade: 1 .. N (N inteiro) associado à Entidade (Sujeito, Objeto, Complemento ou Sintagma);
- b) Modificador: tracejado com conteúdo modificador.

Legenda- Notação para Labels

Representação	Descrição
1 .. N	Cardinalidade
-----	Modificador

**FIGURA 10 - Legenda da notação dos *labels* adicionais do grafo FGR.**

A FIG. 10 apresenta a legenda correspondente à notação para os *labels* adicionais do grafo da FGR.

A FIG. 11 que apresenta a FGR do padrão de projeto “*Chain of Responsibility*”, é utilizada para ilustrar vários elementos da notação:

- a) Núcleo de Relação: “Evitar”, “Acoplar”, “Encadear”, “Passar”, “Tratar”
- b) Entidades: “Solicitação”, “Objeto”, “Cadeia”
- c) Entidades referenciadas: “E1 – Remetente”, “E2 – Solicitação”
- d) Entidade Sujeito: entidade “Objeto” sujeito do Núcleo da Relação “Tratar”
- e) Sintagmas: “Remetente da Solicitação”, “Destinatário do Remetente da Solicitação”
- f) Co-referente: “E1”, “E2”
- g) Modificador de Entidade: “receptor” da Entidade “Objeto”
- h) Associação de Subordinação: “.”, “até que”
- i) Associação de Coordenação: “E”
- j) Associação com *label*: “a”
- k) Cardinalidade: “N” associada a Entidade “Objeto receptor”

A FIG. 12, que apresenta a FGR do padrão de projeto “Memento”, é utilizada para ilustrar elementos adicionais da notação:

- a) Modificador de Núcleo da Relação: “posteriormente” do Núcleo da Relação “Restaurar”
- b) Arco com origem múltipla rotulado com “E”: arco com origem “Capturar e Externalizar”

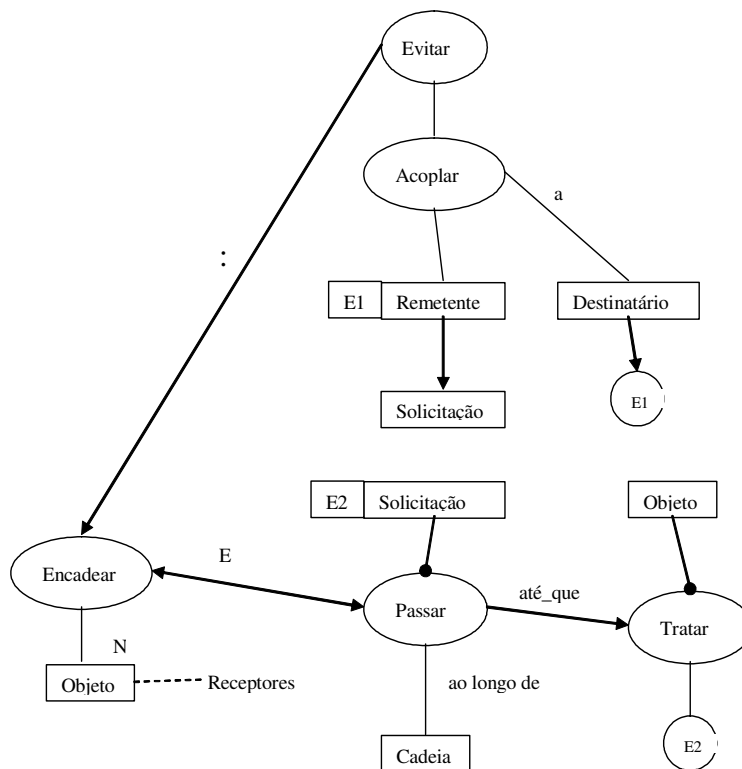


FIGURA 11 - FGR da faceta propósito do padrão de projeto "Chain of Responsibility".

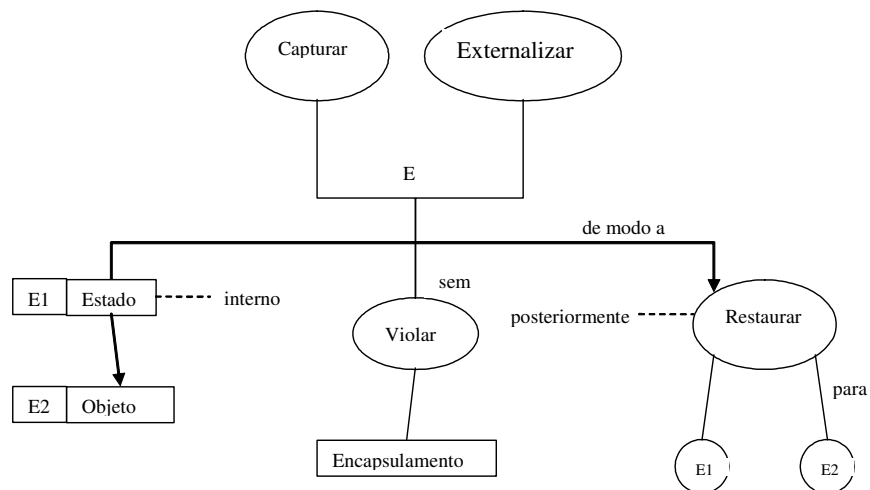


FIGURA 12 - FGR da faceta propósito do padrão de projeto "Memento".

### 3.4.3 FGR dos Propósitos a partir da AS Base

O Anexo A “AS Base para gerar FGRs dos Propósitos” define para o propósito de cada padrão de projeto, uma redação do propósito (denominada redação solução) que permite mapear a FGR de forma automática. Como o método proposto ainda não é suportado por ferramentas, as FGRs de cada propósito foram geradas manualmente, estão disponíveis no Anexo B “FGRs dos Propósitos dos Padrões de Projeto”. A geração das FGRs seguiu as regras para mapeamento automatizado definidas na seção 3.4.6. A FIG. 11 e a FIG. 12, apresentadas na seção anterior, ilustram FGRs de dois padrões de projeto.

### 3.4.4 Reduções Possíveis de Automação na FGR

As FGRs dos propósitos dos padrões de projeto foram mapeadas a partir das AS Base geradas para a redação solução de cada propósito. Esta seção investiga as possíveis reduções, para cada um dos padrões de projeto, que são passíveis de automação.

O Anexo C “Reduções Automatizáveis das FGRs dos Propósitos” descreve, para cada propósito dos 23 padrão de projeto, as reduções nos grafos das FGRs de cada propósito que são passíveis de automação. Os padrões de projeto, cujos propósitos não apresentam possibilidades de reduções, foram omitidos.

A partir das reduções possíveis, para cada um dos padrões de projeto, o referido anexo agrupa as reduções encontradas nos tipos possíveis de redução. Para cada tipo de redução possível, o anexo descreve: o tipo da redução, as condições que devem ser cumpridas para que a aplicação da redução seja possível, a alteração decorrente da aplicação da redução na representação original, e os propósitos dos padrões de projeto para os quais a redução é aplicável.

As reduções encontradas foram agrupadas nos seguintes tipos:

a) Redução 1: Remoção do Núcleo de Relação ligado a um Núcleo de Relação

b) Redução 2: Remoção do Núcleo de Relação ligado a um segundo Núcleo de Relação e à Entidade Objeto Direto

c) Redução 3: Remoção do Núcleo de Relação ligado a um segundo Núcleo de Relação e à Entidade Sujeito de um terceiro Núcleo de Relação

d) Redução 4: Remoção de Núcleo de Relação ligado a dois Núcleos de Relações

e) Redução 5: Remoção do Núcleo de Relação auxiliar de Núcleo de Relação composto

f) Redução 6: Conversão de *label* de associação com conotação de negação para Modificador de negação do respectivo Núcleo de Relação.

As cinco primeiras reduções capturam as cinco situações encontradas nas quais um Núcleo de Relação (caracterizado pelo verbo de uma oração) pode ser removido.

A aplicação de cada tipo de redução está detalhada no Anexo C. A seguir é ilustrada a aplicação de uma dada redução. A FIG. 13 apresenta a FGR original do padrão *Abstract Factory*. A FIG. 14 apresenta a FGR reduzida do mesmo padrão, após a aplicação da redução 2.

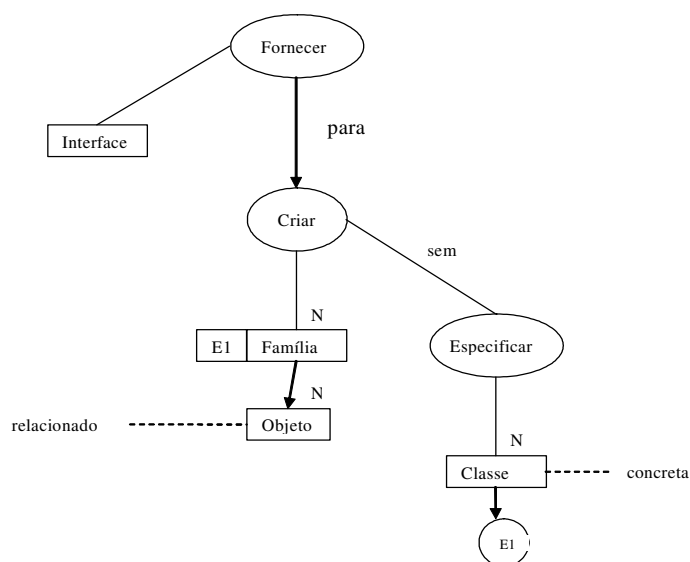
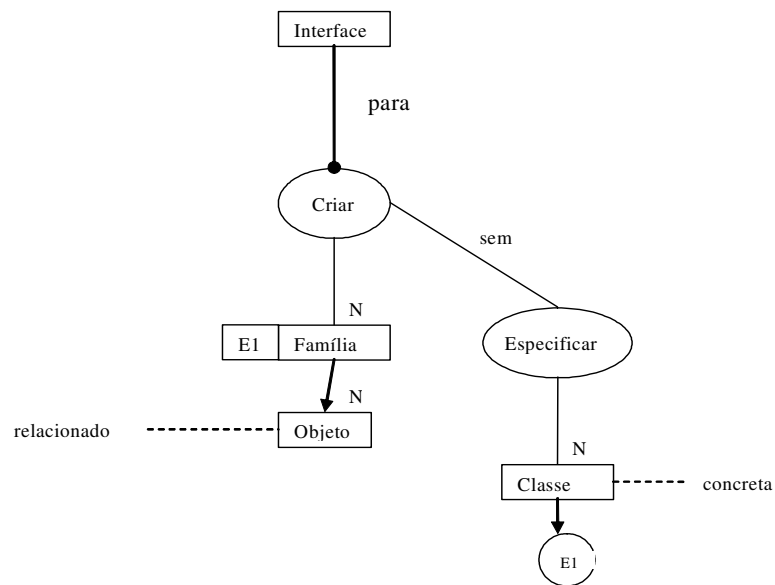


FIGURA 13 - FGR original da faceta Propósito do padrão de projeto “*Abstract Factory*”.

Observe que o Núcleo da Relação (verbo) “Fornecer” é removido, e a Entidade Interface passa a ser o Sujeito do Núcleo da Relação “Criar”. Esta redução elimina um verbo com pouca semântica para a compreensão do propósito do padrão e associa a Entidade Interface a um verbo com semântica relevante para o domínio de aplicação de interesse.



**FIGURA 14 - FGR reduzida da faceta propósito do padrão de projeto “Abstract Factory”.**

### 3.4.5 Sugestões de Redação para Bibliotecário e Usuário da Biblioteca

O Anexo D “Análise das Alterações de Redação para gerar Sugestões para os Usuários” analisa, a partir do Anexo A “AS Base para gerar FGRs dos Propósitos”, as necessidades de alterações na redação de cada um dos padrões de projeto de 1 a 23.

As sugestões para redação de facetas e questões descritas no referido anexo, podem ser agrupadas de acordo com os grupos descritos abaixo.

Grupo 1 de Sugestões: dar preferência ao infinitivo do verbo

a) utilizar verbo no infinitivo, em vez de verbo substantivado, em construção iniciada pela preposição “para”. Exemplo: utilizar “para criar” em vez de “para criação de”.

b) utilizar verbo principal no infinitivo, em vez verbo substantivado, em construções que um verbo auxiliar precede o verbo principal. Exemplo: utilizar ““Evita acoplar o”” em vez de “Evita o acoplamento do” .

c) utilizar verbo no infinitivo para iniciar uma oração subordinada, em vez do verbo substantivado correspondente. Exemplo: utilizar “enfileirar as” em vez de “enfileiramento das”.

d) utilizar verbo no infinitivo para iniciar uma oração subordinada, em vez do verbo no gerúndio.

Grupo 2 de Sugestões: separação por vírgulas, ponto e vírgula, dois pontos ou conjunção coordenada “e”

a) delimitar sintagma precedido da preposição “de” entre vírgulas, para evitar sua vinculação incorreta com um verbo que o antecede.

b) separar da oração principal a primeira oração iniciada por verbo no gerúndio através do uso de dois pontos.

c) separar uma próxima oração iniciada por verbo no gerúndio, da oração anterior também iniciada por verbo no gerúndio, por vírgula ou por conjunção coordenada “e”.

d) separar através do uso dois pontos, verbo no gerúndio de orações que o sucedem atuando como objeto direto (orações subordinativas objetivas diretas).

e) separar, através do uso de dois pontos, a oração principal da oração subordinada final (iniciada pela preposição “para”).

f) separar, por vírgula, a oração principal da oração subordinada iniciada por gerúndio.



g) remover vírgulas que delimitam parte de um sintagma iniciada pela preposição “de”.

Grupo 3 de Sugestões: mover objeto indireto ou adjunto adverbial para suceder o verbo

a) representar o objeto indireto iniciado com a preposição “de” para a posição entre o verbo e o objeto direto, evitando sua interpretação incorreta como parte de um sintagma.

b) representar o adjunto adverbial iniciado com a preposição “de” para a posição entre o verbo e o objeto direto, evitando sua interpretação incorreta como parte de um sintagma.

Grupo 4 de Sugestões: não utilizar pronome relativo antecedendo substantivo correlacionado

a) eliminar o uso do pronome relativo “qual”, substituindo a construção iniciada pelo referido pronome e sucedida por substantivo, verbo de ligação e verbo no particípio, pela construção iniciada por substantivo sucedido de verbo no infinitivo preposicionado. Exemplo: utilizar “a classe para instanciar” em vez de “qual classe a ser instanciada”.

Grupo 5 de Sugestões: utilizar advérbio em vez de adjunto adverbial

a) utilizar advérbio como uma única palavra com sufixo “mente” em vez de adjunto adverbial iniciado pela preposição “de”, para evitar que o adjunto adverbial seja erroneamente interpretado com parte de um sintagma. Exemplo: utilizar “uniformemente” em vez de “de maneira uniforme”. Exemplo: utilizar “eficientemente” em vez de “de maneira eficiente”

Grupo 6 de Sugestões: utilizar adjetivo no lugar de particípio

a) utilizar adjetivo no lugar de: sintagma verbal formado por verbo de ligação sucedido de verbo no particípio. Exemplo: trocar “classe a ser utilizada” por “classe existente”.

Observe que a alternativa de representar como “interface do cliente” não é adequada, porque elimina a semântica do verbo no particípio, que pode ser significativa; como por exemplo: “interface rejeitada pelo cliente”.

Grupo 7 de Sugestões: encadear sintagmas nominais utilizando a preposição “de”

a) utilizar sintagmas encadeados pela preposição “de” em vez de encadeados pela preposição “em”. Exemplo: utilizar “interfaces de um sistema” por “interfaces em um sistema”.

Grupo 8 de Sugestões: substituir palavras agrupadas por hífen

a) substituir expressões do tipo “todo-partes” por dependente de sintagma. Exemplo: utilizar “hierarquias de agregação” em vez de “hierarquias todo-partes”.

### **3.4.6 - Mapeamento Automático da AS para FGR**

A captura do mapeamento da AS para a FGR a partir da AS Base é simples. As principais regras de mapeamento são as seguintes:

- a) sujeito é mapeado para Entidade sujeito da relação;
- b) verbo principal é mapeado para o Núcleo da Relação;
- c) verbo auxiliar é descartado;
- d) objeto direto é mapeado para Entidade não preposicionada;
- e) objeto direto e adjunto adverbial são mapeados como Entidades preposicionadas;
- f) adjetivos são mapeados como Modificadores de Entidade;
- g) advérbios são mapeados como Modificadores de Núcleo da Relação;

h) substantivos dependentes são mapeados como Entidades que encadeiam um sintagma nominal;

i) conjunções coordenadas, subordinadas são mapeadas nas respectivas associações;

j) separações (dois pontos, ponto e vírgula, e vírgula) são mapeadas nas respectivas associações;

k) substantivos no plural são mapeados em Entidades com cardinalidade N.

### 3.4.7 - Algoritmo de *Matching* para FGRs

Em essência o algoritmo de *matching* entre FGRs deve buscar interseção entre a FGR de uma dada questão e as FGRs da faceta correlata. Observe que embora o experimento deste trabalho tenha sido realizado com a faceta “Propósito”, o algoritmo de *matching* é genérico, ou seja, compara a FGR de uma questão com as FGRs da faceta correlata ao tipo de questão enunciado (ver seção 1.3).

Para a FGR de uma dada questão, o algoritmo de *matching* deve calcular o grau de interseção desta FGR com as FGRs da faceta relacionada. Então, a resposta à questão formulada por um usuário da biblioteca, é definida pela lista dos padrões de projeto que atendem à questão, em ordem decrescente de grau de similaridade, que neste trabalho está sendo denominado de grau de interseção, com a questão do usuário. Portanto, o foco desta seção é a definição do cálculo do grau de interseção entre duas FGRs.

A análise das FGRs, do Anexo B “FGRs dos Propósitos dos Padrões de Projeto”, resulta na possibilidade de replicação de Entidade Sujeito em algumas FGRs e nos tipos de relações encontrados nas FGRs.

A replicação de Entidade Sujeito é possível nas FGRs dos padrões de projeto “*Abstract Factory*” e “*Template Method*”. Esta situação ocorre na seguinte configuração de uma parte da FGR:

a) o Núcleo de Relação NR1 está associado a: uma Entidade sujeito E1, uma Entidade objeto E2 e a um outro Núcleo de Relação NR2;

b) o Núcleo de Relação NR2 está associado somente a uma Entidade objeto E3, e portanto não tem sujeito.

Nesta configuração, a Entidade sujeito E1 pode ser replicada como sujeito também do Núcleo de Relação NR2.

Os tipos de relações encontrados nas 23 FGRs do referido anexo, são os seguintes: relações simples, duplas ou triplas. Para as considerações que se seguem, “En” é utilizado para denotar a “Entidade n”, para n inteiro positivo, e “NRn” é utilizado para denotar o “Núcleo de Relação”.

Relações simples são do tipo:

- a) E1 – NR;
- b) NR – E2;
- c) E1 – NR – E2.

Relações duplas são do tipo:

- a) NR – E2 – E3;
- b) E1 – NR – E2 – E3.

Relações triplas são do tipo:

- a) NR – E2 – E3 – E4;
- b) E1 – NR – E2 – E3 – E4.

Nestes tipos de relações: E1 é a Entidade sujeito; E2 é uma Entidade não preposicionada (objeto direto) ou preposicionada (objeto indireto ou adjunto); E3 e E4 são Entidades preposicionadas.

O Núcleo da Relação pode ser simples ou composto (dois núcleos de relação encadeados), sem modificador, com modificador negação, ou com outro tipo de modificador.

Entidades podem ser simples ou compostas (com relação de conjunto: E ou OU), isoladas ou sintagmas, sem modificador, ou com modificador.

Associações entre Núcleo de Relação podem ser de subordinação, de coordenação ou decorrentes de separadores (dois pontos, ponto e vírgula ou vírgula).

Passos do algoritmo de *matching*:

a) reduções das FGRs conforme os tipos de reduções previstas na seção 3.4.4;

b) replicação da Entidade sujeito quando ocorrer uma configuração favorável à replicação, conforme definido anteriormente nesta seção;

c) ampliação de termos (Núcleo de Relações, Entidades, Modificadores de Núcleo de Relações, Modificadores de Entidades, *labels* de associações) antes de efetuar a comparação entre termos afins, com exceção de modificadores de negação que são sempre reduzidos ao modificador “não”;

d) determinação da interseção entre relações das FGRs que estão sendo comparadas;

e) cálculo do grau de intersecção entre as FGRs, a partir da soma ponderada dos graus de intersecção de suas relações.

A caracterização de uma interseção entre relações das FGRs é determinada da seguinte forma:

a) os termos dos Núcleos das Relações devem ser coincidentes;

b) se os Núcleos das Relações forem compostos, pelo menos um deles deve coincidir;

c) os termos das Entidades devem coincidir;

d) se a Entidade for um Sintagma, pelo menos uma das entidades que compõe o Sintagma devem coincidir.

A coincidência entre dois termos é suportada por uma ontologia externa hipotética projetada para o domínio de padrões de projeto. Desta forma, o termo1 e o termo2 são considerados coincidentes, se após a aplicação da lematização prevista na seção 2.7.1, ocorre umas das seguintes situações:

a) o termo1 é idêntico ao termo2;

b) existe, na ontologia que suporta o processo de *matching*, um termo3 similar ou mais genérico que termo2 e idêntico ao termo1.

A ponderação de cada relação de interseção é determinada da seguinte forma:

a) uma relação simples tem valor 1;

b) uma relação dupla tem valor 3, sendo portanto, priorizada em relação a duas relações simples de valor 1;

c) uma relação tripla tem valor 5, sendo portanto, priorizada em relação a três relações simples de valor 1;

d) a não coincidência de um Modificador de negação do Núcleo de Relação, implica que a relação nega a outra relação, sendo portanto penalizada pela aplicação de um fator multiplicador de correção de valor -1;

e) a não ocorrência de coincidência do outro termo de um Núcleo de Relação composto, é penalizada pela aplicação de um fator multiplicador de correção de valor 0,7;

f) a não ocorrência de outros termos de entidades que compõe um Sintagma é penalizada pela aplicação de um fator multiplicador de correção de valor 0,8;

g) a não coincidência de um Modificador de Núcleo de Relação, com exceção do Modificador de negação, é penalizada pela aplicação de um fator multiplicador de correção de valor 0,9;

h) a não coincidência de um Modificador de Entidade, com exceção do Modificador de negação, é penalizada pela aplicação de um fator multiplicador de correção de valor 0,9.

Observe que como o cálculo do grau de interseção entre duas FGRs está considerando somente a interseção de relações, são excluídas deste cálculo as associações entre relações e as coincidências de termos isolados.

O cálculo do grau de interseção entre FGRs, definido nesta seção, é uma proposta preliminar que será refinada com base na avaliação do método proposto descrita no capítulo seguinte.

## 4 AVALIAÇÃO E ANÁLISE DO MÉTODO PROPOSTO

Este capítulo 4 trata da validação do método proposto. Para questões, extraídas de fonte externa (ROCHA, 2005), que endereçam os objetivos dos problemas a serem solucionados pela utilização de padrões de projeto, são definidas as formas de representação gráfica. As reduções automatizáveis são aplicadas nas formas gráficas de representação e então o algoritmo de *matching* é aplicado para cada questão de usuário, em relação aos três padrões de projeto mais promissores.

Finalmente os resultados obtidos são apresentados identificando os padrões de projeto recomendados para recomendados pelo algoritmo de *matching* como solução para resolver um problema proposto.

### 4.1 Rastreabilidade das Avaliações

Para ilustrar o funcionamento do algoritmo de *matching* são utilizados os padrões *Composite*, *Abstract Factory* e *Bridge*.

O cenário de uso do método pelos usuários da biblioteca é descrito a seguir. Para cada problema apontado pelo questionário é criada uma Forma Gráfica de Representação (FGR) com base no *parser* PALAVRAS. Caso o usuário observe que a FGR gerada automaticamente a partir da descrição do padrão não representa adequadamente a sua intenção, o usuário poderá recorrer às sugestões de redação, alterar a redação e solicitar a geração automática de uma nova FGR. O algoritmo de *matching* aplica reduções nas FGRs (conforme descrito na seção 3.4.4) antes de realizar o *matching* das FGRs. O resultado do processo de *matching* é uma lista com os Padrões de Projetos recomendados como solução para o problema.

As questões obtidas na literatura (ROCHA, 2005) que endereçam problemas cuja solução pode potencialmente utilizar um padrão de projeto, estão descritas a seguir:

- a) Sempre que for necessário, adaptar uma interface para um cliente.



- b) Necessidade de um *driver* que faça Implementações Específicas para tratar objeto em diferentes meios persistentes.
- c) Utilização de Herança está complicando minha implementação.
- d) É possível criar um objeto sem ter conhecimento algum de sua classe concreta?
- e) Como criar famílias inteiras de objetos que têm algo em comum sem especificar suas interfaces?
- f) Cliente precisa tratar de maneira uniforme objetos individuais e composições desses objetos.
- g) Garantir que apenas um objeto exista, independente do número de requisições que receber para criá-lo.
- h) Como garantir que objetos que dependem de outro objeto fiquem em dia com mudanças naquele objeto?
- i) Como fazer com que os observadores tomem conhecimento do objeto de interesse?
- j) Como fazer com que o objeto de interesse atualize os observadores quando seu estado mudar?
- k) Como permitir que um grupo de objetos se comunique entre si sem que haja acoplamento entre eles?
- l) Como remover o forte acoplamento presente em relacionamentos muitos para muitos?
- m) Como permitir que novos participantes sejam ligados ao grupo facilmente?
- n) Sistema quer usar um objeto real, mas ele não está disponível (remoto, inacessível).

o) O acesso a um objeto concreto será através da interface conhecida através de sua superclasse, mas o cliente também não quer (ou não pode) saber qual implementação concreta está usando.

p) É possível criar um objeto sem ter conhecimento algum de sua classe concreta?

q) Criar um objeto novo, mas aproveitar o estado previamente existente em outro objeto.

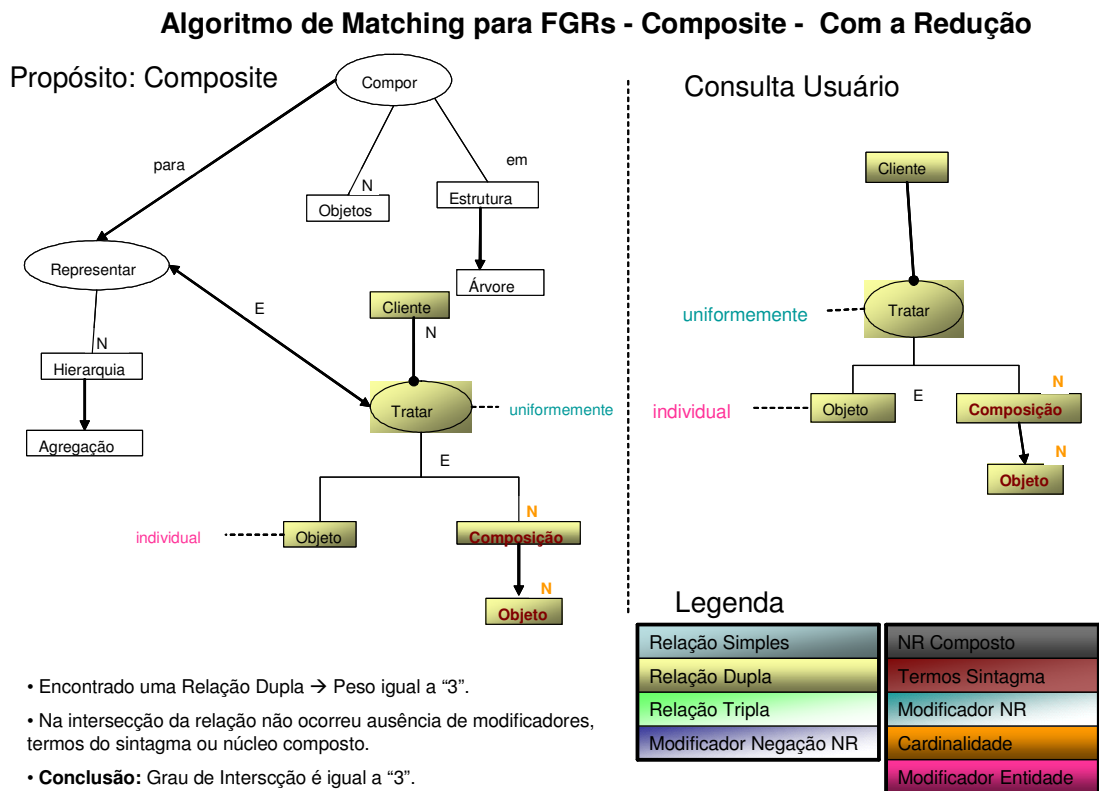
r) É preciso guardar informações sobre um objeto suficiente para desfazer uma operação, mas essas informações não devem ser públicas.

s) Usar objetos para representar estados e polimorfismo para tornar a execução de tarefas dependentes de estado transparentes.

As AS Bases dos padrões de projetos e das questões utilizadas encontram-se no Anexo A.

A FIG. 15 apresenta a forma de Representação Gráfica para o propósito do padrão “*Composite*” e para a questão: “Cliente precisa: tratar uniformemente objetos individuais e composições desses objetos.” O grau de intersecção poder ser visualizado através das cores similares entre o propósito do padrão que está na biblioteca e a consulta do usuário.

Para cada FGR, o algoritmo de *matching* proposto, identifica as relações existentes (simples, dupla ou tripla) com fator essencial de *matching*, e adicionalmente, os sintagmas, e os modificadores. As relações encontradas, no padrão de projeto e na questão de usuário, são comparadas para determinar a intersecção entre as FGRs. Para cada intersecção realizada é atribuído um peso de acordo com a relação (simples, dupla ou tripla). De acordo com a FIG. 15 é identificada apenas uma relação dupla com o peso igual a 3. Neste caso, nenhuma “não ocorrência” foi encontrada e conseqüentemente nenhuma penalidade foi aplicada. Para esta comparação o grau de intersecção é igual a 3.



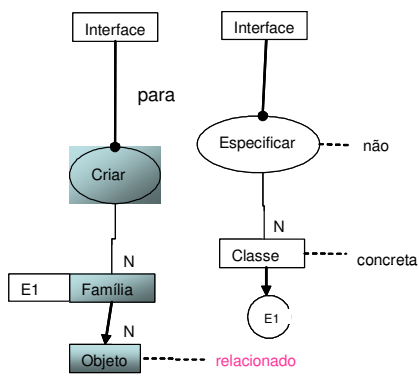
**FIGURA 15 - Algoritmo de *matching* das FGRs – Composite.**

O próximo exemplo é referente ao Padrão *Abstract Factory*. A questão a ser tratada é :” Como criar famílias inteiras de objetos que têm algo em comum sem especificar suas interfaces?”. Para a comparação desta questão e do propósito do *Abstract Factory* são encontradas duas relações simples. Estas relações são apresentadas na FIG. 16 e na FIG. 17.

Na FIG. 16 é encontrada uma relação simples. Para esta relação é atribuído o peso igual a “1”. É possível observar a ausência de dois modificadores de entidade. O primeiro é o termo “relacionado” e o segundo o termo “inteira”. Para cada ausência de modificador deve-se multiplicar o peso da relação por 0,9. Neste caso como são dois modificadores, é multiplicado duas vezes por 0,9. Na FIG. 16 ainda é encontrada a ausência de um modificador da relação (NR) e o peso é multiplicado mais uma vez por 0,9. O grau de intersecção para esta relação é igual a 0,729.

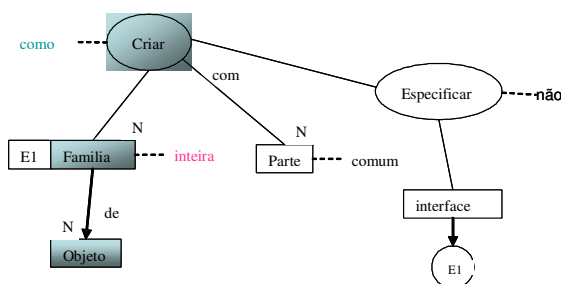
### Algoritmo de Matching para FGRs – Abstract Factory – 1a Relação

Propósito: Abstract Factory



Consulta Usuário

1a Parte



- Encontrado uma Relação Simples → Peso igual a “1”.
- Encontrado ausência de dois modificadores de entidade → Multiplicar duas vezes por 0,9
- Encontrado ausência de um modificador da relação (NR) → multiplicar por 0,9.
- **Conclusão:** Grau de Intersecção =  $1 * 0,9 * 0,9 * 0,9 = 0,729$

#### Legenda

Relação Simples	NR Composto
Relação Dupla	Termos Sintagma
Relação Tripla	Modificador NR
Modificador Negação NR	Cardinalidade
	Modificador Entidade

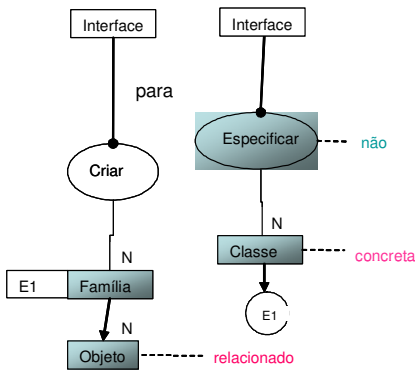
FIGURA 16 - Algoritmo de *matching* das FGRs – *Abstract Factory*.

Na FIG. 17 é apresentada a segunda relação encontrada na comparação entre o propósito do padrão *Abstract Factory* e a questão: “É possível criar um objeto sem ter conhecimento algum de sua classe concreta?”. Nesta comparação é encontrada uma relação simples também. Para esta relação é atribuído o peso igual a “1”. É possível observar a ausência de dois modificadores de entidade. O primeiro é o termo “relacionado” e o segundo o termo “concreta”. Para cada ausência de modificador deve-se multiplicar o peso da relação por 0,9. Neste caso como são dois modificadores, é multiplicado duas vezes por 0,9. Na FIG. 17 ainda é encontrada a ausência de um termo de sintagma e o peso é multiplicado mais uma vez por 0,8. O grau de intersecção para esta relação é igual a 0,648.

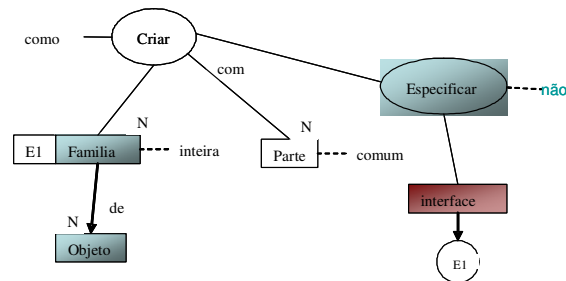
## Algoritmo de Matching para FGRs – Abstract Factory – 2a Relação

2a Parte

Propósito: Abstract Factory



Consulta Usuário



- Encontrado uma Relação Simples → Peso igual a “1”.
- Encontrado ausência de dois modificador da entidade → multiplicar duas vezes por 0,9.
- Encontrado ausência de um termo de sintagma → multiplicar por 0,8
- **Conclusão:** Grau de Intersecção =  $1 * 0,9 * 0,9 * 0,8 = 0,648$

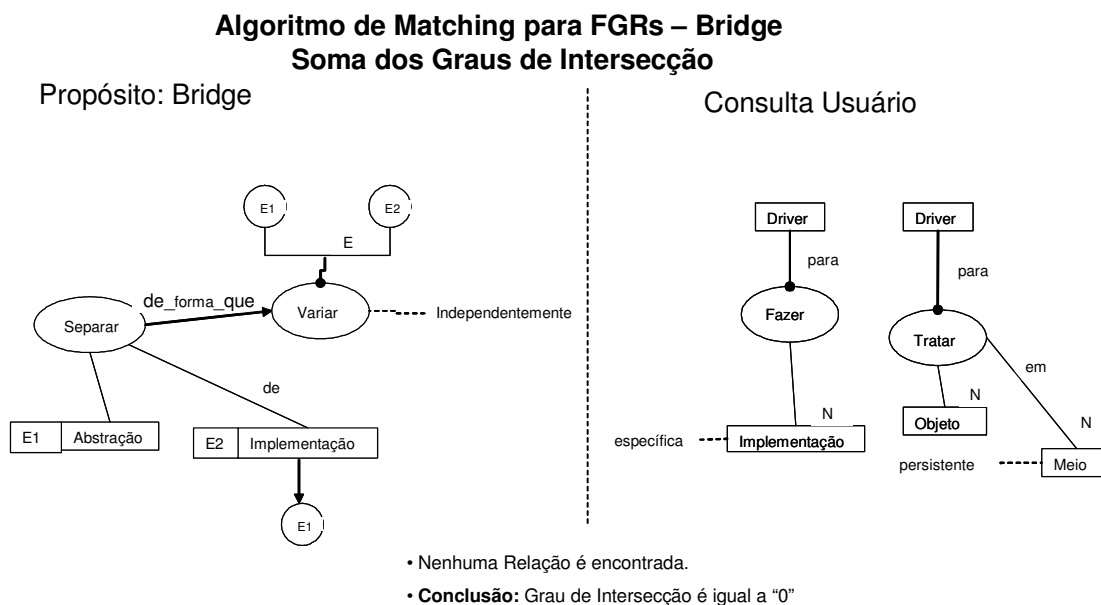
### Legenda

Relação Simples	NR Composto
Relação Dupla	Termos Sintagma
Relação Tripla	Modificador NR
Modificador Negação NR	Cardinalidade
	Modificador Entidade

FIGURA 17 - Algoritmo de *matching* das FGRs – *Abstract Factory*.

O terceiro e último exemplo é referente ao Padrão *Bridge*. A questão a ser tratada é: “Necessidade de um *driver* que faça Implementações Específicas para tratar objeto em diferentes meios persistentes;”. Para a comparação desta questão e do propósito do *Bridge* nenhuma relação é encontrada. Veja a FIG. 18.

Cada uma das questões foi comparada à faceta do propósito dos 23 padrões de projeto. O resultado foram 42 Formas Gráficas de Representação, referentes aos 23 propósitos dos padrões de projetos e às 19 questões. O algoritmo de *matching* é realizado comparando as 19 questões com os 23 propósitos dos padrões de projetos resultando num total de 437 comparações; portanto, a fórmula do Grau de Intersecção também foi aplicada 437 vezes.



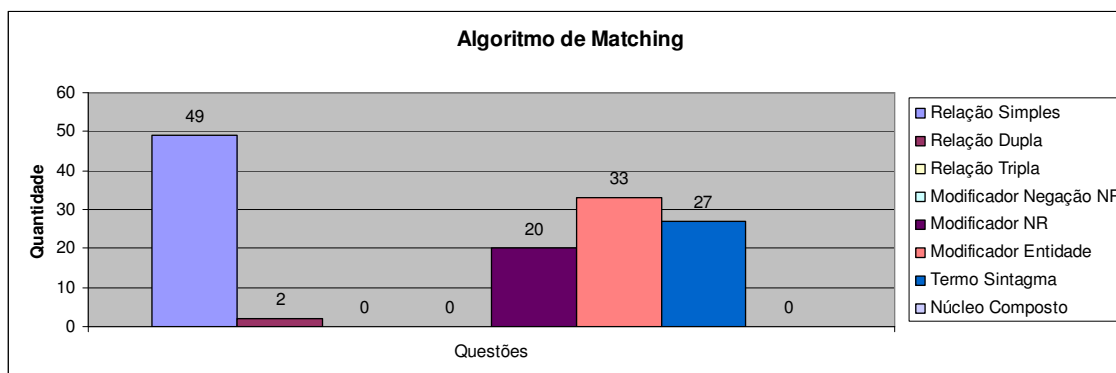
**FIGURA 18 - Algoritmo de *matching* das FGRs – *Bridge*.**

## 4.2 Resultados Obtidos

Esta seção compara os resultados obtidos através do método proposto com o resultado que seria obtido pelo ser humano. Para as comparações efetuadas entre os 23 padrões de projetos e as 19 questões do usuário, foi elaborado um gráfico com o objetivo de identificar quais as relações e os modificadores mais frequentes. Observa-se na FIG. 19, que a relação simples é tipo de relação que ocorre mais freqüentemente, como era de se esperar. Não foi encontrada nenhuma relação tripla durante as comparações. Entre os modificadores, o que apresentou maior freqüência foi o modificador de entidade. O termo de sintagma também apresenta uma freqüência significativa. Para os núcleos das relações o gráfico apresenta uma freqüência significativa para o modificador da relação (NR).

Foi analisada uma amostra de 19 questões em um experimento de busca nos 23 padrões analisados. Nesta análise foram encontrados três grupos de resultados: *matching* com padrão correto, *matching* com padrão incorreto e não ocorrência de *matching*. Para cada um dos três grupos foram analisadas as

características da redação utilizada. Como resultado desta análise, foram definidas algumas heurísticas específicas para a redação de questões. Estas heurísticas foram apresentadas na seção que descreve as sugestões de redação.



**FIGURA 19 - Relações e modificadores mais frequentes.**

O processo de busca de um componente que endereça um problema descrito por uma dada questão de usuário é interativo. A inspeção da FGR gerada automaticamente pelo sistema para uma dada questão de usuário, provê uma interface amigável para que o usuário altere a redação da questão e visualize uma nova FGR.

Quando a busca de um componente, para a solução de um dado problema, resulta na recuperação do padrão de projeto incorreto, o usuário pode verificar para o padrão recuperado com menor prioridade, se a FGR deste padrão atende a sua questão. Caso contrário, o usuário pode combinar a alteração da redação da sua questão com a inspeção das FGRs recuperadas. Para os casos onde nenhum padrão de projeto é recuperado, o usuário também pode combinar a alteração de redação com a inspeção das FGRs recuperadas.

Resultados de *matching* com indicação correta do padrão mais adequado para a questão ocorrem normalmente quando são utilizados verbos ou entidades (genéricos ou específicos) que pertencem ao domínio dos Padrões de Projetos. Neste tipo de *matching* uma parte da questão pode ser atendida (Exemplo: *Composite*) ou toda a questão é atendida (Exemplo: *Adapter*) e as entidades com semântica significativa para o domínio de padrões de projeto contribuem para o *matching*. Para a classificação geral dos padrões pode ser necessário analisar a

prioridade do resultado, pois o algoritmo de *matching* pode indicar outro padrão como o mais adequado para um dado problema ou mais de um padrão pode ter o mesmo grau de intersecção. O QUADRO 7 ilustra as questões onde o padrão de projeto recuperado, com maior grau de similaridade com a questão do usuário, corresponde ao padrão mais indicado.

**QUADRO 7 - Questões com indicação correta de *matching*.**

Intersecção FGR	Padrão Recomendado	Relações		
		Simples	Dupla	Tripla
Questão 1	Composite	2	1	0
Questão 4	Abstract Factory	3	0	0
Questão 5	Abstract Factory	7	0	0
Questão 6	Adapter	0	1	0
Questão 10	Observer	2	0	0
Questão 11	Mediator	2	0	0
Questão 13	Mediator	1	0	0
Questão 16	Factory Method	7	0	0
Questão 17	Prototype	7	0	0

Resultados de *matching* com indicação incorreta do padrão mais adequado para a questão ocorrem normalmente quando são utilizados verbos ou entidades (genéricos ou específicos) que pertencem ao domínio dos Padrões de Projetos. O QUADRO 8 ilustra as questões onde o padrão de projeto recuperado, com maior grau de similaridade com a questão do usuário, não corresponde ao padrão mais indicado.

**QUADRO 8 - Questões com indicação incorreta de *matching*.**

Intersecção FGR	Padrão Recomendado	Padrão Recuperado	Relações		
			Simples	Dupla	Tripla
Questão 2	Bridge	Composite	4	0	0
Questão 7	Singleton	Composite	8	0	0
Questão 14	Proxy	Prototype	1	0	0
Questão 15	Factory Method	Singleton	2	0	0
Questão 19	State	Prototype	3	0	0

Resultados com nenhuma ocorrência de *matching* ocorrem principalmente quando são utilizados verbos genéricos e fora do domínio dos padrões de projetos (Exemplo: Existir, Ter). Entidades com termos muito específicos também dificultam o *matching* (Exemplo: "Dependência"). O QUADRO 9 ilustra as questões para as quais



o padrão de projeto recuperado, com maior grau de similaridade com a questão do usuário, não corresponde ao padrão mais indicado.

**QUADRO 9 - Questões sem indicação de *matching*.**

<b>Intersecção FGR</b>	<b>Padrão Recomendado</b>
Questão 3	Bridge
Questão 8	Observer
Questão 9	Observer
Questão 12	Mediator
Questão 18	Memento

## 5 CONCLUSÕES E TRABALHOS FUTUROS

Neste capítulo os resultados obtidos são utilizados para analisar as forças e fraquezas do método proposto. Adicionalmente, são indicados trabalhos futuros para complementar o trabalho desta dissertação.

O objetivo do método proposto nesta dissertação é promover o reuso através da utilização de uma interface em linguagem natural para representação e busca de Padrões de Projeto em uma biblioteca corporativa. O reuso destes componentes proporciona às organizações maior flexibilidade e agilidade permitindo oferecer seus serviços com maior competitividade e com um grau mais elevado de personalização.

O analisador léxico e sintático desenvolvido por BICK (2000) possibilitou o suporte do modelo proposto, permitindo abranger a aplicação de regras gramaticais e fornecendo uma análise completa, tanto morfológica como sintática, de um texto qualquer. Sua representação em forma de árvore facilitou a análise e extração dos termos relevantes.

Para o método de busca foram definidas regras para avaliar o *matching* entre as FGRs de definição dos padrões (armazenadas na biblioteca corporativa) e a FGR de uma dada questão de usuário.

Foi observado que o grau de sucesso no processo de busca é muito dependente de como o bibliotecário e o usuário descrevem respectivamente o padrão e o problema a ser solucionado pelo padrão. O maior problema encontrado pela aplicação do método proposto é que a descrição do padrão é baseada em uma abstração que representa a concepção do padrão, enquanto a questão é muitas vezes formulada em relação a um problema em particular. Esta discrepância de foco, da formulação dos textos de descrição do padrão e da questão associada ao problema a ser resolvido pelo padrão, pode conduzir a um fator de *matching* nulo ou irrelevante entre os textos.

Entretanto, mesmo que o usuário consiga descrever corretamente sua necessidade, pode ser que não seja recuperado exatamente o que ele espera. Isso porque cada pessoa descreve um mesmo objeto de diversas formas. Esses são considerados os problemas do vocabulário (ou problemas da diferença de vocabulário). Estas ambiguidades causam problemas de sinonímia (onde vários termos podem denotar um mesmo objeto) e polissemia (onde um termo possui vários significados). Como o vocabulário, contido na consulta do usuário e nos padrões da biblioteca, pode variar, deve existir ainda uma base de conhecimento de apoio que contenha informações semânticas e relacionamentos (sinônimos, termos mais genéricos e termos mais específicos) para que o sistema possa utilizar associações e sugerir os padrões mais adequados.

O processo de *matching* é baseado na interseção ponderada das FGRs de descrição com a FGR de questão e sua eficácia depende da similaridade da redação do texto original a partir do qual cada FGR é mapeada. As melhorias na eficácia do *matching* dependem de reescrita na redação original com base nas FGRs geradas automaticamente

As sugestões para redação das questões devem procurar compensar situações nas quais questões descrevem algum problema em particular, minimizando a possibilidade de *matching* com descrições baseadas em abstrações genéricas.

As sugestões de redação, associadas ao suporte da geração automática de FGR para cada texto submetido, provêm uma interface bastante amigável para interações de melhoria da redação utilizada para as descrições de componentes e para as questões.

No processo de geração das FGRs toda a estrutura gramatical é representada; não só as palavras-chave mas também o relacionamento entre elas. Esta análise sintática é realizada com o auxílio de um analisador sintático desenvolvido por BICK (2000). A representação da estrutura gramatical de cada um dos períodos descreve os padrões de projeto e ou as questões de busca. A geração automática da FGR é realizada para cada período do texto, permitindo ao

bibliotecário e ao usuário uma interação amigável para aperfeiçoar a redação da descrição ou questão.

O *matching* entre os períodos das descrições da biblioteca e das questões do usuário é realizado de forma automática baseado na interseção de relações entre os dois períodos com ponderações baseadas no papel que os termos do período representam nas relações.

Como trabalho futuro pode-se destacar a elaboração de um ambiente de *software* para automatizar o método proposto. A realização desta ferramenta concorrerá para o refinamento do método proposto e facilitará a geração das descrições das demais facetas de descrição dos padrões de projeto. A realimentação decorrente da representação das demais facetas dos padrões de projeto concorrerá para o refinamento do proposto.

Outro ponto de grande interesse para um trabalho futuro é a análise de oportunidades de automação para as sugestões de redação para auxiliar de maneira eficiente a análise sintática feita pelo *parser*, diminuindo a necessidade de interações adicionais para refinar a redação das descrições e questões. Finalmente, a análise das associações entre relações no algoritmo de *matching* para identificar a relevância das associações no processo de intersecção entre as orações, permitirá explorar a semântica das orações coordenadas e subordinadas, contribuindo assim para o refinamento do método.

## 6 REFERÊNCIAS

AONE C.; RAMOS-SANTACRUZ, M. *REES: A Large-Scale Relation and Event Extraction System*. SRA International, Inc. In: Proceedings of the 6th Applied Natural Language Processing Conference, 2000

BARANOSKI F.L. *Verificação da autoria em documentos manuscritos usando SVM*. Pontifícia Universidade Católica do Paraná. 2005. 106f. Dissertação (Mestrado em Informática Aplicada) - Programa de Pós-Graduação em Informática Aplicada, Metodologias e Técnicas de Computação, Pontifícia Universidade Católica do Paraná, Paraná, 2005.

BICK, E. *The parsing system palavras: automatic grammatical analysis of portuguese in a constraint grammar framework*.[s.l.]: Arhus University, 2000.

BUCHHOLZ, S. Using grammatical relations, answer frequencies and the World Wide Web for question answering. In: TEXT RETRIEVAL CONFERENCE, 10., 2001.

BUNESCU R.C.; MOONEY R.J. *A Shortest Path Dependency Kernel for Relation Extraction*. Department of Computer Sciences ,University of Texas at Austin,2005

BUSCHMANN, F. et. al. *Pattern oriented software architectur: a system of patterns*. ChiChester: John Wiley & Sons, 1996.

CULOTTA A.; SORENSEN J. *Dependency Tree Kernels for Relation Extraction*. [s.l.]: University of Massachusetts,2003

DERSHOWITZ, N. *Shallow Parsing. Seminar in natural language processing and computational linguistics*, 2006.

FERRET, O. et al. Terminological variants for document selection and question/answer matching. In: Association for Computational Linguistics Workshop on Open-Domain Question Answering, p. 46-53, 2001.

GAMMA, E. et al. *Design patterns: elements of reusable object-oriented software*. [s.l.]: Addison-Wesley, 1995.

HARABAGIU, S. et al. FALCON: Boosting knowledge for answer engines. In: TEXT RETRIEVAL CONFERENCE, 9., 2001.

HOVY, E. et al. Question answering in webclopedia. In: TEXT RETRIEVAL CONFERENCE, 9., 2001.

IRIA J. *T-Rex: A Flexible Relation Extraction Framework*. Department of Computer Science, The University of Sheffield, UK, 2004.

ITTYCHERIAH, A; FRANZ, W-J; RATNAPARKHI, A. IBM's statistical question answering system. In: TEXT RETRIEVAL CONFERENCE, 9., 2001.

KANDO, N; SHIKAWA, H. Evaluation of Information Access Technologies. In: NTCIR WORKSHOP, 4., Tokyo, Japan, 2004.

KATZ, B. From sentence processing to information access on the world wide web. In: AAAI SPRING SYMPOSIUM, Stanford University, Stanford, CA, 1997. (*Natural Language Processing for the World Wide Web*)

KATZ, B. et al. Omnibase: Uniform access to heterogeneous data for question answering. In: INTERNATIONAL WORKSHOP ON APPLICATIONS OF NATURAL LANGUAGE TO INFORMATION SYSTEMS, 7., 2002.

LEONEL, J; MAUSBACH, A. *Sistemas de suporte à operações*. Disponível em: <[www.promon.com/portugues/noticias/download/OSS\\_4Web.pdf](http://www.promon.com/portugues/noticias/download/OSS_4Web.pdf)>. Acesso em: 28 out. 2006.

LIN, J. et. al. Extracting answers from the Web using knowledge annotation and knowledge mining techniques. In: TEXT RETRIEVAL CONFERENCE, 8., 2002.

MILI, H. et. al. Another nail to the coffin of faceted controlled-vocabulary component classification and retrieval. *Software Engineering Notes*. v.22, n.3, p. 89-98, 1997.

MOLDOVAN, D. et al. LASSO: A tool for surfing the answer net. In: TEXT RETRIEVAL CONFERENCE, 8., 2000.

MOLLA, D; BERRI, J; HESS, M. A real world implementation of answer extraction. In: INTERNATIONAL CONFERENCE ON DATABASE AND EXPERT SYSTEMS APPLICATIONS, 9., p. 143-148, 1998.

OSÓRIO, L.M. Operadoras e redes. *O Globo*, 2005. Disponível em: <<http://www.ibm.com/br/services/articles/2005/11/a36m051t09s012005.shtml>> Acesso em: 10 nov. 2006.

PRAGER, J. One search engine or two for question answering. In: TEXT RETRIEVAL CONFERENCE, 9., 2001.

PRIETO-DIAZ, R; FREEMAN, P. Classifying software for reusability. *IEEE Software*, v.4, n.1, p. 6-16, 1987.

PRIETO-DIAZ, R. Implementing faceted classification for software reuse. *Communications of ACM*, v.34, n.5, p. 88-97, 1991.

RAVICHANDRAN, D; HOVY E. Learning Surface Text Patterns for a Question Answering System. In: MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS PHILADELPHIA,40.,2002, Philadelphia. *Anais...* Philadelphia, p. 41-47, Jul.y 2002.

RILOFF, E. M. *Information extraction as a basis for portable text classification systems*. 1994. 166f. Graduate School of the University of Massachusetts in partial fulfillment of the requirements for the degree of Doctor of Philosophy, Department of Computer Science, Sep.1994.

ROCHA, H. *Padrões de design patterns com aplicações em Java.(Versão 2.1)*  
Disponível em: <[www.argonavis.com.br](http://www.argonavis.com.br)>. Acesso em: 28 out. 2006.

SCOTT, S; GAIZAUSKAS, R. University of Sheffield TREC-9 Q&A System. In: TEXT RETRIEVAL CONFERENCE, 9., 2001.

SORUMGARD, L.S; SINDRE, G; STOKKE, F. Experiences from application of a faceted classification scheme. In: INTERNATIONAL WORKSHOP ON SOFTWARE REUSABILITY, 2., 1993, Los Alamitos, CA. *Select papers...* Los Alamitos, CA, IEEE Computer Society Press, 1993, p. 116-124.

SOUBBOTIN, M. M; SOUBBOTIN S. M. Patterns of potential answer expressions as clues to the right answer. In: TEXT RETRIEVAL CONFERENCE, 10., Gaithersburg, p. 175-182, 2001.

SRIHARI, R; LI, W. Information extraction supported question answering. In: TEXT RETRIEVAL CONFERENCE, 8., 2000.

XAVIER, J. R. *Criação e instanciação de arquiteturas de software específicas de domínio no contexto de uma infra-estrutura de reutilização*. Rio de Janeiro: Universidade Federal do Rio de Janeiro, 2001.

SALTON, G; MACGILL, M. J. *Introduction to modern information retrieval*. New York: McGRAW-Hill, 1983. p.448.

WORDNET: a lexical database for the English language. Disponível em: <<http://wordnet.princeton.edu/>>. Acesso em: 27 dez. 2006.

ZAJAC, R.; Towards ontological question answering. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS WORKSHOP ON OPEN-DOMAIN QUESTION ANSWERING, p. 31-37, 2001.

ZELENKO D.; AONE C.; RICHARDELLA A. *Kernel Methods for Relation Extraction*. Journal of Machine Learning Research 3, 2003.

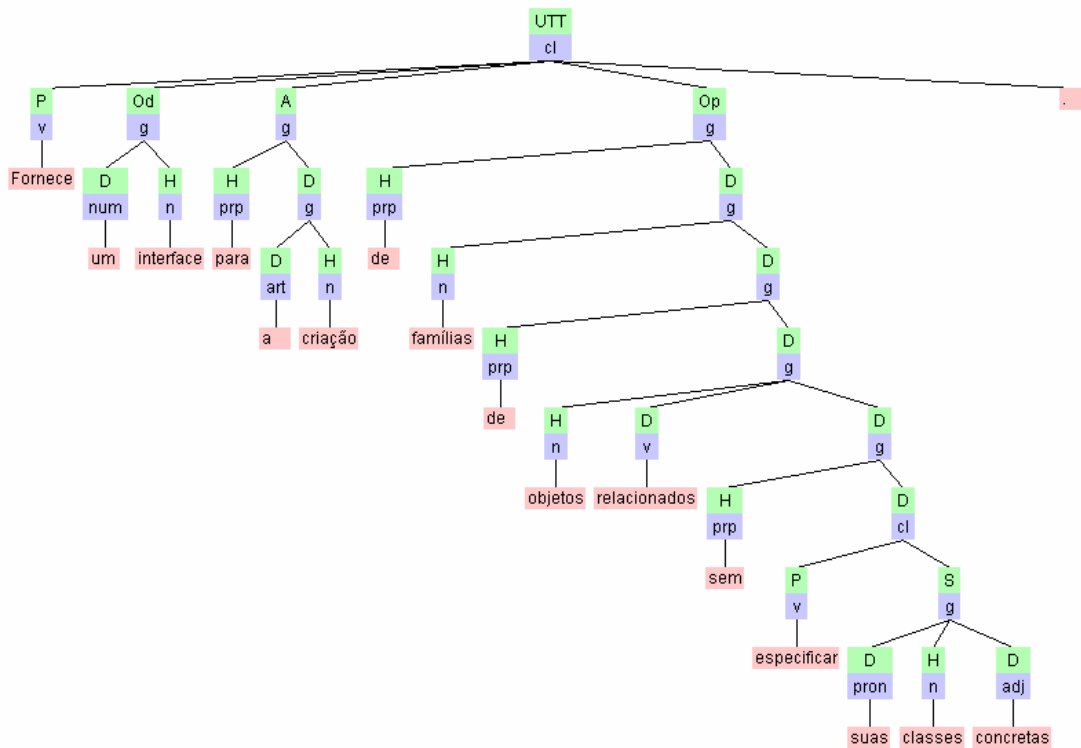
ZHANG, D; LEE, W. S. Web based pattern mining and matching approach to question answering. In: TEXT RETRIEVAL CONFERENCE, 8., 2002.

## 7 ANEXOS

### 7.1 Anexo A – AS Base para gerar FGRs dos Propósitos.

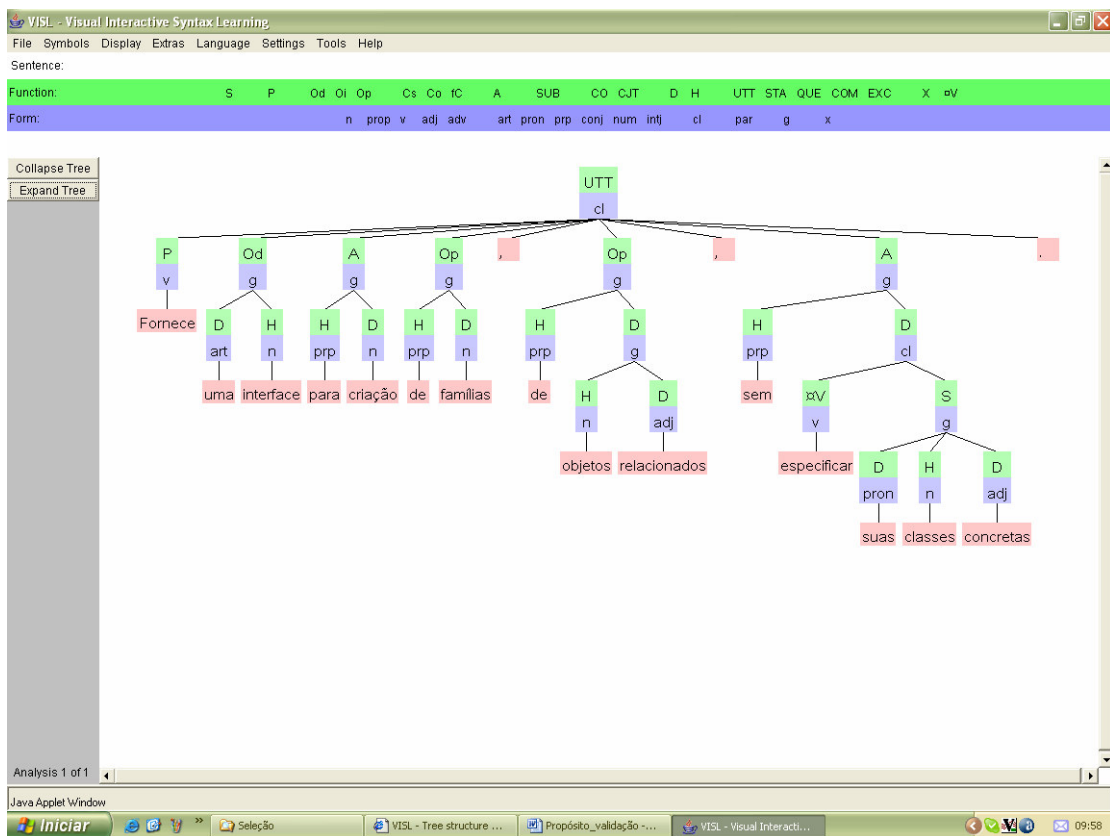
#### 01 - Abstract Factory

**Original** → Fornece uma interface para a criação de famílias de objetos relacionados sem especificar suas classes concretas.

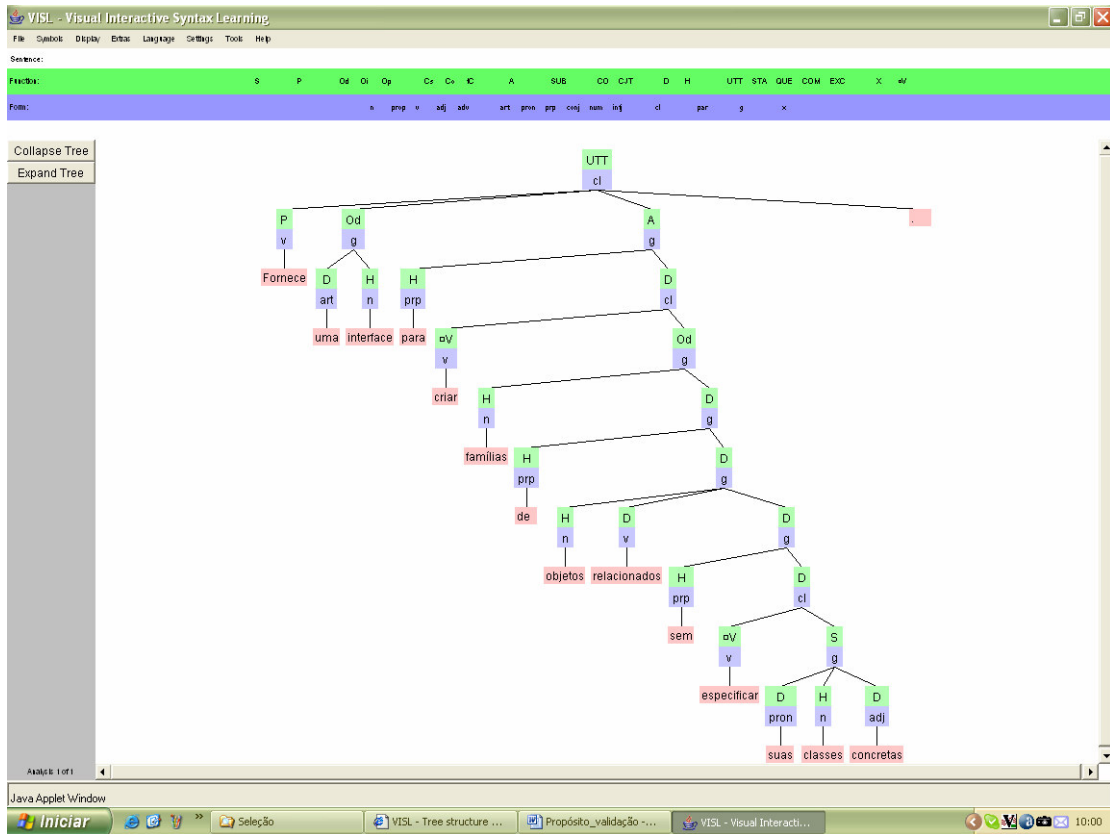




**Tentativa** → Fornece uma interface para criação de famílias, de objetos relacionados, sem especificar suas classes concretas.



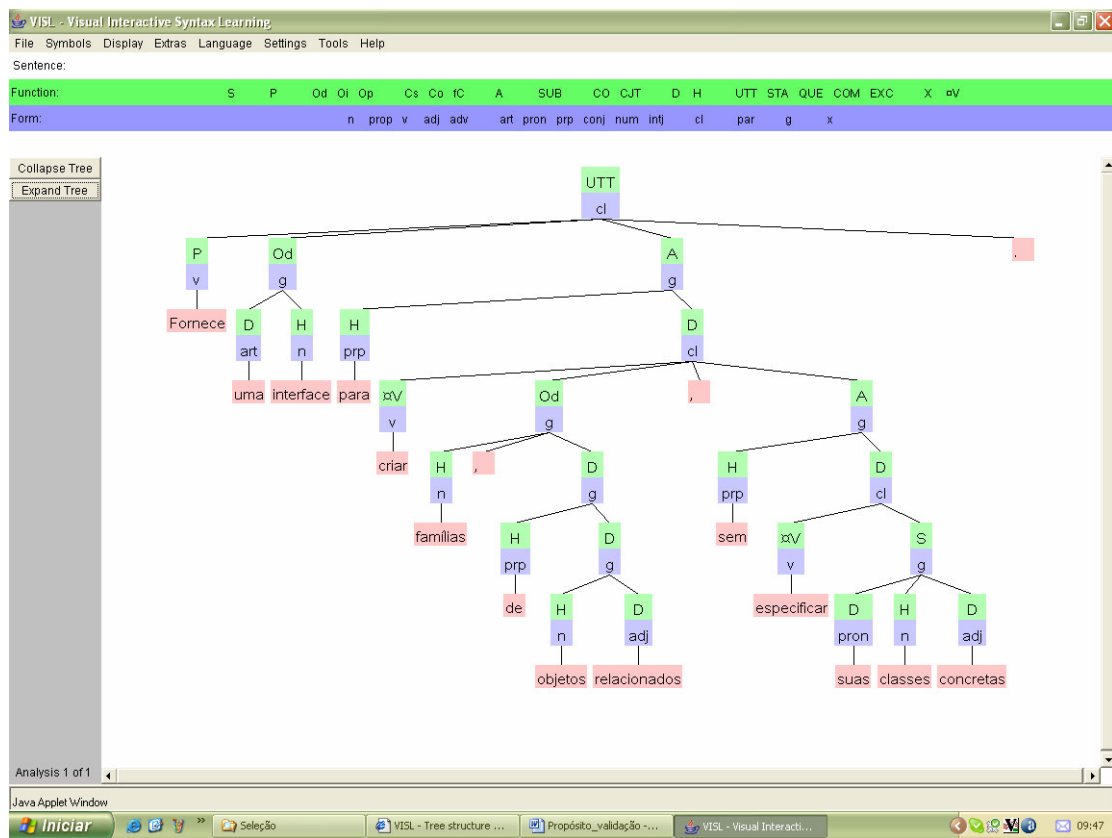
**Tentativa** → Fornece uma interface para criar famílias de objetos relacionados sem especificar suas classes concretas.



**Solução** → Fornece uma interface para criar famílias de objetos relacionados, sem especificar suas classes concretas.

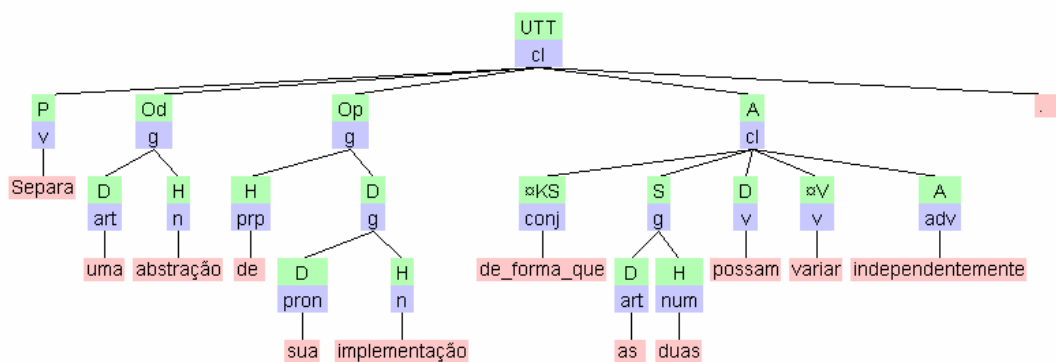
Obs

- com “para criação” vincula inadequadamente “sem especificar ...” a objetos relacionados → a caracterização de uma oração infinitiva resolve o problema



### 03 – Bridge

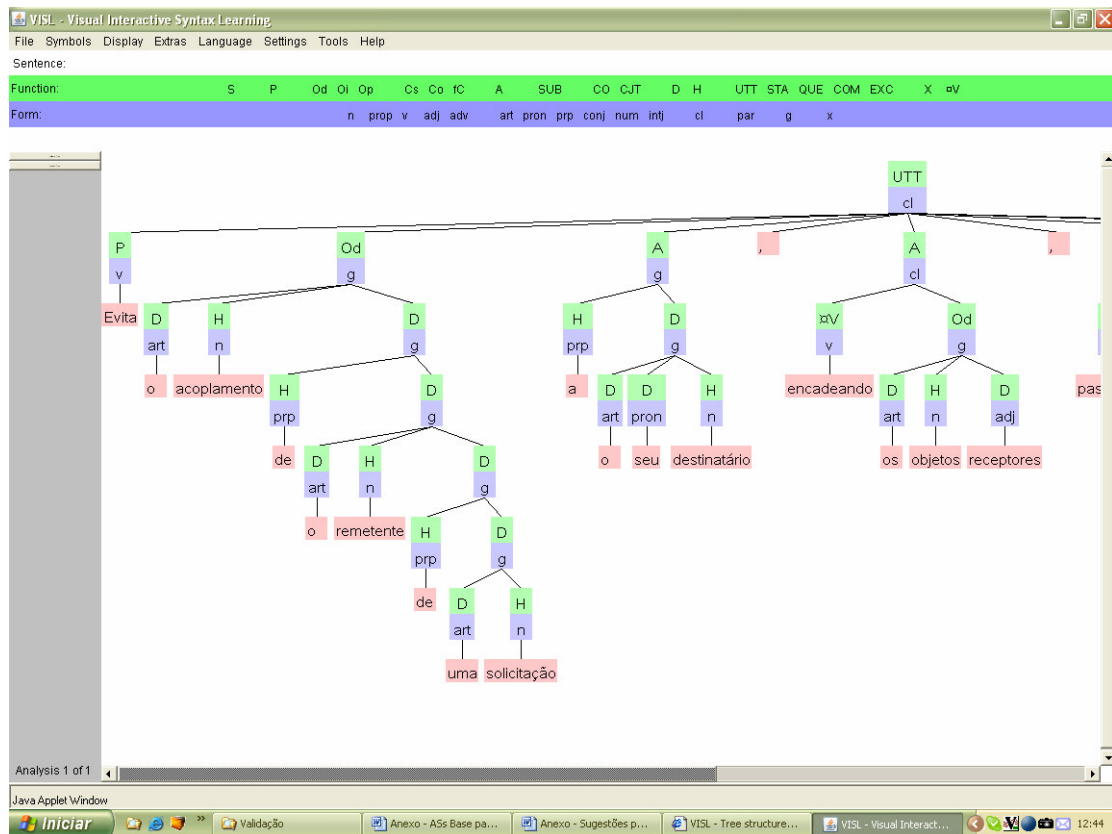
**Original e Solução** → Separa uma abstração de sua implementação de forma que as duas possam variar independentemente.



## 05 – Chain of Responsibility

**Original** → Evita o acoplamento do remetente de uma solicitação ao seu destinatário, encadeando os objetos receptores, passando a solicitação ao longo de uma cadeia até que um objeto a trate.

Início (a diminuição para um única figura a torna ilegível)



Continuação (a diminuição para um única figura a torna ilegível)

VISL - Visual Interactive Syntax Learning

File Symbols Display Extras Language Settings Tools Help

Sentence:

Function: S P Od Oi Op Cs Co tC A SUB CO CJT D H UTT STA QUE COM EXC X ev

Form: n prop v adj adv art pron prp conj num infj cl par g x

Analysis 1 of 1

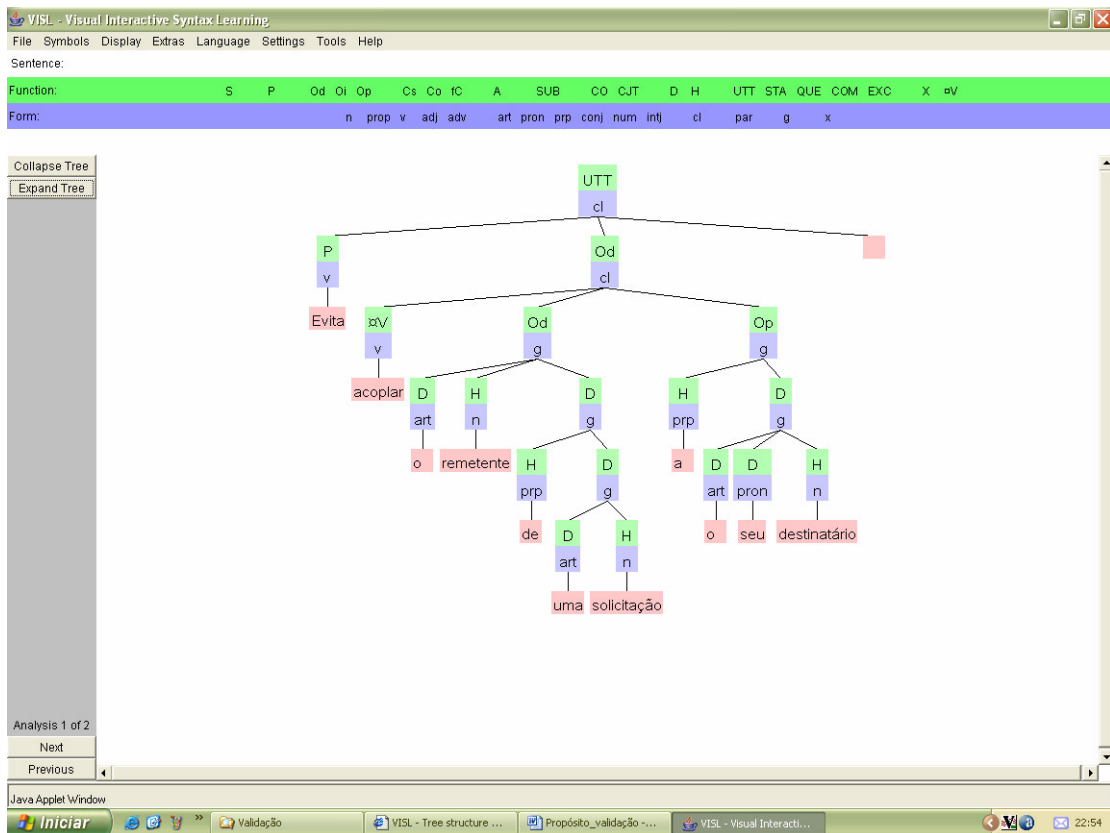
Java Applet Window

Windows taskbar: Iniciar, Validação, Anexo - ASs Base pá..., Anexo - Sugestões p..., VISL - Tree structure..., VISL - Visual Interact..., 12:45

Detailed description of the syntax tree: The root node is UTT (cl). It branches into three main parts: 1) A clause starting with the verb 'encadeando' (v) and a noun phrase 'os objetos receptores' (Od: art 'os', n 'objetos', adj 'receptores'). 2) A clause starting with the verb 'passando' (v) and a prepositional phrase 'a solicitação ao longo de uma cadeia' (S: art 'a', n 'solicitação', H: prp 'ao longo de', D: art 'uma', H: n 'cadeia'). 3) A clause starting with the preposition 'até' (prp) and a relative clause 'que um objeto trate' (SUB: conj 'que', S: art 'um', n 'objeto', Od: pron 'a', P: v 'trate').

**Solução** → Evita acoplar o remetente de uma solicitação ao seu destinatário: encadeando os objetos receptores e passando a solicitação ao longo de uma cadeia, até que um objeto a trate.

Parte Inicial (a diminuição para um única figura a torna ilegível)



Parte Final (a diminuição para um única figura a torna ilegível)

VISL - Visual Interactive Syntax Learning

File Symbols Display Extras Language Settings Tools Help

Sentence:

Function: S P Od Oi Op Cs Co IC A SUB CO CJT D H UTT STA QUE COM EXC X +V

Form: n prop v adj adv art pron prp conj num intj cl par g x

Collapse Tree

Expand Tree

```

graph TD
    Root[A cl] --> N1[αV y]
    Root --> N2[Od g]
    Root --> N3[CO conj]
    Root --> N4[A cl]
    Root --> N5[.]

    N1 --> E1[encadeando]

    N2 --> N2D[D art]
    N2 --> N2H[H n]
    N2 --> N2O[D adj]
    N2D --> E2[os]
    N2H --> E3[objetos]
    N2O --> E4[receptores]

    N3 --> E5[e]

    N4 --> N4V[αV y]
    N4 --> N4S[S g]
    N4 --> N4PRP[αPRP g]
    N4 --> N4C[cl]
    N4 --> N4D[D cl]

    N4V --> E6[passando]

    N4S --> N4SD[D art]
    N4S --> N4SH[H n]
    N4SD --> E7[a]
    N4SH --> E8[solicitação]

    N4PRP --> N4PRPH[H prp]
    N4PRP --> N4PRPD[D g]
    N4PRPH --> E9[ao longo de]
    N4PRPD --> N4PRPDH[D art]
    N4PRPDH --> E10[uma]
    N4PRPD --> N4PRPDH2[H n]
    N4PRPDH2 --> E11[cadeia]

    N4C --> E12[.]

    N4D --> N4DA[A g]
    N4DA --> N4DAH[H prp]
    N4DA --> N4DASUB[SUB conj]
    N4DA --> N4DAS[S g]
    N4DA --> N4DAO[Od pron]
    N4DA --> N4DAP[P v]
    N4DAH --> E13[até]
    N4DASUB --> E14[que]
    N4DAS --> N4DASD[D art]
    N4DAS --> N4DASH[H n]
    N4DASD --> E15[um]
    N4DASH --> E16[objeto]
    N4DAO --> E17[a]
    N4DAP --> E18[trate]
  
```

Analysis 2 of 2

Next

Previous

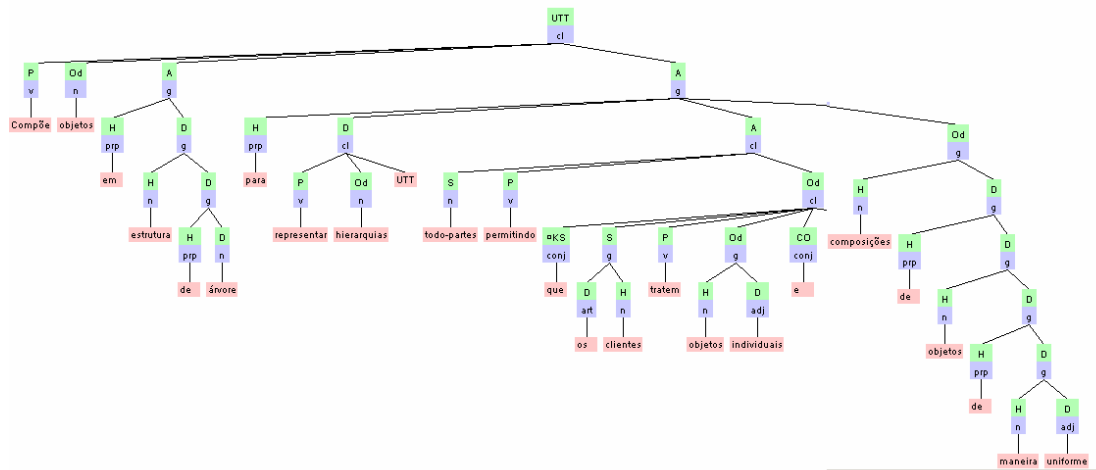
Java Applet Window

Windows taskbar: Iniciar, Validação, VISL - Tree structure..., Propósito\_validação..., VISL - Visual Interacti..., 22:55



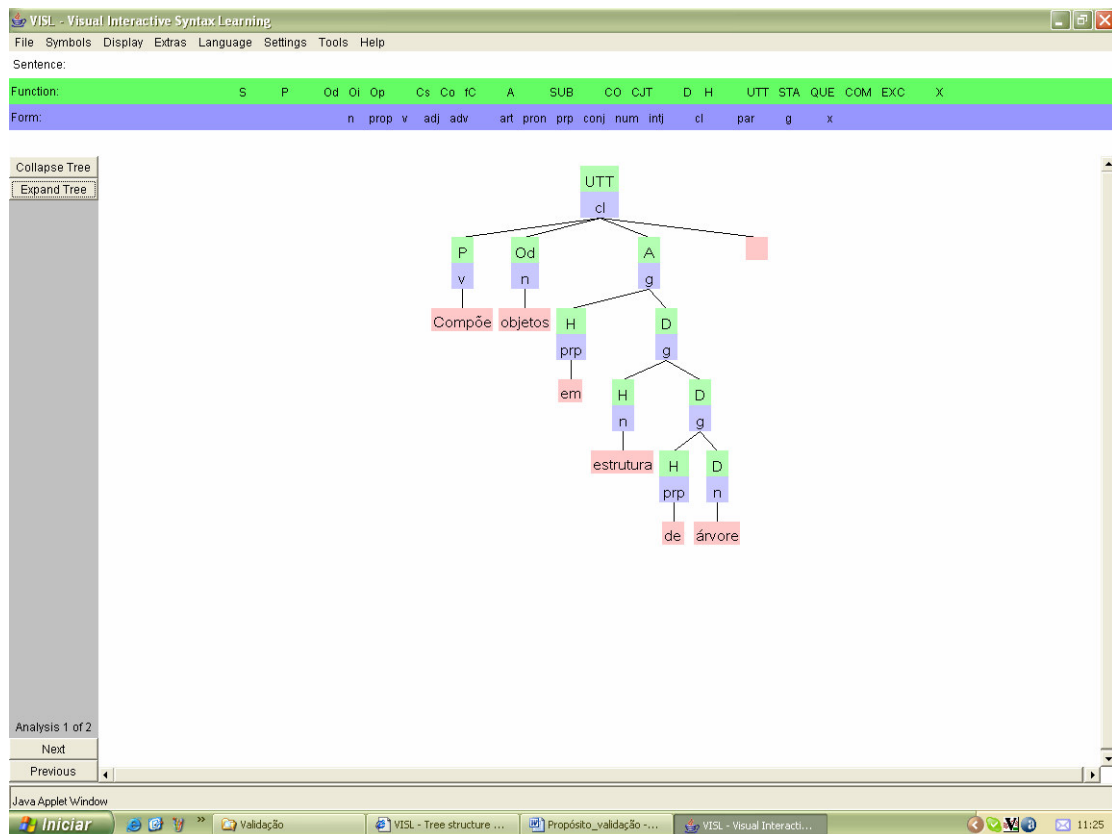
## 07 – Composite

**Original** → Compõe objetos em estrutura de árvore para representar hierarquias todo-partes permitindo que os clientes tratem objetos individuais e composições de objetos de maneira uniforme.

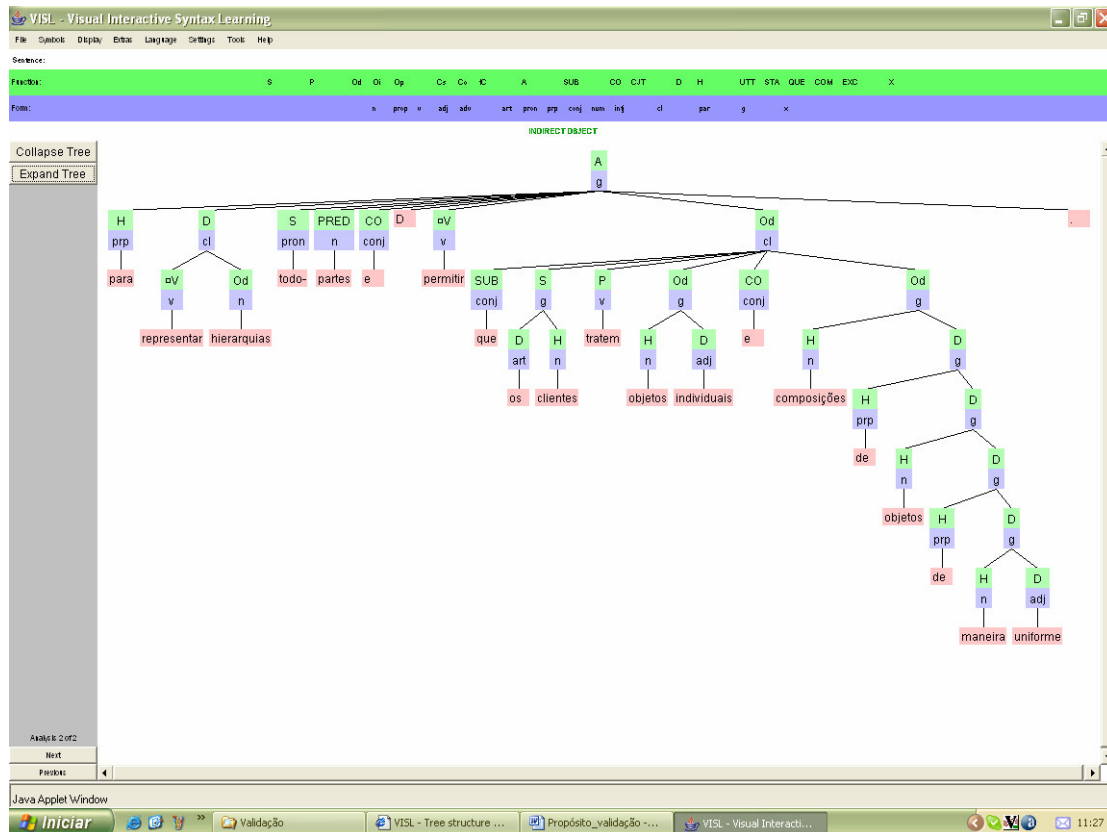


**Tentativa** → Compõe objetos em estrutura de árvore: para representar hierarquias todo-partes e permitir que os clientes tratem objetos individuais e composições de objetos de maneira uniforme.

Parte Inicial (a diminuição para um única figura a torna ilegível)



Parte Final (a diminuição para um única figura a torna ilegível)

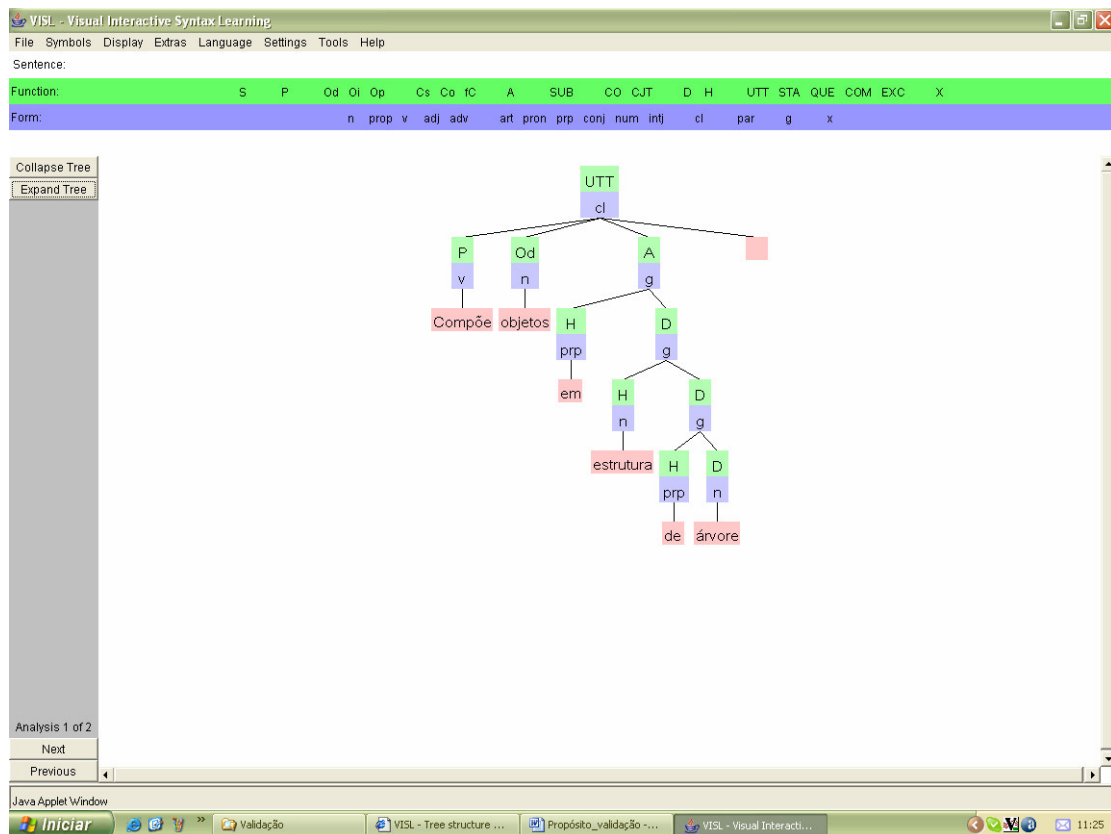


**Solução** → Compõe objetos em estrutura de árvore: para representar hierarquias de agregação e permitir que os clientes tratem objetos individuais e composições de objetos uniformemente.

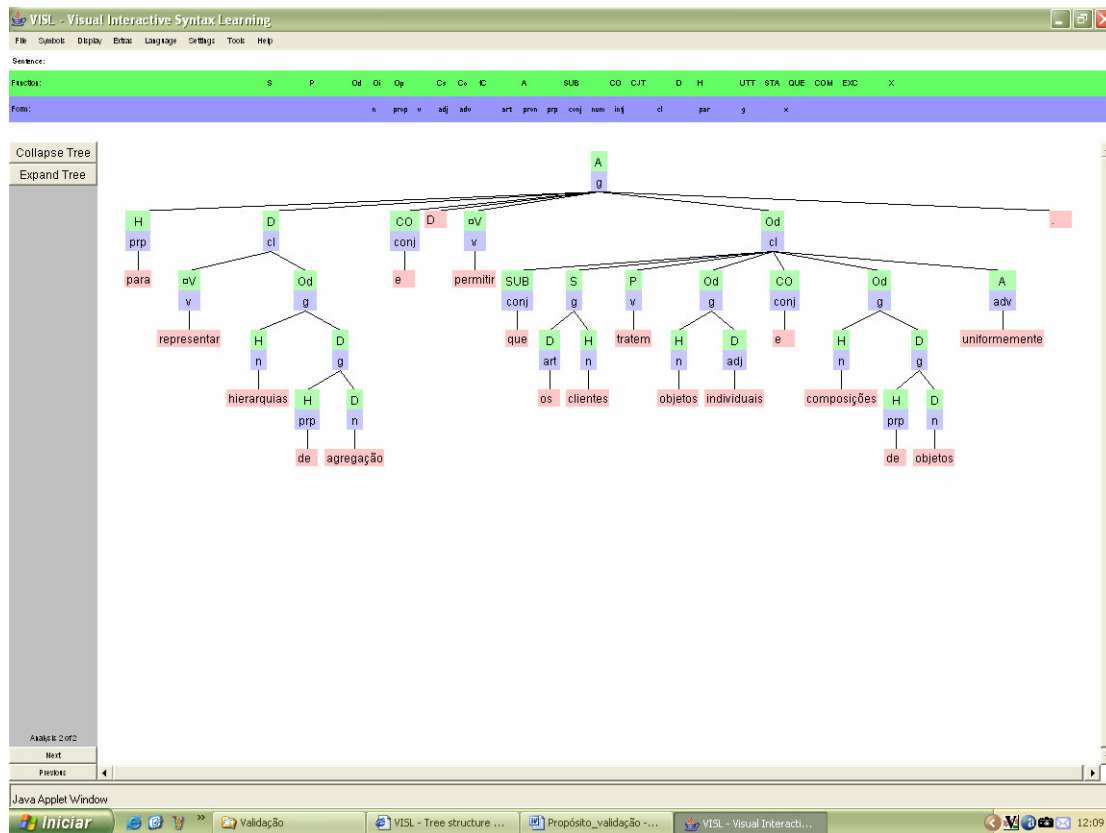
Obs

- “todo-partes” leva a uma representação sintática de pronome e não de adjetivo com era a intenção do redator → a substituição da expressão por um sinônimo resolve
- trocar “de maneira uniforme” por “uniformemente” é mais adequado para representar o modificador do verbo, e pode ser considerado uma boa regra de escrita

Parte Inicial (a diminuição para um única figura a torna ilegível)



Parte Final (a diminuição para um única figura a torna ilegível)

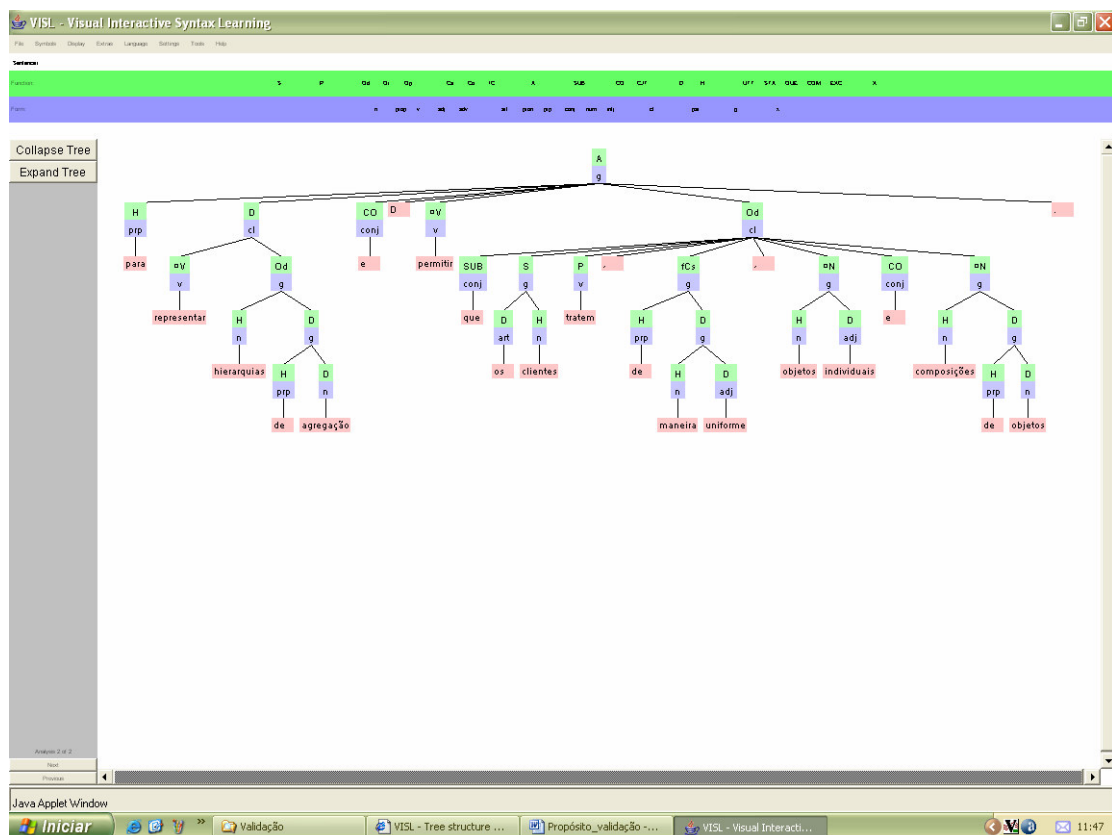


**Alternativa de Solução** → Compõe objetos em estrutura de árvore: para representar hierarquias de agregação e permitir que os clientes tratem, de maneira uniforme, objetos individuais e composições de objetos.

Obs

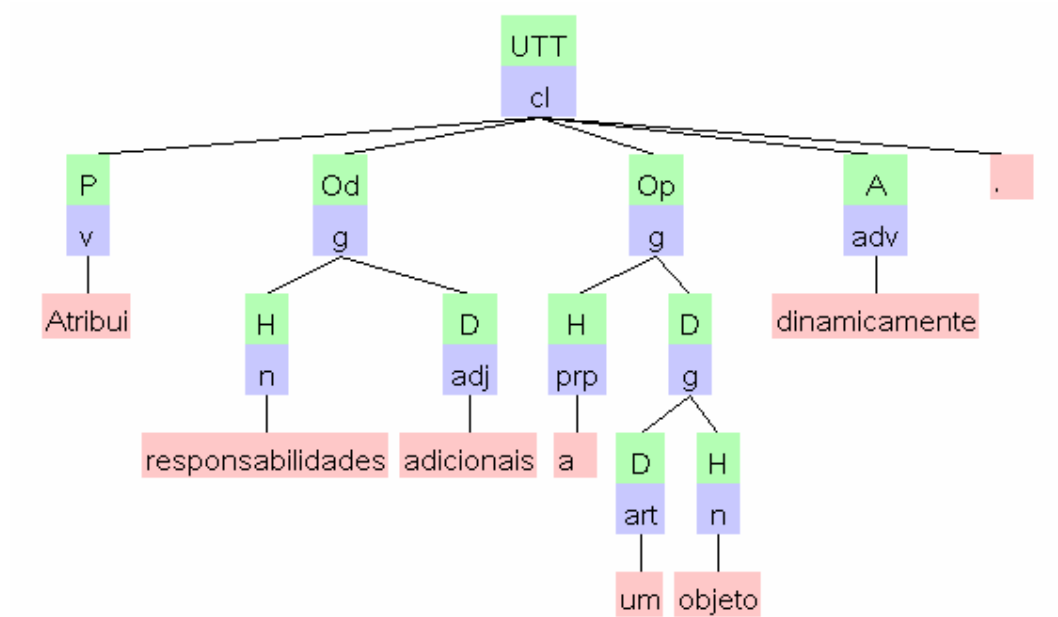
- alternativa pior, pois além de alterar posição do advérbio em relação ao texto original não utiliza a regra de escrita mais legível para representação do advérbio

Parte Final (a diminuição para um única figura a torna ilegível)



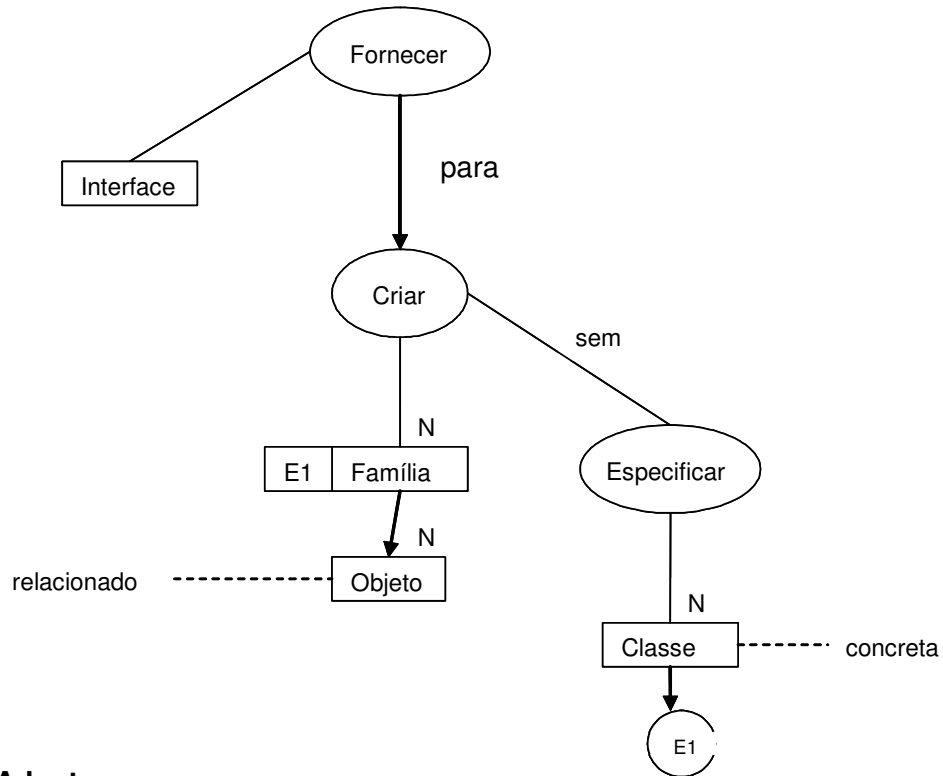
## 08 – Decorator

**Original e Solução** → Atribui responsabilidades adicionais a um objeto dinamicamente.

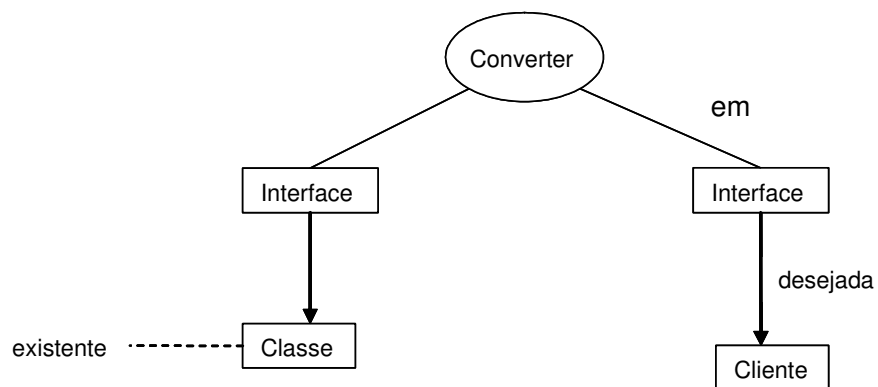


## 7.2 Anexo B – FGRs dos Propósitos dos Padrões de Projeto.

### 01 - Abstract Factory

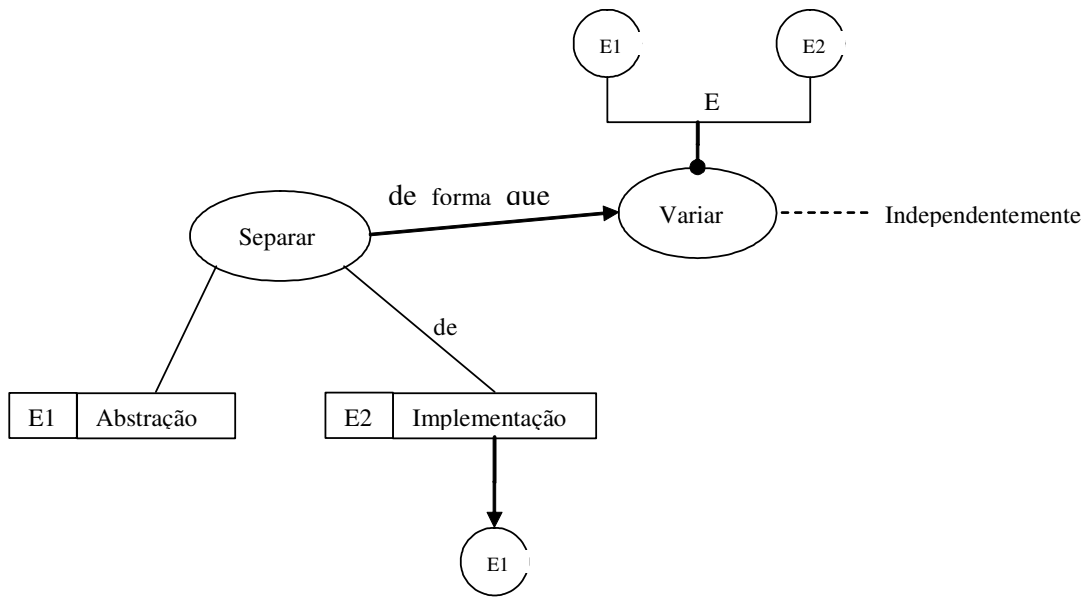


### 02 – Adapter

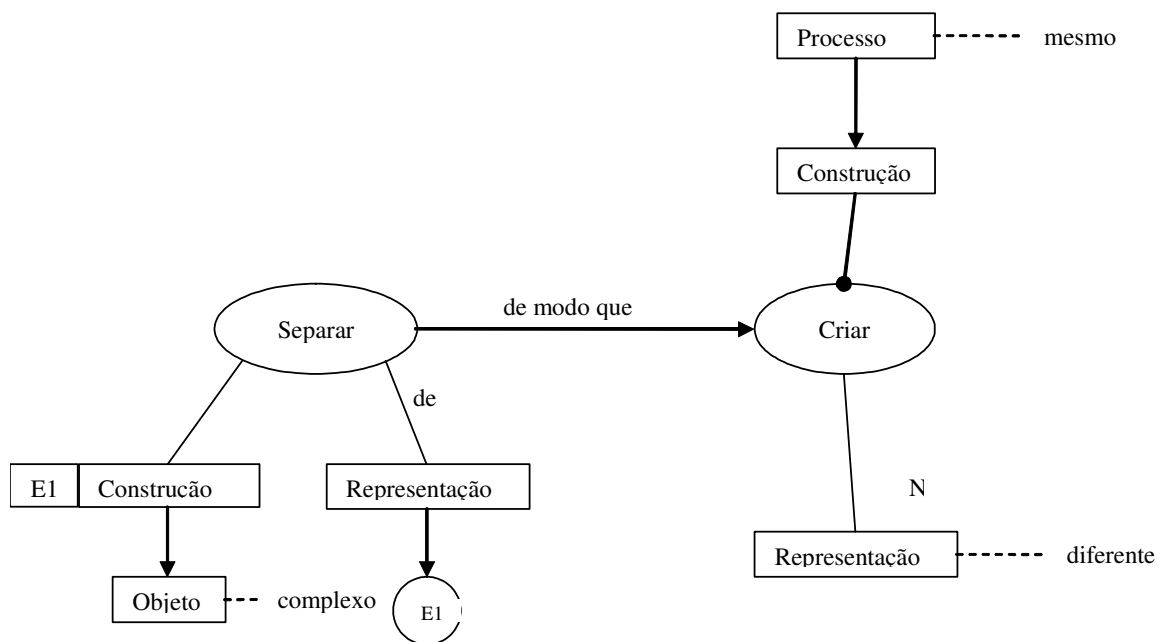




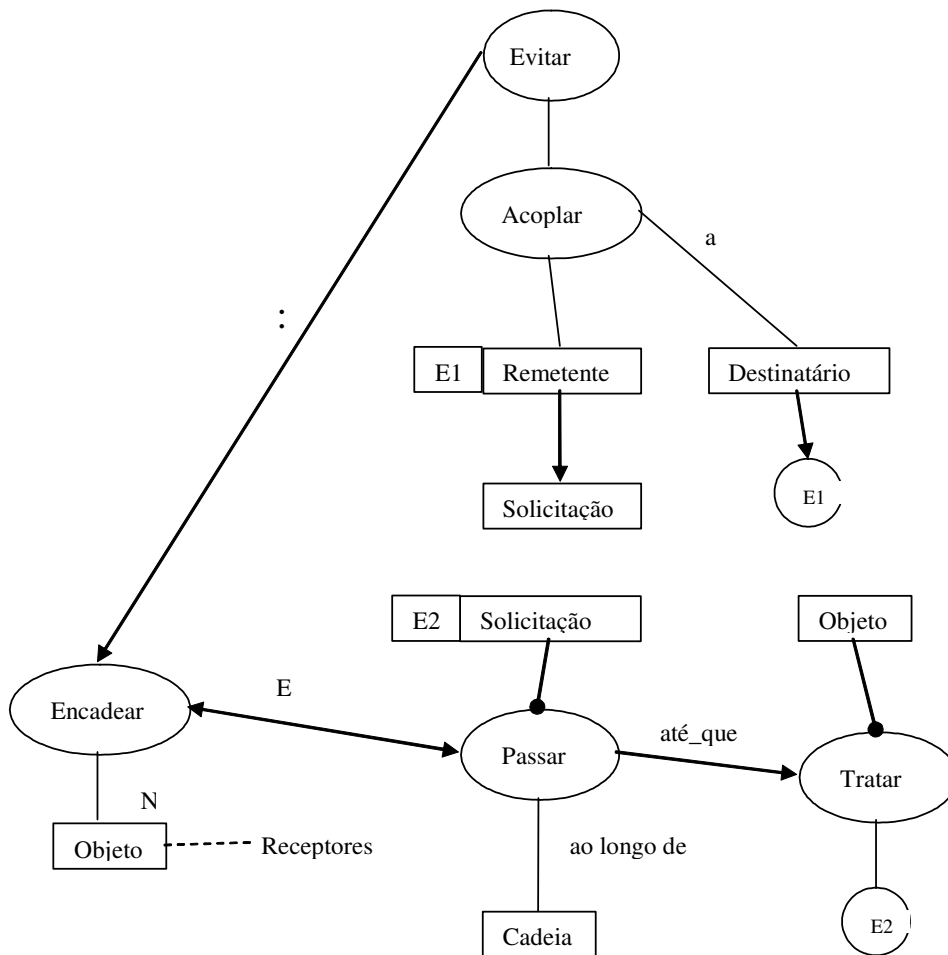
**03 – Bridge**



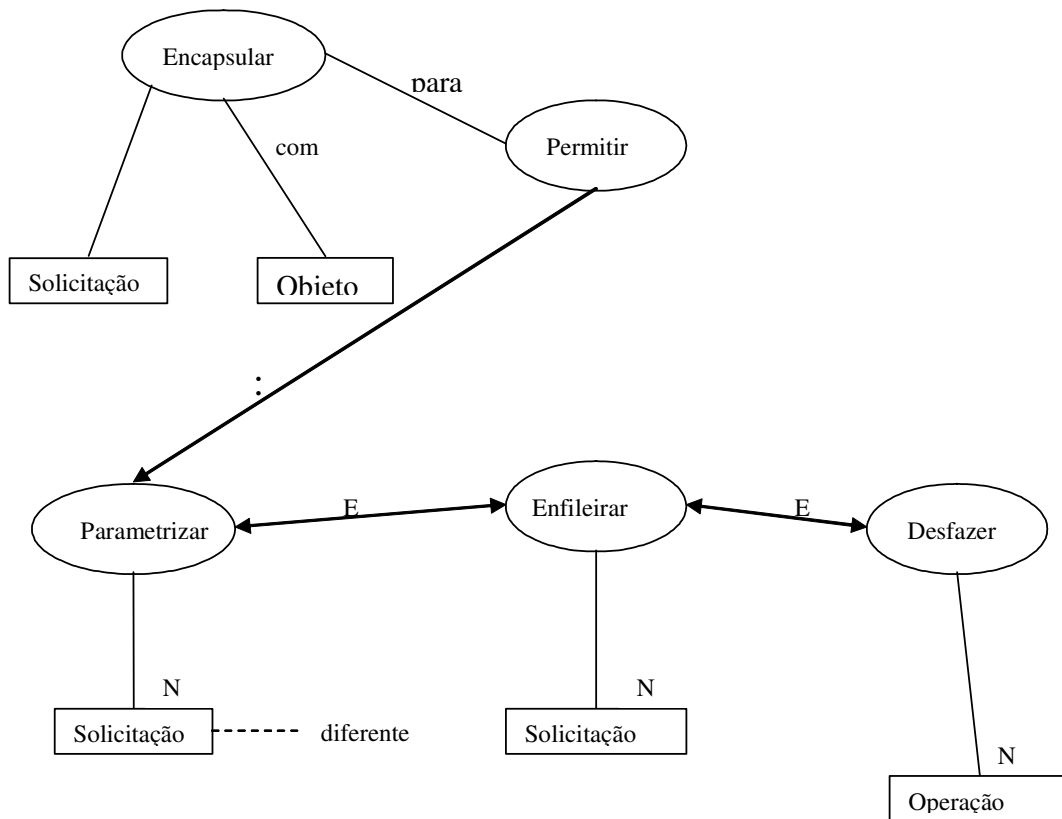
**04 – Builder**



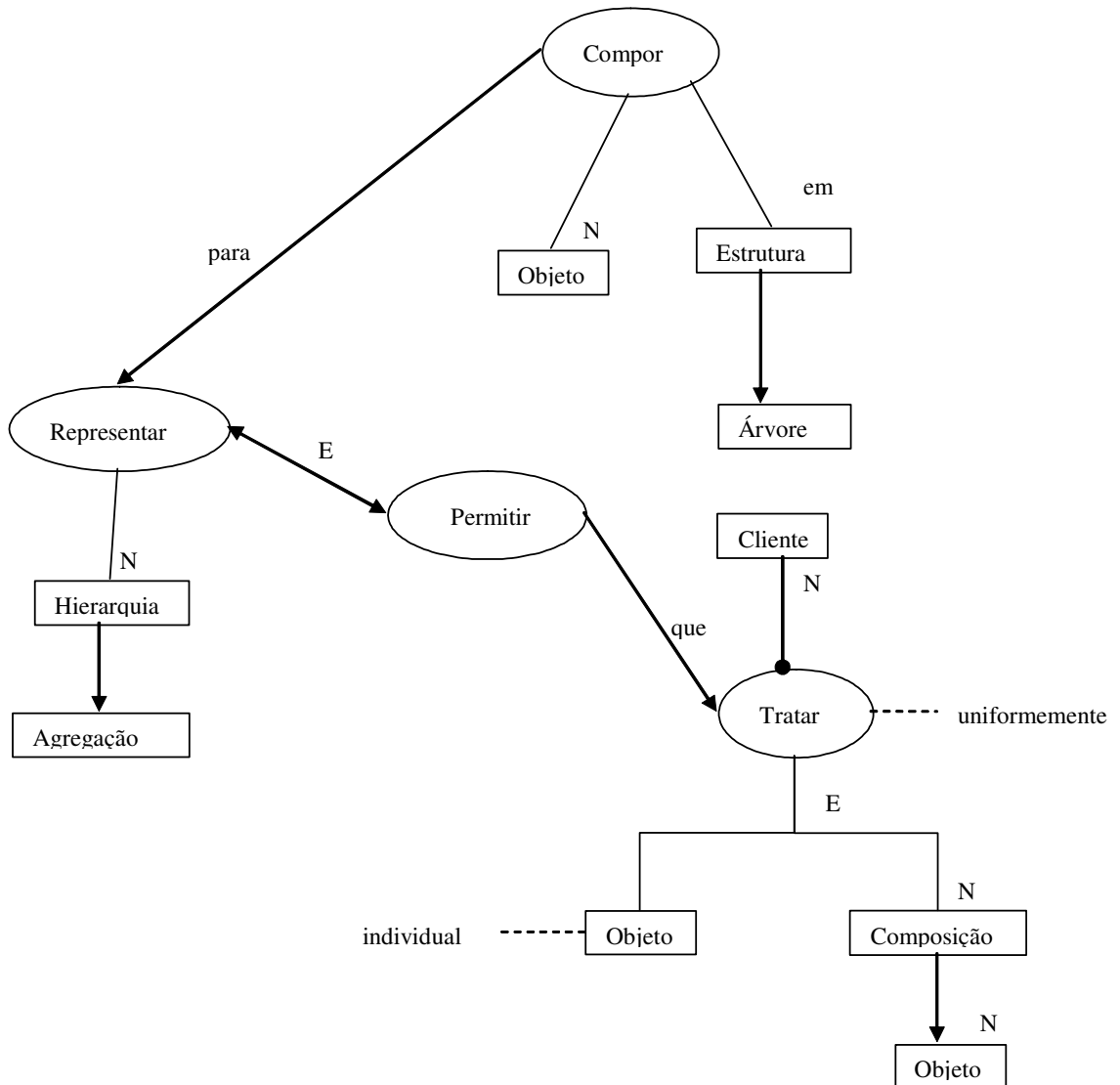
05 – Chain of Responsibility



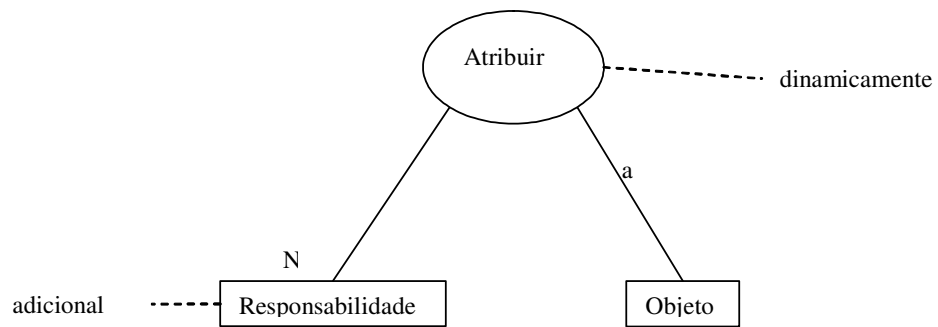
## 06 – Command



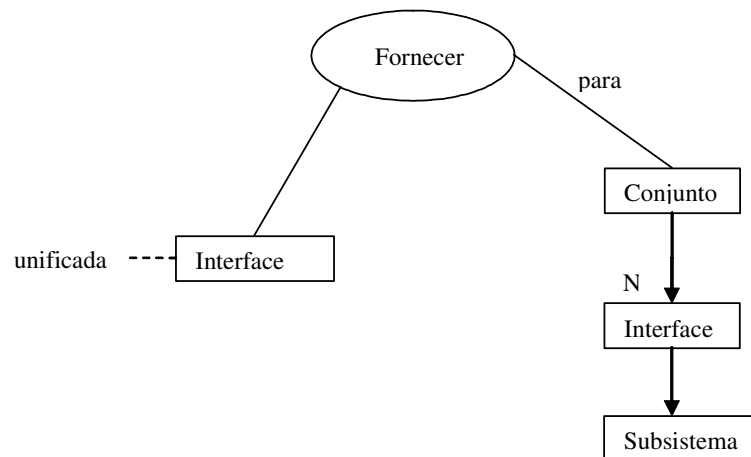
## 07 – Composite



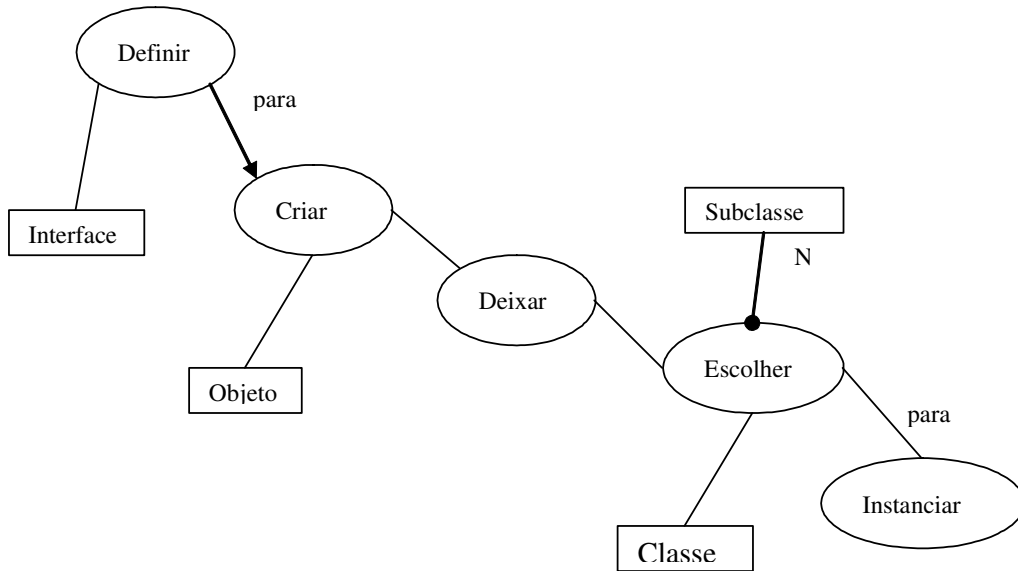
## 08 – Decorator



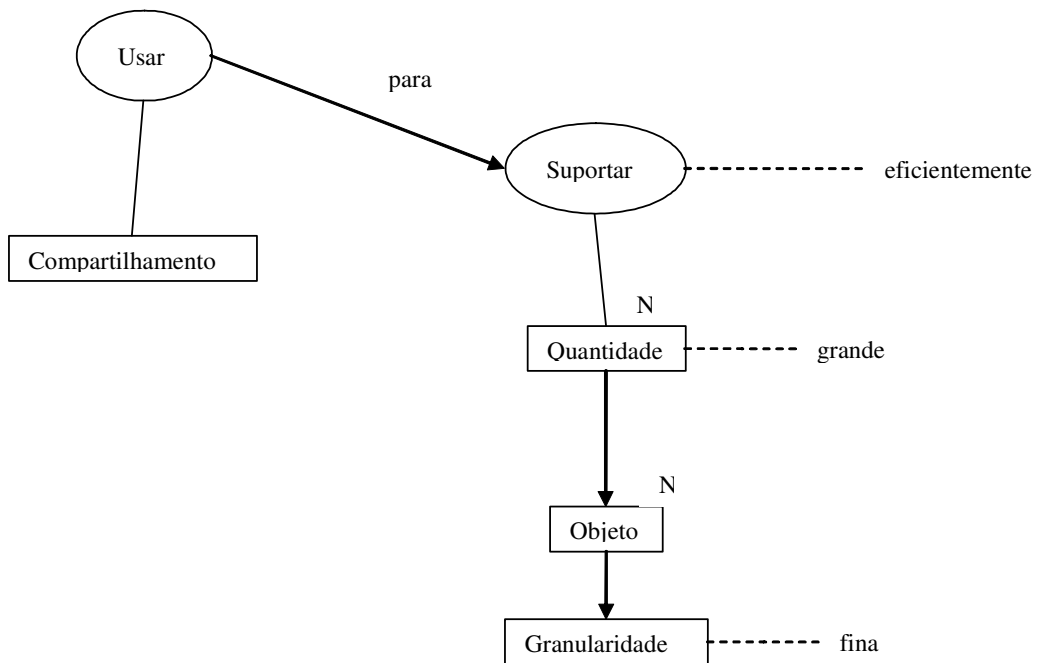
## 09 – Facade



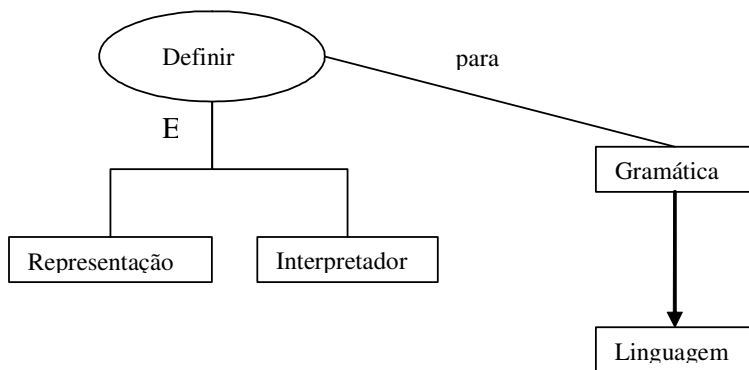
**10 – Factory Method**



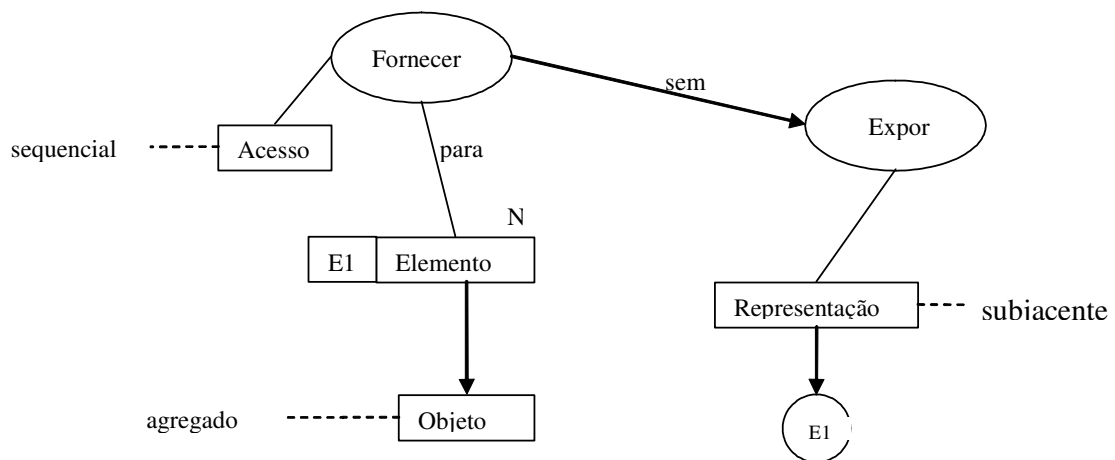
**11 – Flyweight**



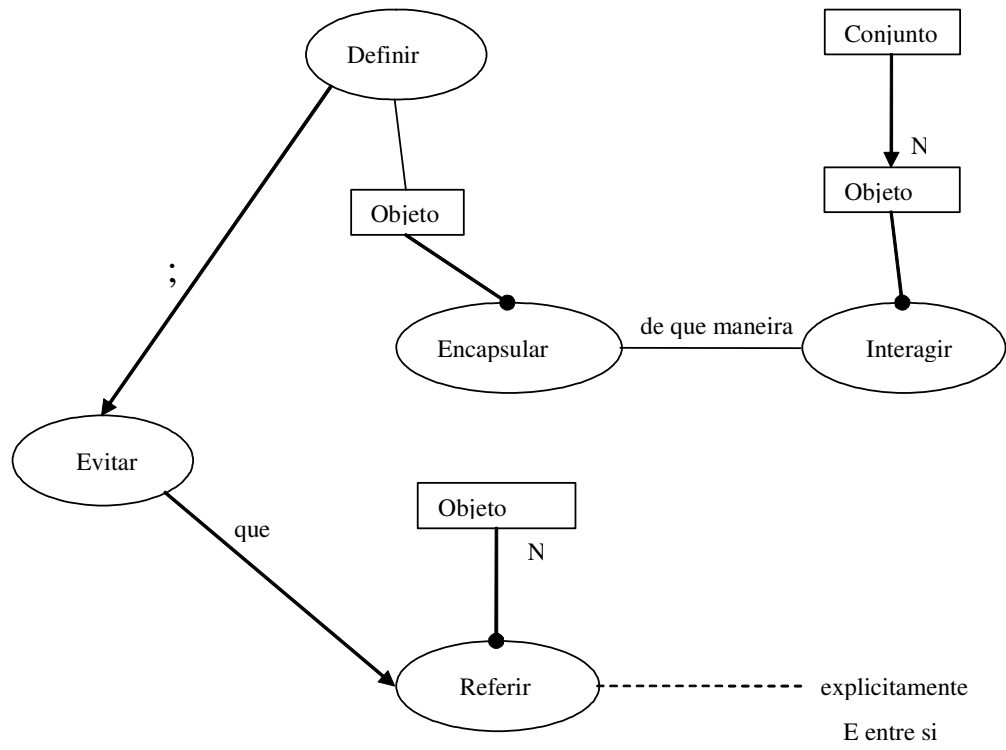
12 – Interpreter



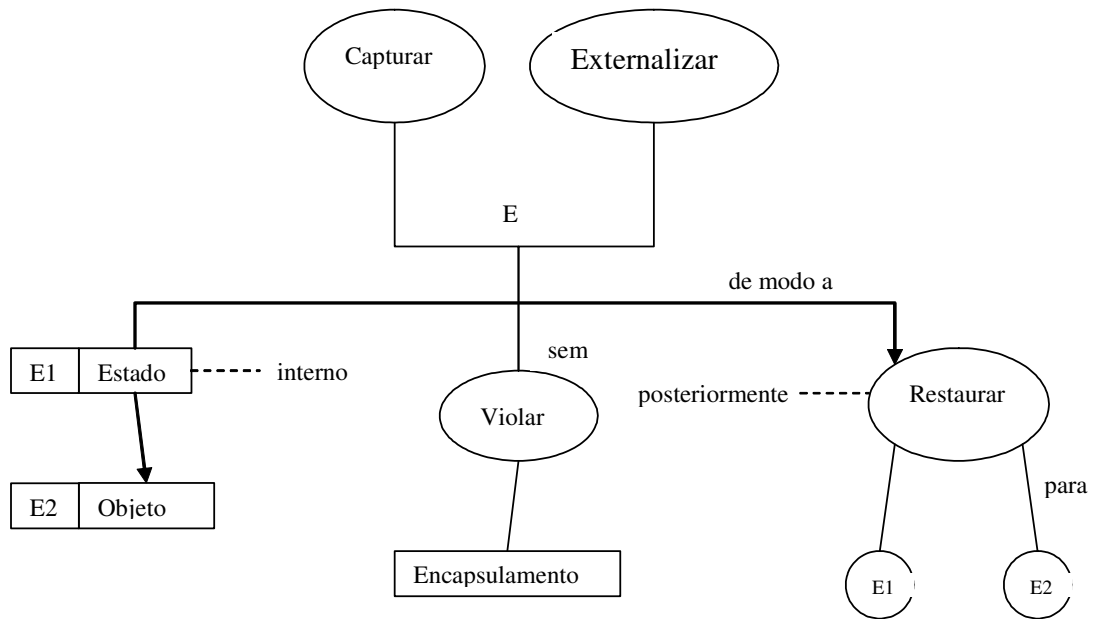
13 – Iterator



14 – Mediator

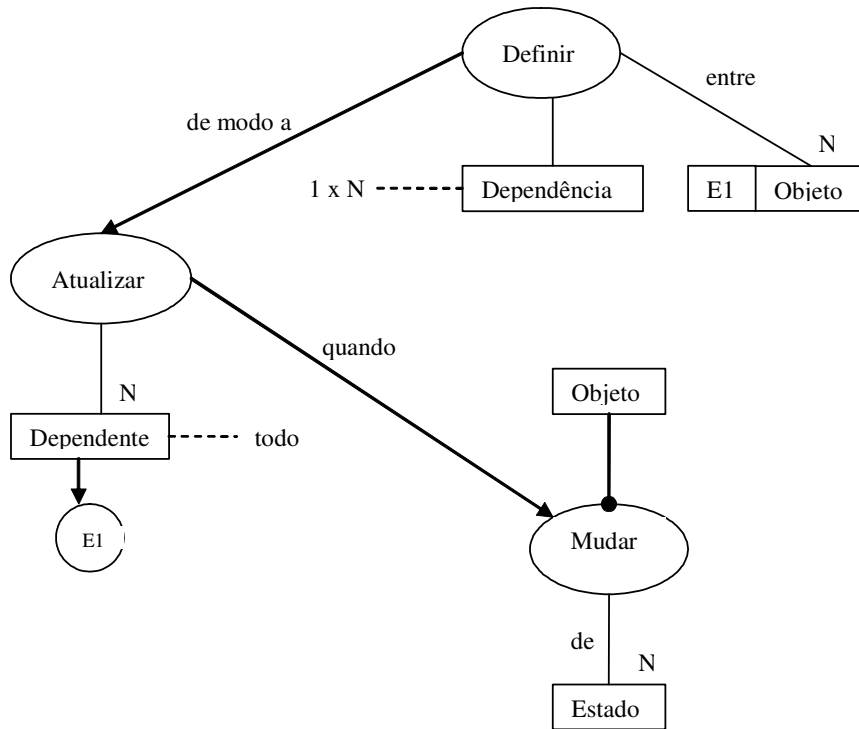


15 – Memento

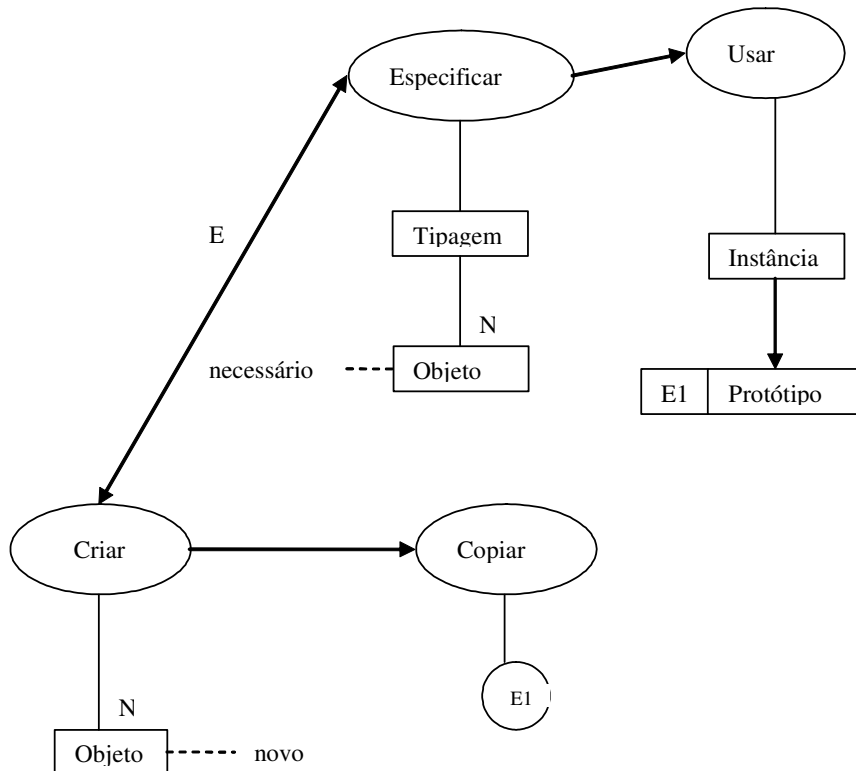




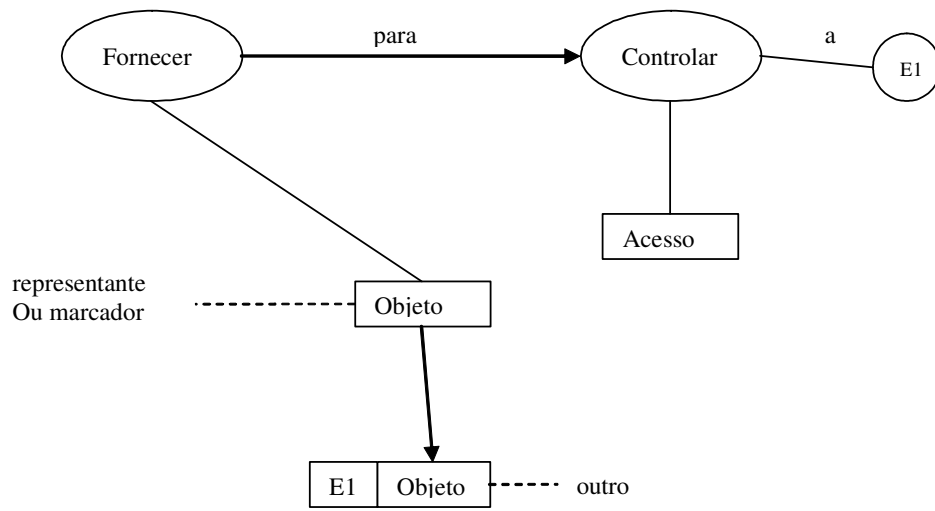
16 – Observer



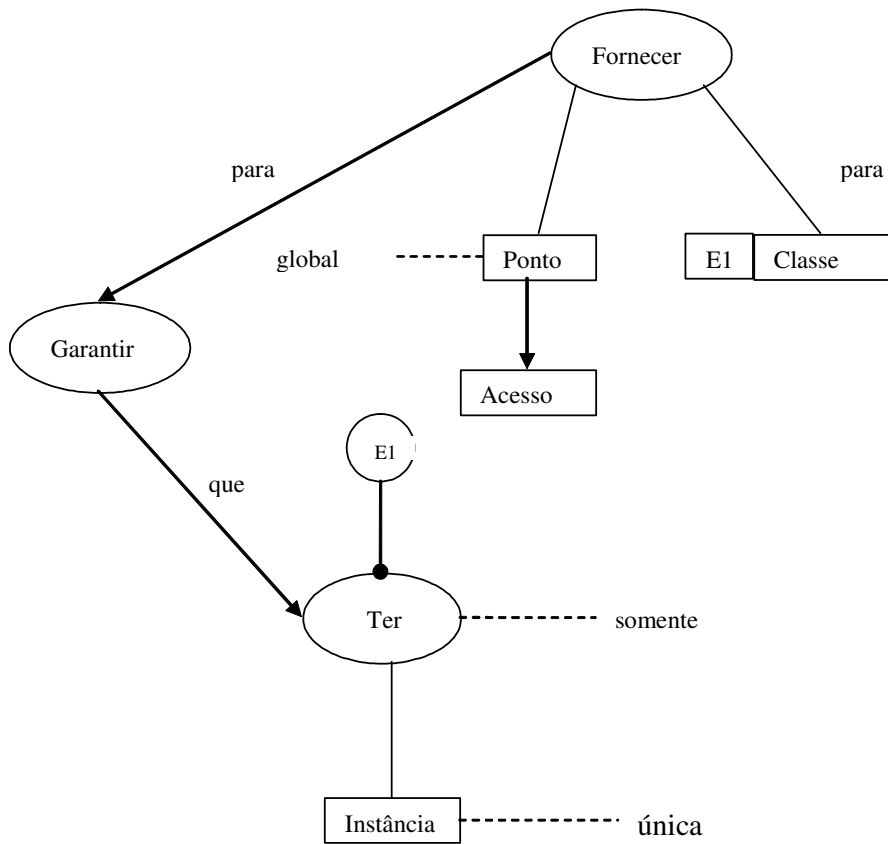
17 – Prototype



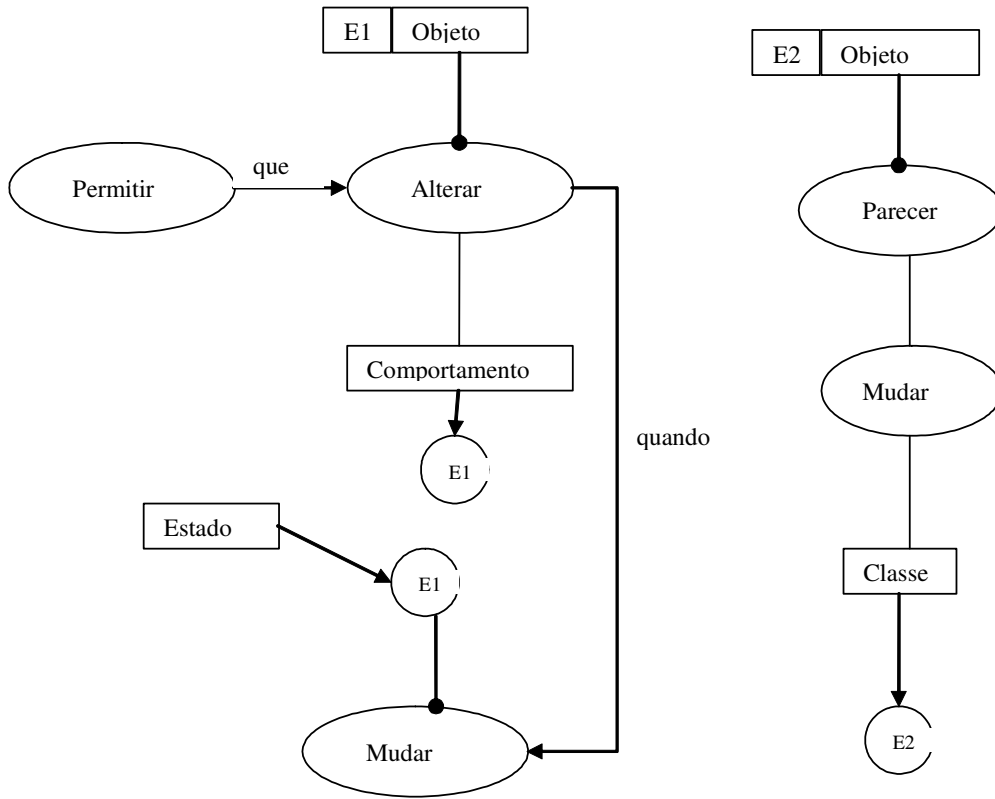
18 – Proxy



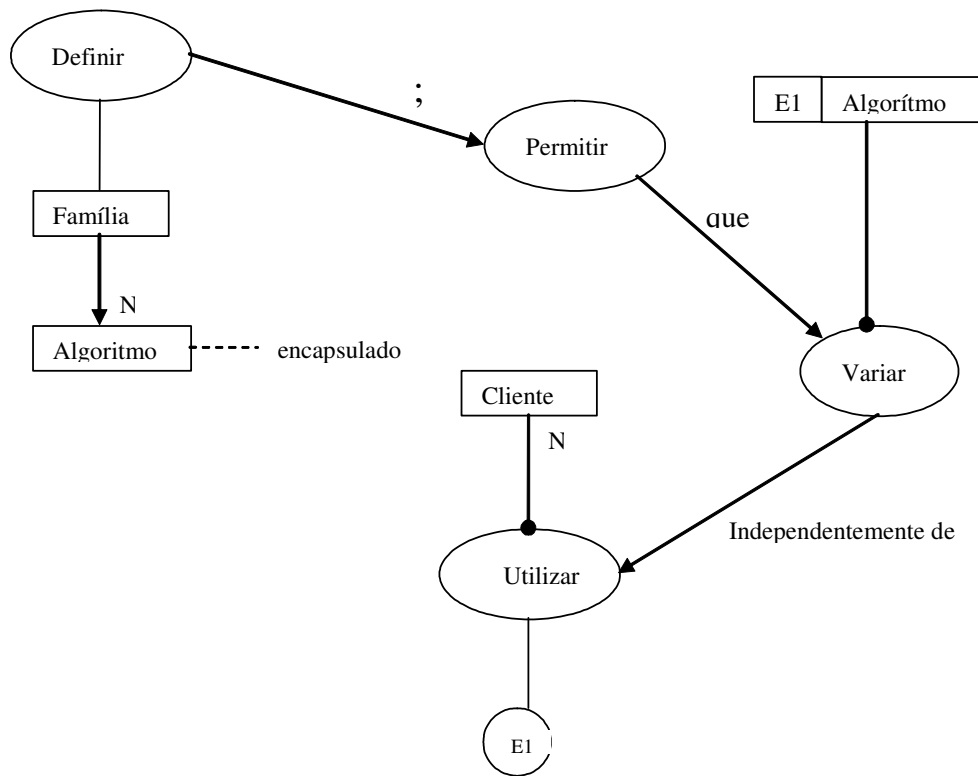
19 – Singleton



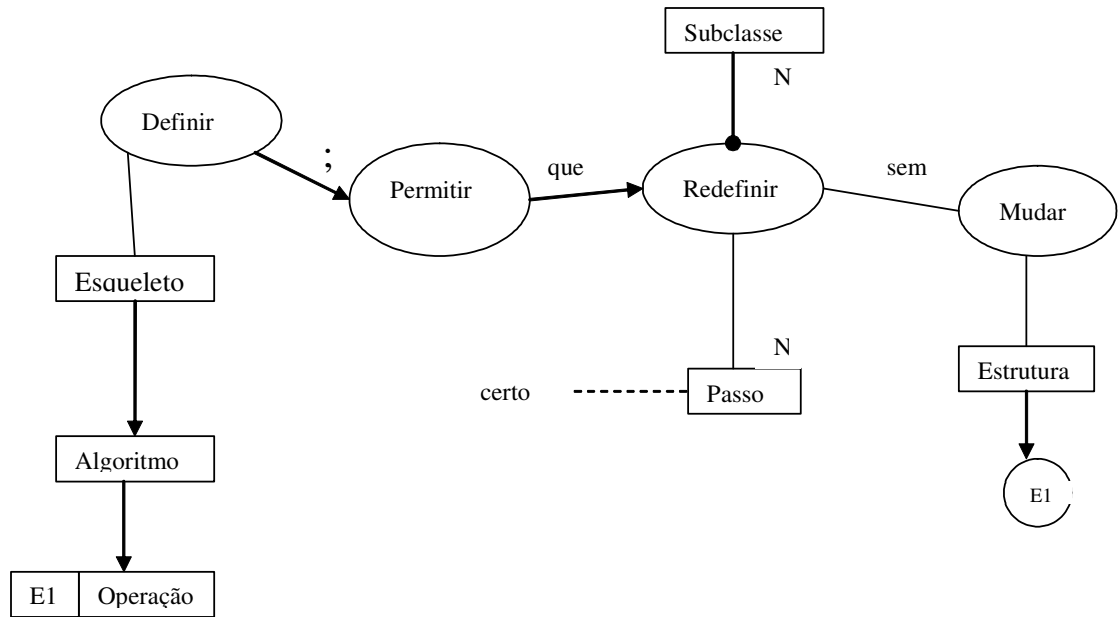
20 – State



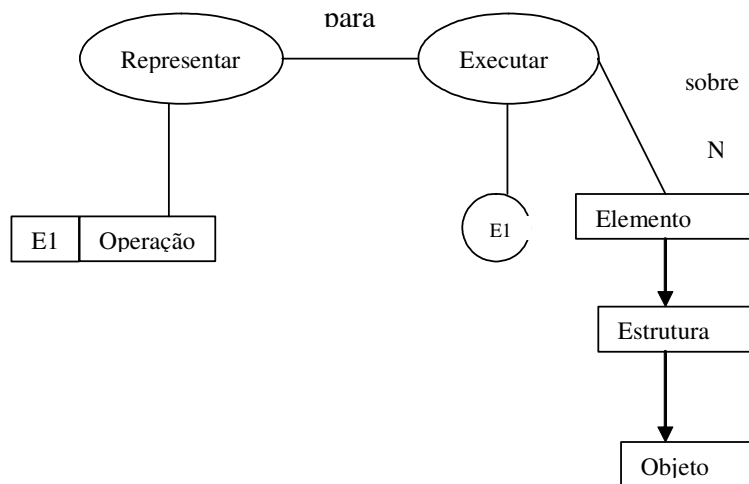
## 21 – Strategy



22 – Template Method



23 – Visitor



### 7.3 Anexo C – Reduções Automatizáveis das FGRs dos Propósitos.

Para cada FGR, de cada Propósito de cada padrão de projeto, são enumeradas as reduções passíveis de automação. Os padrões de projeto, para os quais não se aplica nenhuma redução, foram omitidos.

#### Abstract Factory (1)

- remover Núcleo da Relação “Fornecer”
  - a Entidade “Interface” passa a ser o sujeito do Núcleo da Relação “Criar”
- transformar “sem” para Modificador do Núcleo da Relação “Especificar”

#### Chain of Responsibility (5)

- remover Núcleo da Relação “Evitar”
  - acrescentando o Modificador “não” ao Núcleo da Relação “Acoplar”
  - associando diretamente o Núcleo da Relação “Acoplar” com o Núcleo da Relação “Encadear”

#### Command (6)

- remover o Núcleo da Relação “Permitir”
  - associando diretamente o Núcleo da Relação “Encapsular” aos Núcleo da Relação “Parametrizar”
  - mantendo o label “para” que estava sendo utilizado entre o Núcleo de Relação “Encapsular” e o Núcleo de Relação “Permitir”

#### Composite (7)

- remover o Núcleo da Relação “Permitir”
  - associando diretamente o Núcleo da Relação “Representar” ao Núcleo da Relação “Tratar”
  - mantendo o tipo de associação (coordenação) e o label (“E”) que estava sendo utilizado entre o Núcleo de Relação “Representar” e o Núcleo de Relação “Permitir”

### Factory Method (10)

- remover o Núcleo da Relação “Definir”
  - a Entidade “Interface” passa a ser o sujeito do Núcleo da Relação “Criar”
  - utilizando o label “de modo que” para a nova associação

### Flyweight (11)

- remover o Núcleo da Relação “Usar”
  - a Entidade “Compartilhamento” passa a ser o sujeito do Núcleo da Relação “Suportar”

### Iterator (13)

- converter label “sem” da associação entre o Núcleo da Relação “Acessar” e o Núcleo da Relação “Expor” para Modificador “não” do Núcleo da Relação “Expor”

#### Mediator (14)

- remover o Núcleo da Relação “Evitar”
  - associar Modificador “não” associado ao Núcleo da Relação “Referir”
  - associando diretamente o Núcleo da Relação “Definir” ao Núcleo da Relação “Referir”
  - utilizando o label “de modo que” para a nova associação
- remover o Núcleo da Relação “Definir”
  - associando diretamente o Núcleo da Relação “Encapsular” ao Núcleo da Relação “Referir”
  - mantendo o label “de modo que” para a associação

#### Memento (15)

- converter label “sem” da associação entre o Núcleo da Relação “Capturar” e o Núcleo da Relação “Violar” para Modificador “não” do Núcleo da Relação “Violar”

#### Proxy (18)

- remover o Núcleo da Relação “Fornecer”
  - o Sintagma “Objeto (representante OU marcador) de Objeto (outro)” passa a ser Núcleo da Relação “Controlar”
  - mantendo o label da associação “para”

#### Singleton (19)

- remover Núcleo da Relação “Garantir”
  - associando diretamente o Núcleo da Relação “Fornecer” ao Núcleo da Relação “Ter”
  - nova associação “para que” agrupa os labels das associações anteriores “para” e “que”

#### State (20)

- remover Núcleo da Relação “Permitir”
- remover Núcleo de Relação “Parecer”
  - associar Modificador “não” associado ao Núcleo da Relação “Mudar”
  - associando diretamente a Entidade “Objeto” ao Núcleo da Relação “Mudar”

#### Strategy (21)



- remover o Núcleo da Relação “Permitir”
  - associando diretamente o Núcleo de Relação “Definir” ao Núcleo da Relação “Variar”
  - utilizando o label “de modo que” para a nova associação

#### Template Method (22)

- remover o Núcleo da Relação “Permitir”
  - associando diretamente o Núcleo de Relação “Definir” ao Núcleo da Relação “Redefinir”
  - utilizando o label “de modo que” para a nova associação

Visitor (23)

- remover o Núcleo da Relação “Representar”
  - associando diretamente a Entidade “Operação” ao Núcleo da Relação “Executar”

A análise, das reduções descritas acima, resulta nas heurísticas de redução descritas a seguir. Para cada tipo de redução encontrada são definidos:

- o tipo da redução;
- as condições necessárias para que a aplicação da redução seja possível;
- a alteração, na representação original, decorrente da aplicação da redução;
- os padrões de projeto aos quais a redução se aplica.

Redução 1 : Remoção do Núcleo de Relação ligado a um Núcleo de Relação

- condições necessárias
  - Núcleo de Relação inicial (não subordinado a nenhum outro Núcleo de Relação)
  - Núcleo de Relação ligado a somente a um único Núcleo de Relação
- alteração na representação
  - Núcleo de Relação inicial é simplesmente removido
    - se o Núcleo de Relação removido tiver conotação de negação, um Modificador de negação é associado ao seu Núcleo de Relação subordinado (que permaneceu na representação)
- aplicável ao padrão: 20

Redução 2 : Remoção do Núcleo de Relação ligado a um segundo Núcleo de Relação e à Entidade Objeto Direto

- condições necessárias
  - Núcleo de Relação inicial (não subordinado a nenhum outro Núcleo de Relação e sem Entidade Sujeito)
  - Núcleo de Relação ligado a somente a uma Entidade (ou Sintagma) e a um outro Núcleo de Relação
  - Entidade atua com Objeto Direto do Núcleo de Relação
  - segundo Núcleo de Relação não tenha Entidade associado como sujeito
- alteração na representação
  - Núcleo de Relação inicial é removido
  - Entidade (ou Sintagma) passa a ser o Sujeito do outro Núcleo de Relação
- aplicável aos padrões: 1 – 10 – 11 – 18 – 23

Redução 3 : Remoção do Núcleo de Relação ligado a um segundo Núcleo de Relação e à Entidade Sujeito de um terceiro Núcleo de Relação

- condições necessárias
  - Núcleo de Relação inicial (não subordinado a nenhum outro Núcleo de Relação e sem Entidade Sujeito)
  - Núcleo de Relação ligado a somente a uma Entidade (ou Sintagma) e a um terceiro Núcleo de Relação
  - Núcleo de Relação não tenha Entidade associado como sujeito
  - Entidade atua como Sujeito do terceiro Núcleo de Relação
- alteração na representação
  - Núcleo de Relação inicial é removido
  - terceiro Núcleo de Relação passa a ser associado ao segundo Núcleo de Relação
  - o label da nova associação será o mesmo label da associação que existia entre o Núcleo de Relação removido e o segundo Núcleo de Relação
- aplicável ao padrão: 14

Redução 4 : Remoção de Núcleo de Relação ligado a dois Núcleos de Relações

- condições necessárias
  - Núcleo de Relação ligado a somente dois outros Núcleos de Relações
- alteração na representação
  - Núcleo de Relação é removido
    - se o Núcleo de Relação removido tiver conotação de negação, um Modificador de negação é associado ao seu Núcleo de Relação subordinado (que permaneceu na representação)
  - os outros dois Núcleos de Relações passam a ser associados, com base nas associações que tinham com o Núcleo de Relação removido
- aplicável aos padrões: 5 – 6 – 7 – 10 – 14 – 19 – 21 – 22

Redução 5 : Remoção do Núcleo de Relação auxiliar de Núcleo de Relação composto

- condições necessárias
  - Núcleo de Relação auxiliar de outro Núcleo de Relação principal
  - Núcleo de Relação auxiliar tem conotação passível de atuar como Modificador do Núcleo de Relação principal
- alteração na representação
  - Núcleo de Relação auxiliar de Núcleo de Relação composto é removido
  - passando conotação do Núcleo de Relação auxiliar para Modificador do Núcleo de Relação principal
    - observe que a manutenção de verbo composto auxiliar na geração da FGR, só se justifica quando o verbo auxiliar tem relevância semântica e

portanto não pode ser descartado; sua remoção na redução só se justifica porque sua conotação semântica esta sendo repassada para o verbo principal (que permanece)

- aplicável aos padrão: 20

Redução 6 : Conversão de label de associação com conotação de negação

para Modificador de negação do respectivo Núcleo de Relação

- condições necessárias
  - label de associado com Núcleo de Relação
  - label com conotação de negação
- alteração na representação
  - label da associação é removido
  - conotação de negação do label é convertida em Modificador do Núcleo de Relação que era afetado pelo label
- aplicável aos padrões: 1 – 13 – 15 – 22

#### **7.4 Anexo D – Análise das Alterações de Redação para gerar Sugestões para os Usuários.**

Neste anexo, são analisadas as alterações de redação, descritas no Anexo “AS Base para gerar FGRs dos Propósitos”. O objetivo desta análise é o de determinar as sugestões de redação para o Bibliotecário e para o Usuário da Biblioteca. Para cada padrão de projeto foram identificados:

- a redação original da faceta “Propósito”;
- a redação final que soluciona o mapeamento da AS para a FGR que captura adequadamente a referida faceta, buscando gerar o mínimo de alterações no texto original;
- quais os problemas encontrados na redação original, e qual a solução proposta para cada problema encontrado.

**Nota →** Até o presente momento foram analisados os padrões de projeto 1 a 12. Os padrões 13 a 23 deverão ser analisados antes da conclusão deste trabalho.

## 1 - Abstract Factory

### Redação Original

- Fornece uma interface para a criação de famílias de objetos relacionados sem especificar suas classes concretas.

### Redação Final

- Fornece uma interface para criar famílias de objetos relacionados, sem especificar suas classes concretas.

### Problema 1.1

- o *parser* associou “Fornece uma interface de famílias de objetos relacionados”, mas a associação correta é “para criação de famílias de objetos relacionados”

### Solução 1.1

- trocar “para criação de” por “para criar”

### Problema 1.2

- o *parser* associou “objetos relacionados sem especificar suas classes concretas”, mas a associação correta é “Fornecer ... sem especificar suas classes concretas”

### Solução 1.2

- delimitar “de objetos relacionados” entre vírgulas

## 2 – Adapter

### Redação Original

- Converte a interface de uma classe a ser utilizada em uma interface desejada pelo cliente.

### Redação Final

- Converte a interface de uma classe existente na interface desejada pelo cliente.

### Problema 2.1

- o *parser* associou “classe a ser utilizada em uma interface desejada pelo cliente”, mas a associação correta é “converte a interface (a ser utilizada) em uma interface (desejada pelo cliente)”

### Solução 2.1

- trocar “classe a ser utilizada” por “classe existente”

### Observação

- “interface desejada pelo cliente” é um sintagma no qual o verbo no particípio (desejada) qualifica a interface distinguindo-a de “interface existente”, portanto, deve ser mantido conservando a forma original
  - observe que representar como “interface do cliente” não é adequado, porque a expressão poderia ser “interface rejeitada pelo cliente”

### **3 – Bridge**

#### **Redação Original e Final**

- Separa uma abstração de sua implementação de forma que as duas possam variar independentemente.



## 4 – Builder

### Redação Original

- Separa a construção de um objeto complexo de sua representação, de modo que o mesmo processo de construção possa criar diferentes representações.

### Redação Final

- Separa de sua representação a construção de um objeto complexo, de modo que o mesmo processo de construção possa criar diferentes representações.

### Problema 4.1

- o *parser* associou “Separa a construção de um objeto complexo de sua representação”, mas a associação correta é “Separa a construção de um objeto complexo de sua representação”

### Solução 4.1

- mover o objeto indireto “de sua representação” para suceder o verbo “Separa” antecedendo o objeto direto “a construção de um objeto complexo”

### Observação

- a vírgula antecedendo “de modo que” é desnecessária

## 5 – Chain of Responsibility

### Redação Original

- Evita o acoplamento do remetente de uma solicitação ao seu destinatário, encadeando os objetos receptores, passando a solicitação ao longo de uma cadeia até que um objeto a trate.

### Redação Final

- Evita acoplar o remetente de uma solicitação ao seu destinatário; encadeando os objetos receptores e passando a solicitação ao longo de uma cadeia, até que um objeto a trate.

### Problema 5.1

- o *parser* associou “Evita o acoplamento do remetente de uma solicitação ao seu destinatário”, mas a associação correta é “acoplamento do remetente de uma solicitação ao seu destinatário”

### Solução 5.1

- trocar “Evita o acoplamento do remetente” por “Evita acoplar o remetente”

### Problema 5.2

- o *parser* associou “passando a solicitação ao longo de uma cadeia até que um objeto a trate”, mas a associação correta é “passando a solicitação ao longo de uma cadeia até que um objeto a trate”

### Solução 5.2

- separa primeira oração “Evita ... destinatário” da segunda oração “encadeando ... receptores” através do uso de dois pontos
- troca vírgula entre a segunda oração “encadeando ... receptores” e a terceira oração “passando ... cadeia” por conjunção coordenada “e”
- acrescenta vírgula entre a terceira oração “passando ... cadeia” e a quarta oração “até que um objeto a trate”

## 6 – Command

### Redação Original

- Encapsula uma solicitação como um objeto, permitindo parametrizar diferentes solicitações, enfileiramento das solicitações e o desfazimento das operações.

### Redação Final

- Encapsula uma solicitação como um objeto, permitindo parametrizar diferentes solicitações, enfileirar as solicitações e desfazer as operações.

### Problema 6.1

- o *parser* associou “parametrizar diferentes solicitações enfileiramento das solicitações e o desfazimento das operações”, mas a associação correta é “parametrizar diferentes solicitações, enfileiramento das solicitações e o desfazimento das operações”

### Solução 6.1

- acrescenta “:” após para separar a oração “permitindo” da oração “parametrizar diferentes solicitações”
- troca “enfileiramento das” por “enfileirar as”
- troca “o desfazimento das” por “desfazer as”

## 7 – Composite

### Redação Original

- Compõe objetos em estrutura de árvore para representar hierarquias todo-partes permitindo que os clientes tratem objetos individuais e composições de objetos de maneira uniforme.

### Redação Final

- Compõe objetos em estrutura de árvore; para representar hierarquias de agregação e permitir que os clientes tratem objetos individuais e composições de objetos uniformemente.

### Problema 7.1

- o *parser* associou “para representar hierarquias todo-partes permitindo que os clientes tratem objetos individuais”, mas a associação correta é “para representar hierarquias todo-partes permitindo que os clientes tratem objetos individuais”

### Solução 7.1

- trocar “hierarquias todo-partes” por “hierarquias de agregação”

### Problema 7.2

- o *parser* associou “para representar hierarquias de agregação permitindo que ...”, mas a associação correta é “para representar hierarquias de agregação permitindo que ..”

### Solução 7.2

- acrescentar dois pontos após “Compõe objetos em estrutura de árvore”
- trocar “permitindo” por “permitir”

**Problema 7.3**

- o *parser* associou “os clientes tratem objetos individuais e composições de objetos de maneira uniforme”, mas a associação correta é “os clientes tratem objetos individuais e composições de objetos de maneira uniforme”

**Solução 7.3**

- trocar “de maneira uniforme” por “uniformemente”

**8 – Decorator****Redação Original e Final**

- Atribui responsabilidades adicionais a um objeto dinamicamente.

## 9 – Facade

### Redação Original

- Fornece uma interface unificada para um conjunto de interfaces em um subsistema.

### Redação Final

- Fornece para um conjunto de interfaces de um subsistema uma interface unificada.

### Problema 9.1

- o *parser* associou “Fornece uma interface unificada para um conjunto de interfaces”, mas a associação correta é “Fornece uma interface unificada para um conjunto de interfaces”

### Solução 9.1

- mover o adjunto adverbial “para um conjunto de interfaces em um subsistema” para suceder o verbo “Fornece” antecedendo o objeto direto “uma interface unificada”

### Problema 9.2

- o *parser* associou “para um conjunto de interfaces em um sistema”, mas a associação correta é “para um conjunto de interfaces em um sistema”

### Solução 9.2

- trocar “interfaces em um sistema” por “interfaces de um sistema”

## 10 – Factory Method

### Redação Original

- Define uma interface para criar um objeto deixando para as subclasses decidirem qual classe a ser instanciada.

### Redação Final

- Define uma interface para criar um objeto, deixando que as subclasses escolham a classe para instanciar.

### Problema 10.1

- o *parser* associou “Define uma interface para criar um objeto deixando para”, mas a associação correta é “Define uma interface para criar um objeto deixando para”

### Solução 10.1

- separar a oração “Define uma interface para criar um objeto” da oração “deixando para ...” por vírgula

### Problema 10.2

- o *parser* associou “as subclasses decidirem qual classe a ser instanciada”, mas a associação correta é “as subclasses decidirem qual classe a ser instanciada”
- **Analisar** → se isto se constitui de fato em um problema

### Solução 10.2

- trocar “qual classe a ser instanciada” por “a classe para instanciar”

## 11 – Flyweight

### Redação Original

- Usa o compartilhamento para suportar grandes quantidades de objetos, de granularidade fina, de maneira eficiente.

### Redação Final

- Usa o compartilhamento para suportar grandes quantidades de objetos de granularidade fina eficientemente.

### Problema 11.1

- o *parser* associou “suportar grandes quantidades de objetos, de granularidade fina, de maneira eficiente”, mas a associação correta é “suportar grandes quantidades de objetos, de granularidade fina, de maneira eficiente”

### Solução 11.1

- remover vírgulas que delimitam “de granularidade fina”
- trocar “de maneira eficiente” por “eficientemente”



## 12 – Interpreter

### Redação Original e Final

- Define uma representação e um interpretador para a gramática de uma linguagem.

### Observação

- o *parser* rotula erroneamente “uma representação” como sujeito, pois obviamente não seria possível um sujeito composto (pela conjunção entre dois termos da oração) com o objeto
  - neste caso, não faz sentido alterar a redação; a solução depende de uma correção no *parser*

A seguir são caracterizadas as situações de alterações necessárias para os padrões de projeto analisados (por enquanto de 1 a 12). Os padrões de projeto, para os quais nenhuma alteração de redação foi necessária, foram omitidos.

### **1 - Abstract Factory**

- utilizar verbo no infinito, em vez de verbo substantivado, em construção iniciada pela preposição “para”
  - Exemplo : utilizar “para criar” em vez de “para criação de”
- delimitar sintagma precedido da preposição “de” entre vírgulas, para evitar sua vinculação incorreta com um verbo que o antecede

### **2 – Adapter**

- utilizar adjetivo no lugar de: sintagma verbal formado por verbo de ligação sucedido de verbo no particípio
  - Exemplo : trocar “classe a ser utilizada” por “classe existente”
  - observe que a alternativa de representar como “interface do cliente” não é adequada, porque elimina a semântica do verbo no particípio, que pode ser significativa; como por exemplo: “interface rejeitada pelo cliente”

### **4 – Builder**

- representar o objeto indireto iniciado com a preposição “de” para a posição entre o verbo e o objeto direto, evitando sua interpretação incorreta como parte de um sintagma

### **5 – Chain of Responsibility**

- utilizar verbo principal no infinito, em vez verbo substantivado, em construções que um verbo auxiliar precede o verbo principal
  - Exemplo : utilizar ““Evita acoplar o”” em vez de “Evita o acoplamento do”
- separa da oração principal a primeira oração iniciada por verbo no gerúndio através do uso de dois pontos
- separa uma próxima oração iniciada por verbo no gerúndio, da oração anterior também iniciada por verbo no gerúndio, por vírgula ou por conjunção coordenada “e”

## 6 – Command

- separa através do uso dois pontos, verbo no gerúndio de orações que o sucedem atuando como objeto direto (orações subordinativas objetivas diretas)
- utilizar verbo no infinitivo para iniciar um oração subordinada, em vez do verbo substantivado correspondente
  - Exemplo : utilizar “enfileirar as” em vez de “enfileiramento das”

## 7 – Composite

- substituir expressões do tipo “todo-partes” por dependente de sintagma
  - Exemplo : utilizar “hierarquias de agregação” em vez de “hierarquias todo-partes”
- separa, através do uso de dois pontos, a oração principal da oração subordinada final (iniciada pela preposição “para”)
- utilizar verbo no infinitivo para iniciar um oração subordinada, em vez do verbo no gerúndio
- utilizar advérbio como uma única palavra com sufixo “mente” em vez de adjunto adverbial iniciado pela preposição “de”, para evitar que o adjunto adverbial seja erroneamente interpretado com parte de um sintagma
  - utilizar “uniformemente” em vez de “de maneira uniforme”

## 9 – Facade

- representar o adjunto adverbial iniciado com a preposição “de” para a posição entre o verbo e o objeto direto, evitando sua interpretação incorreta como parte de um sintagma
- utilizar sintagmas encadeados pela preposição “de” em vez de encadeados pela preposição “em”
  - utilizar “interfaces de um sistema” por “interfaces em um sistema”

## 10 – Factory Method

- separar, por vírgula, a oração principal da oração subordinada iniciada por gerúndio
- trocar “qual classe a ser instanciada” por “a classe para instanciar”
- eliminar o uso do pronome relativo “qual”, substituindo a construção iniciada pelo referido pronome e sucedida por substantivo, verbo de ligação e verbo no particípio, pela construção iniciada por substantivo sucedida de verbo no infinitivo preposicionado
  - Exemplo : utilizar “a classe para instanciar” em vez de “qual classe a ser instanciada”

## 11 – Flyweight

- remover vírgulas que delimitam parte de um sintagma iniciada pela preposição “de”
- utilizar advérbio como uma única palavra com sufixo “mente” em vez de adjunto adverbial iniciado pela preposição “de”, para evitar que o adjunto adverbial seja erroneamente interpretado com parte de um sintagma
  - utilizar “eficientemente” em vez de “de maneira eficiente”

## 7.5 Anexo E – Notação Léxica utilizada pelo Parser Anexo A – AS Base para gerar FGRs dos Propósitos.

Forma	Descrição	Tradução	Resumo descritivo
N	Noun	Substantivo	Palavra que designa os seres em geral; nome de estados ou qualidades; ações e sentimentos
Prop	Proper Noun	Substantivo Próprio	Tipo especial de Substantivo que designa entidades em geral pessoas.
V	Verb	Verbo	Palavra que expressa estado, ação mudança de estado, fenómeno da natureza.
adj	Adjective	Adjetivo	Palavra que qualifica os seres: qualidade, aparência ou aspecto, apresentação estado de ser
adv	Adverb	Adverbio	Palavra que se emprega como: modificador do adjetivo ou do próprio adverbio; determinante do verbo
art	Article	Artigo	Determina o Indetermina os seres ao qual está ligado. Substantivação: Palavra Ligada ao artigo torna-se um substantivo.
pron	Pronoun	Pronome	Palavra que se usa em lugar do nome ou a ele se refere ou, ainda, que acompanhe o nome qualificando-o de alguma forma.
prp	Preposition	Preposição	Palavra que estabelece uma relação entre dois ou mais termos da oração
conj	Conjunction	Conjunção	Palavra que estabelece uma relação: - Termos semelhantes da oração ou entre oração da mesma função gramatical (Conj. Coordenação) - Entre duas orações. Relação de subordinação (cada um dos elementos ligados depende do outro)
num	Numeral	Numeral	Palavra que atribui quantidade aos seres ou os situa em determinada sequência - termos numéricos
intj	Interjection	Interjeição	Palavra que expressa emoções, sentimentos ou pensamentos súbitos. Ex: Ah!, Hum!
cl	clause	Oração	Segmento linguístico caracterizado basicamente - Pela presença obrigatória do verbo (ou locução adverbial) - Pela propriedade de se tomar, ela mesma um objeto de análise sintática.
		Oração Finita (fd)	Oração Finita: Não acredita que seja verdade.
		Oração Infinita (id)	Oração Infinita: Consertar um relógio não pode ser fácil.
		Oração Adverbial (ad)	Oração Adverbial: Ajudou onde foi possível
par	Paratagma / Compound unit	Paratagma	Quando temos dois ou mais elementos co-ordenados, eles são interpretados como uma forma distinta e são denominados Paratagmas. Ex. Cobaias, coelhos e ratos, são mamíferos como nós.
g	group	Sintagma	Segmento linguístico que expressa uma relação de dependência. Nessa relação de dependência, diz-se que existe um elemento determinado e outro determinante (ou subordinado), estabelecendo um elo de subordinação entre ambos. Cada um desses elementos constitui um sintagma. Dessa forma, o sintagma se caracteriza conforme o tipo gramatical dos seus elementos nucleares. Não são orações, possuem um Núcleo (Head) e um ou mais dependentes (D)
		Sintagma nominal	quando o núcleo do sintagma é um nome
		Sintagma Adjetival	quando o núcleo do sintagma é um adjetivo
		Sintagma Verbal	quando o núcleo do sintagma é um verbo
		Sintagma Preposicional	quando o núcleo do sintagma é uma preposição
vp pp		Sintagma Adverbial	quando o núcleo do sintagma é um adverbio



## 7.6 Anexo F - Notação Sintática utilizada pelo Parser.

Função	Descrição	Tradução	Resumo Descritivo
S	Subject	Sujeito	Aquele que estabelece concordância com o núcleo do predicado. Quando se trata de predicado verbal, o núcleo é sempre um verbo; sendo um predicado nominal, o núcleo é sempre um nome.
P	Predicator/Verbal	Predicado	O predicado é sintaticamente o segmento lingüístico que estabelece concordância com outro termo essencial da oração – o sujeito -, sendo este o termo determinante (ou subordinado) e o predicado o termo determinado (ou principal).
Od	Direct Object	Objeto Direto (Compl. Verbal)	Termo que completa o sentido de um verbo transitivo direto, por isso, é complemento verbal, na grande maioria dos casos, não preposicionado. Do ponto de vista da semântica, o objeto direto é: - o resultado da ação verbal, ou - o ser ao qual se dirige a ação verbal, ou - o conteúdo da ação verbal. Pode ser formado por um substantivo, pronome substantivo, ou mesmo qualquer palavra substantivada. Além disso, o objeto direto pode ser constituído por uma oração inteira que complemente o verbo transitivo direto da oração dita principal. (oração subordinada substantiva objetiva direta.)
Oi	Indirect Object	Objeto Indireto (Compl. Verbal)	Sintaxe: termo que completa o sentido de um verbo transitivo indireto e vem sempre acompanhado de preposição. Semântica: ser ao qual se destina a ação verbal. Pode ser formado por substantivo, ou pronome substantivo, ou numeral, ou ainda, uma oração substantiva objetiva indireta. Tem como traço mais importante e característico do objeto indireto é a presença da preposição.
Op	Prepositional Object	Complemento Nominal	Termo que complementa o sentido de um nome ou um advérbio, conferindo-lhe uma significação completa ou, ao menos, mais específica. São duas as principais características do complemento nominal: - sempre seguem um nome, em geral abstrato; - ligam-se ao nome por meio de preposição, sempre obrigatória. Os complementos nominais podem ser formados por substantivo, pronome, numeral ou oração subordinada completiva nominal.
A	Adverbial	Complemento Adverbial	Os complementos adverbiais exprimem circunstâncias (como fazem os adjuntos adverbiais), mas completam verbos de significação transitiva (como fazem os objetos). <b>Poucos autores trabalham com o complemento adverbial nas gramáticas escolares</b>

Função	Descrição	Tradução	Resumo Descritivo
Cs	Subject Complement/ Subject Predicative	Predicativo do Sujeito	Apresenta duas características básicas: - acompanha o verbo de ligação; - pertence ao predicado nominal. A formação do predicativo do sujeito pode ser feita através de um substantivo, ou adjetivo, ou pronome, ou numeral, ou ainda uma oração substantiva predicativa.
Co	Object Complement / Object Predicative	Predicativo do Objeto	Apresenta duas características básicas: - acompanha o verbo de ligação implícito; - pertence ao predicado verbo-nominal. A formação do predicativo do objeto é feita através de um substantivo ou um adjetivo.
fC/PRED	Adjunct Complement / Adjunct Predicative	Adjunto Adnominal	Palavra ou expressão que acompanha um ou mais nomes conferindo-lhe um atributo. Termo de valor adjetivo que modificará o nome a que se refere. Não interferem na compreensão do enunciado. (termos acessórios da oração.) Podem ser formados por: artigo, adjetivo, locução adjetiva, pronome adjetivo, numeral e oração adjetiva Abreviaturas no VISL: fC ou PRED, - fCs - referência ao sujeito - Fco - referência ao objeto
fA/ADVL	Adjunct Adverbial	Adjunto Adverbial	É a palavra ou expressão que acompanha um verbo, um adjetivo ou um advérbio modificando a natureza das informações que esses elementos transmitem. Formas de modificação que os adjuntos adverbiais conferem aos elementos: - Modificação circunstancial - Modificação de intensidade. Siglas: fA e ADVL, sendo fAs referente ao sujeito, e fAo referente ao objeto.
SUB/#KS	Subordinator	Oração Subordinadas / Conjunção Subordinativa	Existem pelo menos uma oração principal e uma subordinada, ligadas através de uma conjunção ou um pronome relativo. A oração principal é sempre incompleta, ou seja, alguma função sintática está faltando. As orações subordinadas desempenham a função sintática que falta na principal: objeto direto, indireto, sujeito, predicativo, complemento nominal...
CO	Coordinator	Coordenador / Oração Coordenada	Orações se ligam pelo sentido, mas não existe dependência sintática entre elas. As orações são ligadas por conjunções coordenadas (e, como, mas, ora, logo, porque), ou por simples pontuações (",", ";"). Veja aqui os tipos de orações coordenadas. Siglas: CO - Coordenador: Fugiram do zôo um hipopótamo e um crocodilo. CJT – (Elemento) Conjunto : Fugiram do zôo <b>um hipopótamo e um crocodilo</b> .
CJT	Conjunct	Conjunto / Oração Coordenada	
D	Dependent	Dependente	
H	Head	Núcleo	
UTT	UTTERANCE	Enunciado	
STA	Statement	Oração / Enunciado	
QUE	Question	Interrogação / Enunciado	
COM	Command	Enunciado	
EXC	Exclamation	Exclamação / Enunciado	