

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO
(Bacharelado)

**COMPARATIVO DE ESQUEMAS DE CLASSIFICAÇÃO DE
COMPONENTES REUSÁVEIS**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS DE
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA
COMPUTAÇÃO - BACHARELADO

KÁTIA SIMONE HAMANN

BLUMENAU, JUNHO/1999.

1999/1-32

COMPARATIVO DE ESQUEMAS DE CLASSIFICAÇÃO DE COMPONENTES REUSÁVEIS

KÁTIA SIMONE HAMANN

ESTE TRABALHO DE CONCLUSÃO DE CURSO FOI JULGADO ADEQUADO PARA OBTENÇÃO DOS CRÉDITOS DA DISCIPLINA DE TRABALHO DE CONCLUSÃO DE CURSO OBRIGATÓRIO PARA OBTENÇÃO DO TÍTULO DE:

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Prof. Everaldo Artur Grahl - ORIENTADOR

Prof. José Roque Voltolini da Silva - COORDENADOR DO TCC

BANCA EXAMINADORA

Prof. Everaldo Artur Grahl

Prof. Marcel Hugo

Prof. Roberto Heinzle

DEDICATÓRIA

Dedico este trabalho a minha família e ao meu namorado, por tudo que fizeram por mim.

AGRADECIMENTOS

Agradeço aos professores de Bacharelado em Ciências da Computação da Universidade Regional de Blumenau, por todo conhecimento que adquiri ao longo do curso. Em especial ao professor Everaldo Artur Grahl pela dedicação, orientação e incentivo na condução deste trabalho.

Aos meus pais, pelo apoio e confiança que depositaram em meu desempenho nos estudos.

Ao meu namorado e amigos, pelo apoio nos momentos, em que, devido aos compromissos acadêmicos não pude estar presente.

SUMÁRIO

LISTA DE FIGURAS.....	vii
LISTA DE QUADROS	vii
RESUMO.....	viii
ABSTRACT	ix
1. INTRODUÇÃO	1
1.1 ORIGEM.....	1
1.2 OBJETIVOS	2
1.3 ORGANIZAÇÃO DO TEXTO.....	3
2. REUTILIZAÇÃO.....	4
2.1 CONCEITOS INICIAIS	4
2.2 BENEFÍCIOS DO REUSO.....	5
2.3 OBSTÁCULOS DO REUSO.....	6
2.4 BIBLIOTECA DE COMPONENTES REUSÁVEIS	8
2.4.1 SELECIONAR OS COMPONENTES A SEREM REUSADOS	9
2.4.2 ORGANIZAR A ESTRUTURA DA BIBLIOTECA	9
2.4.3 ESCOLHER UM ESQUEMA DE CLASSIFICAÇÃO	11
3 CLASSIFICAÇÃO DE COMPONENTES	12
3.1 PRINCÍPIOS DE CLASSIFICAÇÃO	12
3.2 OBJETIVO DOS ESQUEMAS DE CLASSIFICAÇÃO	13
3.3 ESQUEMAS DE CLASSIFICAÇÃO	14
3.3.1 MÉTODOS PARA REPRESENTAR OS COMPONENTES.....	15
3.3.2 SISTEMAS INDEXADOS COM VOCABULÁRIO NÃO CONTROLADO	17
3.3.2.1 ESQUEMA DE CLASSIFICAÇÃO EM TEXTO LIVRE	17
3.3.2.2 ESQUEMA DE CLASSIFICAÇÃO PALAVRA-CHAVE	18
3.3.2.3 ESQUEMA DE CLASSIFICAÇÃO VALOR-ATRIBUTO	18
3.3.3 SISTEMAS INDEXADOS CONTROLADOS	19
3.3.3.1 ESQUEMA DE CLASSIFICAÇÃO ENUMERADA	20
3.3.3.2 ESQUEMA DE CLASSIFICAÇÃO FACETA	21
3.3.3.3 ESQUEMA DE CLASSIFICAÇÃO FACETA COM A APLICAÇÃO DE UM THESAURUS	23

4	COMPARATIVO ENTRE OS ESQUEMAS DE CLASSIFICAÇÃO	24
4.1	CRITÉRIOS DE AVALIAÇÃO	24
4.2	COMENTÁRIOS	25
4.2.1	FLEXIBILIDADE	26
4.2.2	AMBIGÜIDADE	26
4.2.3	EFETIVIDADE	27
4.2.4	AMIGABILIDADE	27
4.2.5	FACILIDADE DE COMPREENSÃO	27
4.2.6	ESFORÇO NECESSÁRIO PARA A CONSTRUÇÃO DA BIBLIOTECA	28
5	ESPECIFICAÇÃO E IMPLEMENTAÇÃO DO PROTÓTIPO	29
5.1	DIAGRAMA DE CONTEXTO	29
5.2	DFD NÍVEL 0	30
5.3	MODELO ENTIDADE-RELACIONAMENTO	32
5.4	DICIONÁRIO DE DADOS	33
5.5	APRESENTAÇÃO DO SISTEMA	34
5.5.1	CADASTRO	35
5.5.2	CLASSIFICAÇÃO DE COMPONENTES	38
5.5.3	CONSULTA E RECUPERAÇÃO DE COMPONENTES	42
5.5.4	RELATÓRIOS	46
6	CONCLUSÃO	48
6.1	CONSIDERAÇÕES FINAIS	48
6.2	SUGESTÕES	49
	REFERÊNCIAS BIBLIOGRÁFICAS	51

LISTA DE FIGURAS

1	AMBIENTE DE BIBLIOTECA DE REUSO	14
2	TAXONOMIA DE MÉTODOS INDEXADOS PARA BIBLIOTECA.....	16
3	DIAGRAMA DE CONTEXTO	29
4	DFD NÍVEL 0.....	30
5	CONTINUAÇÃO DFD NÍVEL 0	31
6	CONTINUAÇÃO DFD NÍVEL 0	32
7	MODELO ENTIDADE-RELACIONAMENTO	32
8	MENU PRINCIPAL DO SISTEMA DE CLASSIFICAÇÃO DE COMPONENTES	34
9	MENU ESCOLHA DE ESQUEMA DE CLASSIFICAÇÃO.....	35
10	TELA CADASTRO DE VALORES VÁLIDOS PARA FACETAS	36
11	TELA CADASTRO DE ATRIBUTOS ESQUEMA DE CLASSIFICAÇÃO VALOR-ATRIBUTO	37
12	TELA CADASTRO DO ESQUEMA DE CLASSIFICAÇÃO FACETA COM UM THESAURUS.....	38
13	TELA PARA ESCOLHA DO COMPONENTE A SER CLASSIFICADO E ESCOLHA DO ESQUEMA DE CLASSIFICAÇÃO	39
14	TELA DE CLASSIFICAÇÃO DE COMPONENTES NO ESQUEMA VALOR-ATRIBUTO.....	40
15	TELA DE CLASSIFICAÇÃO DE COMPONENTES NO ESQUEMA PALAVRA-CHAVE.....	41
16	TELA DE CLASSIFICAÇÃO DE COMPONENTES NO ESQUEMA FACETA.....	42
17	TELA DE RECUPERAÇÃO DE COMPONENTES NO ESQUEMA VALOR-ATRIBUTO	43
18	TELA DE RECUPERAÇÃO DE COMPONENTES NO ESQUEMA PALAVRA-CHAVE.....	44
19	TELA DE RECUPERAÇÃO DE COMPONENTES ATRAVÉS DO ESQUEMA FACETA	45
20	TELA MENU RELATÓRIOS.....	46
21	TELA RELATÓRIO DE COMPONENTES CLASSIFICADOS POR FACETA	47

LISTA DE QUADROS

1	COMPARATIVO ENTRE ESQUEMAS DE CLASSIFICAÇÃO.....	25
2	DICIONÁRIO DE DADOS DO SISTEMA DE CLASSIFICAÇÃO DE CCOMPONENTES	35

RESUMO

A prática do reuso de software é um conceito bastante discutido no meio computacional e apesar de se tratar de um conceito antigo, ainda não está sendo praticado por muitas empresas desenvolvedoras de software. Há vários fatores que estão impedindo a inclusão deste conceito nas organizações, como o custo elevado para introduzi-lo na organização e a necessidade da alteração da cultura da organização no que se refere ao desenvolvimento de softwares. Vários estudos têm sido feitos para melhorar o entendimento e facilitar o uso desta prática dentro das organizações. Um meio para facilitar a prática do reuso é através da implantação de uma biblioteca que irá classificar e armazenar as partes do software que são possíveis de serem reusadas, também chamadas de componentes. Alguns esquemas de classificação são propostos na literatura, porém nenhum deles é completamente eficiente. Cada esquema de classificação possui características próprias para fazer a classificação dos componentes. Este trabalho apresenta um estudo voltado a esquemas de classificação de componentes reusáveis, através da implementação de uma biblioteca com fins didáticos no ambiente NetExpress 2.0. Os esquemas de classificação utilizados neste trabalho são o esquema de classificação faceta, palavra-chave e valor-atributo para a classificação de componentes. Para a recuperação de componentes, além dos esquemas mencionados foi utilizada a junção do esquema de classificação faceta com um *thesaurus* para melhorar a performance de recuperação.

ABSTRACT

Software reuse practice is a very discussed concept by software engineering area but not yet it is being practice by many of the software houses companies. An important concept of software reuse is the implementation of a reusable library with reusable components that should be include in the library with a classification sheme and can be also search through a classification sheme. Each classification sheme has its own characteristics to make a classification on reusable components. This work is a research of software classification shemes and through development of a reusable components library by NetExpress 2.0 environment, with an academic purpose, we can differentiate the characteristics of each classification sheme. The classification shemes used to make the classification are faceted, key-word and atribute-value. To make the search of recusable componentes was used the same shemes and faceted with a thesaurus to improve the performance by searching.

1 INTRODUÇÃO

1.1 ORIGEM

O reuso de software segundo [HAL94] é um método proposto para melhorar a qualidade e a produtividade no desenvolvimento de software. Segundo [KUT97] e [RAD96], a prática do reuso é considerada um dos caminhos mais eficazes para melhorar tanto o processo de desenvolvimento do software quanto a sua qualidade e consistência.

Com o advento da crise do software, diz [FRE87] que o reuso do software começou a ser considerado como uma meta que poderia ser alcançada de forma mais ampla através de pesquisas e avanços no desenvolvimento do software.

Desde então, segundo [ARN97], o reuso tem sido usado em muitos desenvolvimentos de software, porém o método nunca adquiriu importância na indústria do desenvolvimento de software. A partir de 1978, muitos processos baseados na reusabilidade foram testados na prática. Conforme [FRE87], planos estratégicos incorporaram o reuso e muitas pesquisas e desenvolvimentos iniciaram a prática do reuso.

No início da década de 80, segundo [FRE87], muitas organizações dos Estados Unidos começaram a focalizar a reusabilidade como uma parte importante do desenvolvimento de seus softwares, procurando diminuir os esforços despendidos no desenvolvimento de novos softwares. Igualmente os militares iniciaram esforços para introduzir a reusabilidade em seus processos de desenvolvimento naquela época.

Porém, ainda nos dias atuais, o reuso não é muito praticado nas empresas desenvolvedoras de software. Isto deve-se ao fato de que a idéia básica do reuso é simples e óbvia, segundo [BIA95], porém na prática, quando o reuso é implantado em uma organização, alguns problemas podem surgir, tornando-se difícil a sua implantação. Existe uma certa dificuldade na implantação da prática do reuso dentro de uma organização, pois para que o reuso possa se tornar realidade, há necessidade de se alterar a cultura da empresa, no que diz

respeito ao processo de desenvolvimento do software e às ferramentas de suporte utilizadas no processo de desenvolvimento orientado para o reuso de novos sistemas.

A reutilização de software trata do reaproveitamento não somente das linhas de código, mas de várias partes de um sistema, chamadas de componentes. Conforme [MCC93], podem ser considerados como componentes os códigos de programa, a especificação do projeto, a documentação, ou seja, qualquer informação que é utilizada para criar um software.

No trabalho de [HAL94], é citado que os componentes por si só não proporcionam o benefício desejado, é preciso que estes componentes sejam selecionados, pois devem ser os mais genéricos possíveis para que se tornem eficientes em outros sistemas de software.

Após selecionados os componentes, estes devem ser armazenados de tal forma que sejam fáceis de recuperar e compreender. Para isto, é necessário o uso de uma biblioteca com o objetivo de classificar todos os componentes candidatos ao reuso, objetivando a fácil localização, fácil compreensão, modificação e combinação dos mesmos [MCC93]. Segundo [HAL94], na biblioteca cada componente é identificado e classificado de acordo com esquemas, sendo que cada esquema possui características próprias para identificar e classificar um componente reusável. Um bom esquema de classificação não somente melhora a base de catalogação do software mas também melhora o meio de achar um determinado componente do software mantido na biblioteca.

Este trabalho trata-se de uma sugestão apresentada no trabalho de conclusão de curso de Débora Ramos [RAM98], visto que na literatura não é feita uma comparação entre os esquemas com o intuito de definir qual o esquema de classificação mais apropriado para ser aplicado na classificação de componentes reusáveis.

1.2 OBJETIVOS

O objetivo do trabalho é comparar esquemas de classificação de componentes reusáveis e implementar uma biblioteca com alguns esquemas estudados utilizando o ambiente NetExpress 2.0. Esta biblioteca tem como principal função ser um instrumento didático sobre os conceitos estudados.

1.3 ORGANIZAÇÃO DO TEXTO

A seguir serão descritos brevemente cada capítulo do trabalho.

Este capítulo de introdução apresenta uma visão geral da origem deste trabalho, sua relevância e objetivos.

No segundo capítulo serão abordados conceitos referente a reutilização de componentes de software, os benefícios que o reuso pode prover e os obstáculos encontrados para a implantação deste conceito em uma organização e alguns pré-requisitos para a prática do reuso em uma organização.

No terceiro capítulo será abordada a classificação de componentes, os princípios de classificação e objetivo dos esquemas de classificação. Serão apresentados métodos de representação de componentes e esquemas para a classificação dos mesmos. Alguns dos esquemas abordados são o enumerado, palavra-chave, valor-atributo, *thesaurus* e faceta.

No quarto capítulo é feito um comparativo, de acordo com alguns critérios estabelecidos entre os esquemas de classificação mais conhecidos que são o esquema faceta, palavra-chave, valor-atributo e o esquema faceta com a utilização de um *thesaurus*.

O quinto capítulo trata da especificação e implementação de um sistema de classificação de componentes reusáveis, utilizando alguns esquemas estudados.

O sexto capítulo trata das conclusões e sugestões deste trabalho.

2 REUTILIZAÇÃO

A reutilização de componentes é uma forma de melhorar aspectos relacionados ao desenvolvimento de um software. Neste capítulo serão abordados alguns conceitos, objetivos e obstáculos da reusabilidade. Também serão abordadas as principais etapas para a implementação de uma biblioteca de componentes reutilizáveis.

2.1 CONCEITOS INICIAIS

A constante preocupação com os processos relativos ao desenvolvimento de um software, busca pelo aumento da produtividade, qualidade, confiabilidade e a redução dos custos com o desenvolvimento dos mesmos, têm sido uma constante nas empresas desenvolvedoras de software.

Uma forma para melhorar estes aspectos, segundo [KUT97], é a introdução da reusabilidade no processo de desenvolvimento do software. Segundo autores como [HAL94], [KUT97] e [RAD96], reuso é uma prática que, se aplicada ao desenvolvimento do software, melhora significativamente a produtividade no processo de desenvolvimento do software, melhora também a qualidade dos sistemas e conseqüentemente, facilita a manutenção e reduz os custos no desenvolvimento de um software.

A reusabilidade consiste no reaproveitamento de partes de sistemas, definidos como componentes, que anteriormente foram desenvolvidos e testados, proporcionando desta forma, a redução do custo e do tempo gasto para desenvolver novos sistemas.

Segundo [RAD96], estudos têm mostrado que a reusabilidade pode prover significativo aumento na produtividade e qualidade no desenvolvimento de sistemas. A produtividade aumenta quando partes do sistema são desenvolvidas para serem reutilizadas futuramente no desenvolvimento de novos projetos e a qualidade é adquirida com o freqüente uso destas partes anteriormente testadas em uma variedade de circunstâncias, evitando assim erros na implementação.

Segundo [LUN95], o problema atual não é a necessidade do reuso, mas a necessidade de sistemáticas bem definidas de reuso. Diferentes abordagens para o reuso de software tem sido relatadas na literatura. Entre as abordagens, a análise de domínio é defendida como um passo para o reuso efetivo. A análise de domínio identifica semelhanças entre aplicações em um determinado domínio específico.

Há duas principais razões para a utilização da análise de domínio:

- a) ela enfatiza a reusabilidade da análise e do projeto, em vez de somente códigos de programa;
- b) ela é um estudo da informação considerada relevante em uma área do problema, e não somente em um sistema em particular.

Como resultado do estudo, informações comuns e reusáveis dentro de um domínio podem ser reusados para construir novas aplicações que também tem o mesmo domínio do problema. Deste modo, a análise do domínio pode suportar reuso para grande proporção e ter o potencial para receber alto lucro. Apesar da análise de domínio ter um potencial para proporcionar larga-escala de reuso de software, ela sofre várias barreiras próprias para a complexidade e o esforço requerido. A análise de domínio é geralmente apropriada para domínios bem-compreendidos, mas é difícil para ser iniciado por problemas pouco-compreendidos ou para novas áreas de aplicação.

Para suportar o reuso de software sobre diferentes domínios, o resultado da análise de domínio precisa ser classificado, especialmente os componentes de software de um modelo de domínio. Um modelo de domínio é uma representação de um domínio que representa objetos e relacionamentos, funções e procedimentos. O modelo identifica requisitos genéricos ou semelhantes, e diferenças dos problemas em cada domínio.

2.2 BENEFÍCIOS DO REUSO

Conforme [MCC97], a implantação do reuso no processo de desenvolvimento do software proporciona os seguintes benefícios:

- a) o aumento da produtividade do software;
- b) a redução do tempo de desenvolvimento;

- c) a redução de pessoal destinado a desenvolver o software;
- d) ferramentas e métodos mais fáceis de projeto em projeto;
- e) a redução do custo do desenvolvimento de um software;
- f) melhorias na interoperabilidade do sistema;
- g) uma vantagem competitiva no mercado;
- h) o compartilhamento de especificações de sistemas, projetos, código e outros documentos de projetos produzidos por outras equipes.

Além disto, segundo [MCC94], o reuso de software é visto como um importante meio de prover softwares com qualidade, reduzindo erros, pois os componentes confiáveis de serem reusados são usados para criar novos sistemas de software.

A construção de um software feito através de componentes reusáveis, segundo [MCC97], são mais fáceis e mais baratos de serem mantidos pelo fato dos componentes reusáveis serem mais confiáveis, visto que foram anteriormente testados.

2.3 OBSTÁCULOS DO REUSO

Conforme [MCC97] e [BIA95], a idéia básica do reuso é bastante simples, porém a sua implantação não é. Ainda nos dias atuais, a maioria das organizações dos Estados Unidos utilizam o reuso informalmente. Apesar do reuso ser um conceito bastante antigo no desenvolvimento de softwares, ainda é uma tecnologia que tem permanecido com o conceito de que é a coisa certa de se fazer, mas nunca realmente foi colocado em prática.

Segundo [RAD96], o reuso de software é difícil de ser inicializado em uma organização, porque ele exige um alto investimento inicial, que é recuperado somente após longo prazo. Neste custo estão incluídos a escolha dos componentes reusáveis que serão necessários para inclusão na biblioteca, o treinamento de pessoal para montar a biblioteca, a inclusão de componentes na biblioteca e treinamento da equipe de desenvolvimento para utilizar a biblioteca de software e reusar os atuais componentes.

No trabalho de [MCC97] cita-se que a maioria dos desenvolvedores de software e gerentes não dão a importância devida para a incorporação da reusabilidade em suas estratégias de desenvolvimento de softwares. Uma das razões é que o reuso do software

requer planejamento para a reusabilidade quando um componente de software é originalmente definido e implementado.

Outro problema é que a maioria das metodologias de desenvolvimento de software, como as linguagens de programação, não incluem o reuso. As metodologias não definem explicitamente o que, quando e como praticar o reuso de partes de um processo de desenvolvimento de software.

Outro obstáculo para a implantação do reuso é a dificuldade em se determinar exatamente o que é um componente reusável e como representá-lo de forma que possa ser fácil de ser classificado, descrito, reconhecido e reusado. No trabalho de [KUT97] cita-se também o elevado custo para se desenvolver componentes reusáveis como um dos fatores que dificultam a aceitação da implantação da reusabilidade em uma organização.

[RAD96] cita ainda outro obstáculo que é a tendência natural dos produtores de software, em proteger sua propriedade intelectual contra os concorrentes, para manter uma vantagem comercial. A colaboração destas entidades comerciais em estender os benefícios do reuso sobre os limites das organizações requerem a existência de uma organização benevolente para coordenar e consolidar as atividades de reuso.

Para uma visão mais abrangente de fatores que dificultam a implantação da reusabilidade nos processos de desenvolvimento de software, [MCC97] descreve uma lista dos maiores obstáculos para a prática do reuso do software:

- a) há muitas dúvidas sobre que tipo de componente reusar;
- b) dificuldade em se descobrir componentes que são comuns a outros sistemas;
- c) necessidade de se padronizar os programas;
- d) dificuldade em decidir o que entra na biblioteca de componentes reusáveis;
- e) entender os efeitos que podem causar uma mudança em determinado componente;
- f) descrever e classificar componentes de software;
- g) os benefícios são alcançados somente a longo prazo;
- h) falta de gerenciamento de suporte para a reusabilidade;
- i) falta de entendimento sobre o porque praticar o reuso;
- j) acreditar que reuso é contra a criatividade;
- k) nenhuma metodologia de suporte para reuso;

- l) cultura da corporação e sistemas de recompensa desencorajam o reuso;
- m) necessidade de treinamento para a prática do reuso;
- n) nada para reusar, nenhuma biblioteca de software;
- o) visão de reuso com alto risco tecnológico;
- p) custo, pois um componente de software reusável pode custar 25 % mais para ser desenvolvido que um componente não projetado para a reusabilidade.

Conforme [MCC97], o aspecto chave para o reuso é um processo de desenvolvimento de software dirigido para o reuso, que cultiva o pensamento em reuso e faz do reuso uma parte integral do processo. Um processo dirigido para o reuso descreve como sistemas de software são construídos através da construção de componentes reusáveis e como construir e gerenciar componentes reusáveis.

Somente quando a condição do reuso é integrada no processo de software é que o reuso pode vir a ser um meio natural e normal de trabalho. Conforme [MCC97], para isto é necessário um processo dirigido ao reuso, que guie o desenvolvimento e a manutenção de sistemas através do reuso de componentes pré-definidos, e conduz a identificação e construção de componentes reusáveis.

2.4 BIBLIOTECA DE COMPONENTES REUSÁVEIS

Um pré-requisito para a reusabilidade é ter algo para reusar, porém somente dispor de componentes para reutilizar, segundo [MCC97], não é o suficiente. É necessário que os componentes sejam facilmente encontrados e entendidos para que se possa esperar que os desenvolvedores reutilizem estes componentes mais do que construam os seus próprios.

A biblioteca reusável é um elemento importante para formalizar a prática do reuso em uma organização. Uma biblioteca reusável é essencial para prover a manutenção e gerenciamento de componentes reusáveis, visto que o conteúdo da biblioteca reusável provavelmente será modificado e aumentado com o passar do tempo.

Uma biblioteca de componentes reusável tem a função de tornar os componentes a serem reusados mais fáceis e rápidos de serem encontrados e mais fáceis de serem

compreendidos do que se os mesmos fossem construídos novamente. Uma biblioteca de reuso, organiza, armazena e gerencia componentes reutilizáveis.

Segundo [MCC97], há três principais etapas para a implementação de uma biblioteca de componentes reusáveis, que são:

- a) selecionar os componentes a serem reusados;
- b) organizar a estrutura da biblioteca;
- c) escolher um método de classificação.

Estas etapas serão abordadas nas próximas sessões.

2.4.1 SELECIONAR OS COMPONENTES A SEREM REUSADOS

Os componentes a serem reusados são aqueles que tem maiores chances de serem utilizados por novos projetos importantes para a organização, ou seja, deve-se escolher os componentes que tenham a maior possibilidade de retorno do investimento feito no componente para torná-lo reusável, e que sejam mais requeridos pelos desenvolvedores.

Os tipos de componentes que a empresa deseja reusar devem ser definidos antes mesmo de se iniciar um programa de reutilização, pois estes componentes devem ser conhecidos para definir os requisitos da biblioteca de softwares reusáveis, bem como para acomodar estes componentes no processo de desenvolvimento.

O segredo para se obter o máximo dos benefícios trazidos pela reutilização, está em iniciar a reutilização dos componentes de mais alto nível, reusando arquiteturas genéricas assim como simples módulos a nível de código.

2.4.2 ORGANIZAR A ESTRUTURA DA BIBLIOTECA

Uma biblioteca reusável pode ser projetada fisicamente como centralizada que pode ser acessada por desenvolvedores de software através de um local qualquer ou distribuindo bibliotecas que suportam organizações de desenvolvimento de software diferentes ou diferentes domínios.

A estrutura da biblioteca de componentes reusáveis pode ser:

- a) centralizada: geralmente utilizada por empresas que seguem padrões, tais como, convenções de nomeação e critérios de certificação. O problema com a biblioteca reusável centralizada é que ela pode vir a ficar muito grande, se conter os componentes reusáveis de toda a organização. Assim, ela se torna mais difícil para gerenciar e para encontrar, a partir dela, um possível componente candidato ao reuso.
- b) distribuída e local: se a biblioteca for organizada como várias bibliotecas locais distribuídas, que individualmente suportam as necessidades de um desenvolvedor, grupo ou domínio em particular, o tamanho da biblioteca irá diminuir. Para um mesmo domínio cerca de vinte componentes, armazenados na biblioteca, são suficientes para dar suporte a novos desenvolvimentos no domínio. Com um menor número de componentes armazenados na biblioteca fica mais fácil para os desenvolvedores encontrar e entender o possível candidato ao reuso, além de simplificar bastante a tarefa de gerenciamento da mesma. O problema com este tipo de biblioteca reside no fato de que poderão existir grande quantidade de componentes duplicados entre as várias bibliotecas, ou poderão existir em uma biblioteca que não estão em outra.
- c) centralizada e distribuída: para resolver os problemas com a biblioteca centralizada e a distribuída, muitas empresas optaram por construir um biblioteca centralizada e distribuída. Neste caso a biblioteca pode ser construída sob dois aspectos:
 - dividir a biblioteca em quatro partes:
 - parte 1: componentes que estão armazenados na biblioteca porém não possuem documentação sobre sua qualidade e reusabilidade;
 - parte 2: componentes que foram utilizados na construção do último sistema, foram testados, apresentam documentação e qualidade padrão, mas os componentes não foram projetados para o reuso, isto é, eles não foram generalizados, padronizados, documentados e certificados para o reuso;
 - parte 3: componentes que foram criados para o reuso mas não foram certificados;

- parte 4: componentes que foram criados ou preparados para o reuso e estão totalmente certificados.

Esta divisão apresenta vantagens no gerenciamento da biblioteca e alguns benefícios importantes, tais como:

- a biblioteca pode ser criada mais rapidamente desde que os componentes não estejam totalmente preparados ou certificados para o reuso;
 - os desenvolvedores sabem exatamente o que eles esperam de um componente, desde que a sua posição na biblioteca indique sua qualidade e certificação de reuso. A medida que os componentes vão aumentando suas características em relação a reusabilidade, eles vão passando para a próxima divisão da biblioteca.
- dividir a biblioteca de acordo com o escopo dos componentes.

2.4.3 ESCOLHER UM ESQUEMA DE CLASSIFICAÇÃO

Os componentes armazenados na biblioteca devem ser organizados, caso contrário, eles até não serão difíceis de encontrar, porém serão difíceis de serem entendidos.

Um esquema de classificação é o método de representação para componentes reusáveis que possibilita ao desenvolvedor encontrar facilmente e rapidamente os componentes, bem como entendê-los. Classificar um componente através de esquemas de classificação é um fator motivador para a prática do reuso, pois estando o componente classificado e armazenado em uma biblioteca, o mesmo pode ser recuperado mais facilmente. Com isto, o desenvolvedor sente-se mais motivado a praticar o reuso a refazer um componente anteriormente já desenvolvido.

A escolha de um ou mais esquemas é feita de acordo com a cultura e necessidades da empresa, pois cada esquema de classificação possui características próprias para classificar e recuperar componentes. Este assunto será abordado de forma mais abrangente no próximo capítulo.

3 CLASSIFICAÇÃO DE COMPONENTES

Neste capítulo serão abordados os esquemas de classificação de componentes, seus objetivos e formas de classificação de componentes.

3.1 PRINCÍPIOS DE CLASSIFICAÇÃO

Classificação é um dos métodos mais fundamentais de várias disciplinas relacionadas à ciência e engenharia. A classificação ajuda comparar e compreender diferentes sistemas ou áreas. Recentemente a classificação também tem sido estudada para software.

Classificar, segundo [FRE87], é agrupar algo semelhante. Todos os membros, ou classes de um grupo produzidos pela classificação dividem no mínimo uma característica em comum que os outros grupos não possuem. A classificação mostra o relacionamento entre coisas e entre classes de coisas. O resultado é uma cadeia ou estrutura de relacionamentos.

Já [FRA94] diz que classificar é o processo de criar uma representação usando métodos de biblioteca tradicionais. Um esquema de classificação para software, segundo [MCC97] é um método de representação de componentes reusáveis que possibilita maior compreensão dos componentes e facilidade e rapidez na procura por componentes reusáveis que estão incluídos na biblioteca. A importância de uma biblioteca para o programa de reuso depende de vários fatores como:

- a) alteração de pessoal: bibliotecas são menos importantes em ambientes com poucas alterações de pessoal, pois a informação do componente será avaliada pelas pessoas que trabalham no ambiente;
- b) bibliotecas podem ser menos importantes em ambientes de reuso baseado em métodos generativos, porque estes ambientes requerem pouca intervenção humana, assim como para pesquisas e entendimento de componentes durante o processo de construção do software;
- c) a importância das técnicas de representação é mínima para pequenas coleções, pois o problema a ser pesquisado é fácil. Muitas organizações estão somente

começando a adquirir o reuso, e deste modo, têm pequenas coleções, portanto poucos componentes.

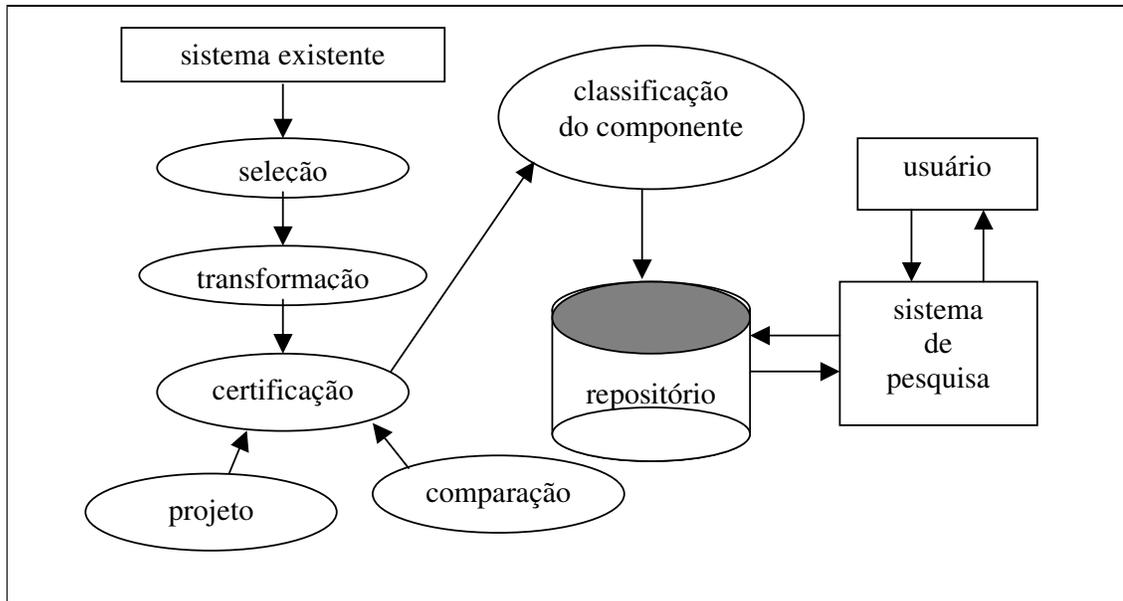
3.2 OBJETIVO DOS ESQUEMAS DE CLASSIFICAÇÃO

Segundo [MCC97], os componentes reusáveis além de armazenados em uma biblioteca devem ser devidamente organizados para facilitar a recuperação e o entendimento dos mesmos. Classificando os componentes em uma estrutura, torna-se mais fácil o processo de recuperação de componentes já armazenados em uma biblioteca. [HAL94] também enfatiza que os componentes reusáveis precisam estar organizados e descritos em uma biblioteca, porém, esta biblioteca precisa estar estruturada para classificar os componentes de software de várias maneiras.

Conforme [PIE86] e [FRE87], uma das maiores dificuldades em reusar partes de um software é localizar e recuperar efetivamente estas partes. Segundo [HAL94], o objetivo de um esquema de classificação para componentes reusáveis é classificar os componentes de tal forma que os mesmos possam ser compreendidos e recuperados mais facilmente.

Pelo fato do crescimento de tamanho das bibliotecas de reuso, da modificação de pessoal, da importância dos métodos de representação de componentes, e do entendimento de domínios, é que métodos de representação são, e continuam sendo, um importante tópico em reuso de software.

Primeiramente, em um ambiente de biblioteca de reuso, são precisos componentes para serem reusados. Estes componentes podem ser adquiridos através de projeto, reengenharia, ou de um sistema de pesquisa, conforme demonstrado na figura 1.



Fonte: [FRA94]

Figura1 - Ambiente de Biblioteca de Reuso

Os componentes após selecionados e aprovados para o reuso em novas aplicações, precisam ser classificados e armazenados no repositório. A classificação é feita através de um esquema de classificação, com o objetivo de facilitar a sua busca e entendimento.

Na recuperação, utiliza-se um sistema de pesquisa, onde o usuário formula a pesquisa para a recuperação do componente desejado e o sistema de pesquisa irá procurar e recuperar os componentes que reúnem os requisitos solicitados pelo usuário para sua aplicação em novos projetos.

Este método de recuperação é bastante eficiente, pois estando os componentes classificados e armazenados, os mesmos podem ser recuperados mais facilmente através de uma simples pergunta ao sistema de pesquisa, contendo as características desejadas do componente a ser recuperado.

3.3 ESQUEMAS DE CLASSIFICAÇÃO

Um esquema de classificação, pode expressar dois modos de relacionamento, que são:

- a) relacionamento hierárquico e
- b) relacionamento sintático.

O relacionamento hierárquico é baseado no princípio de subordinação ou inclusão, onde os componentes são organizados por tipo de aplicação e pela sua função dentro da aplicação. No relacionamento sintático, relata-se dois ou mais conceitos pertencentes a hierarquias diferentes e a representação é feita como classes compostas. Esquemas de classificação típicos são estritamente hierárquicos.

3.3.1 MÉTODOS PARA REPRESENTAR OS COMPONENTES

Segundo [FRA94], métodos para representar os componentes reusáveis e sistemas para suportar estes métodos têm se proliferado há poucos anos. Estes métodos são extraídos de três grandes áreas:

- a) biblioteca e ciência da informação;
- b) inteligência artificial e
- c) hipertextos.

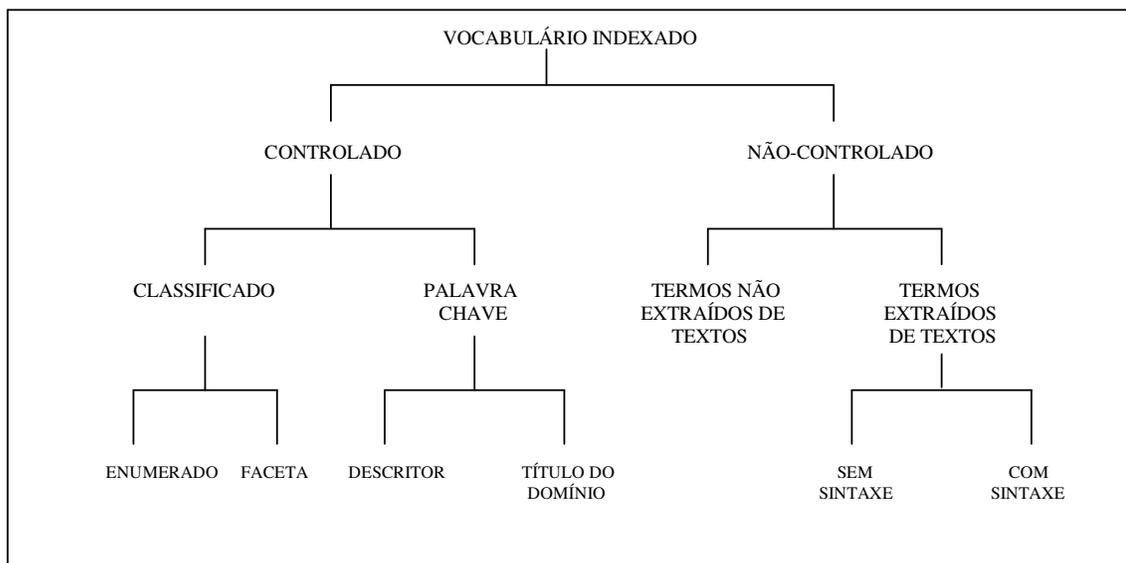
As áreas de inteligência artificial e sistemas de hipertextos têm sido usados somente experimentalmente, sendo que a área biblioteca e ciência da informação está sendo mais utilizada atualmente.

Segundo [HAL94], não há um método padrão de representação e recuperação de componentes reusáveis, mas há dois tipos de abordagem:

- a) método de sistemas de recuperação de informação;
- b) métodos de inteligência artificial.

A abordagem do método de sistemas de recuperação de informação refere-se a sistemas de recuperação de informação, compreendendo várias indexações e técnicas de recuperação desenvolvidas na área de bibliotecas e ciência da informação. A abordagem dos métodos de inteligência artificial tem base em métodos de inteligência artificial, que compreendem técnicas de representação do conhecimento e métodos de compreensão em linguagem natural.

Segundo [LUN95], ambas as abordagens produzem um relacionamento hierárquico. A estrutura de árvore da figura 2 representa uma taxonomia dos métodos indexados para bibliotecas de componentes reusáveis, conforme [FRA94] e [HAL94].



Fonte: [FRA94]

Figura 2 - Taxonomia de Métodos Indexados para Biblioteca.

Sistemas indexados, segundo [HAL94], são muitas vezes classificados de acordo com o tipo de vocabulário ou linguagem de indexação que é aplicada para representações.

Segundo [FRA94] e [HAL94], os métodos de representação baseados na área de biblioteca e ciência da informação são caracterizados por um vocabulário indexado sendo dividido em duas grandes categorias:

- a) vocabulário controlado e
- b) vocabulário não-controlado.

O vocabulário não controlado é aplicado em sistemas de texto livre, na tecnologia que é aplicada no documento texto ou através de palavras-chave do sistema, aonde o assunto do documento é expresso em linguagem natural por um indexador. Este vocabulário não põe restrições nos termos e na sintaxe do vocabulário, que pode ser usado em uma descrição. Já o vocabulário controlado coloca limites nos termos a serem usados para descrever um componente classificado ou na sintaxe que pode ser usada para combinar estes termos.

3.3.2 SISTEMAS INDEXADOS COM VOCABULÁRIO NÃO CONTROLADO

Conforme [MCC97], estes sistemas são baseados em métodos que consideram texto livre para definir um esquema de classificação, onde os componentes são descritos por palavra-chave ou por frases em texto livre. Nestes esquemas não há restrições de vocabulário sobre os termos que podem ser usados para descrever um componente.

Estas palavras chave ou texto livre podem ser extraídas automaticamente da documentação ou diretamente da pessoa que está classificando o componente. Conforme [MCC97], este é o meio mais barato e mais rápido para classificar componentes, porque podem ser automatizadas através do uso de ferramentas de catalogação.

Algumas vantagens no uso de vocabulário não controlado em relação ao uso de um vocabulário controlado, conforme citado por [HAL94]:

- a) baixo custo, visto que os termos utilizados para a classificação são às vezes extraídos diretamente dos textos do componente a ser indexado. A tarefa de indexação pode ser automatizada. Este esquema tem custo relativamente menor se comparado à indexação humana.
- b) fácil especificação, pois os termos não são restritos. Os termos podem ser criados conforme possível. Por exemplo, um pesquisador que deseja encontrar um algoritmo de “lista encadeada”, pode usar o termo principal, que é “lista”.

3.3.2.1 ESQUEMA DE CLASSIFICAÇÃO EM TEXTO LIVRE

Conforme [FRA94], na indexação de texto livre, os termos são automaticamente extraídos de documentações, ou seja, qualquer termo significativo do documento pode ser usado como chave de pesquisa, como por exemplo, o principal descritivo de um código de módulo.

O problema deste esquema de classificação é que componentes que realizam funções diferentes podem ter a mesma descrição ou então, componentes que exercem a mesma função podem apresentar descrições diferentes, gerando ambigüidade e redundância entre os componentes reusáveis. Este esquema também exige um certo grau de conhecimento do sistema por parte da pessoa que está classificando o componente, pois para realizar uma

procura de componente através deste esquema é necessário que se tenha conhecimento dos termos específicos para realizar uma pesquisa de busca dos componentes já incluídos na biblioteca.

Visto que nem sempre a pessoa que classificou o componente é a mesma que irá recuperá-lo, torna-se difícil a recuperação de componentes por desenvolvedores que não possuam este conhecimento específico.

3.3.2.2 ESQUEMA DE CLASSIFICAÇÃO PALAVRA CHAVE

Num esquema de classificação baseado em palavra-chave, as palavras ou termos designados para um componente não são controlados, havendo total liberdade para a escolha do termo ou palavra que irá representar o componente. Isto, segundo [HAL94], pode resultar em representações muito inconsistentes, principalmente para grandes quantidades de componentes onde as pessoas que classificam os componentes podem designar a palavra chave.

Uma dificuldade encontrada no esquema de classificação por palavra chave, segundo [HAL94], é que, como na classificação, os termos designados como palavra chave não são controlados, pode ocorrer que, na hora da pesquisa, existam várias palavras chaves que poderiam ter sido designadas para o componente que se procura. Visto que nem sempre a pessoa que classificou o componente é a mesma que deseja recuperá-lo, há dificuldade em saber qual é exatamente a palavra chave que poderia recuperar o componente desejado. Por esta razão há necessidade de se otimizar a pesquisa com a utilização do operador booleano *or*, que irá estender a pesquisa, pois pode-se solicitar uma pesquisa através de mais de uma palavra-chave.

3.3.2.3 ESQUEMA DE CLASSIFICAÇÃO VALOR-ATRIBUTO

Conforme [MCC97] e [FRA94], neste esquema de classificação utilizam-se atributos e valores para efetuar a classificação de componentes. Os atributos não são ordenados e podem ser tantos quantos forem necessários, sendo que cada atributo possui valores que também não são controlados.

Apesar deste esquema de classificação não estar representado na figura 2, [MCC97] o cita como um esquema de classificação que faz parte dos sistemas indexados não controlados, pois não há limites na quantidade de atributos e valores de atributos a serem designados para um componente.

Para recuperar um componente classificado através do esquema de classificação valor-atributo, deve-se selecionar um valor ou grupo de valores para cada atributo. Pode-se também utilizar os operadores booleanos para estender a pesquisa. O componente ou componentes que possuem os atributos e valores indicados na pesquisa serão selecionados.

Este esquema é simples, conforme [MCC97], porém podem ocorrer ambigüidades e redundância nos valores de atributos, pois não há meio de controlá-los na classificação, podendo ser usados diferentes valores para designar a mesma coisa. Como por exemplo, para um determinado atributo “autor”, pode-se dar os valores “desenvolvedor” ou “programador”.

3.3.3 SISTEMAS INDEXADOS CONTROLADOS.

Um sistema indexado controlado, segundo [HAL94], pode ser organizado:

- a) em *thesaurus* ou
- b) por sistemas de classificação.

Um *thesaurus* lista termos aceitáveis que podem ser usados para descrever componentes. Um *thesaurus* é utilizado para controlar o vocabulário utilizado na classificação de componentes através de esquemas de classificação. Este tipo de organização trata do problema da ambigüidade na classificação e na recuperação de componentes.

O termo *thesaurus*, conforme [FRE87], é necessário em determinados esquemas de classificação para agrupar todos os sinônimos em um único conceito, evitando a ambigüidade na classificação ou estendendo a pesquisa de recuperação de componentes. O termo que melhor descrever um grupo de sinônimos será o termo selecionado para representar os demais termos em uma recuperação de componentes.

Segundo [RAD96], um *thesaurus* pode ser apresentado como uma hierarquia (tabela de conteúdo), estrutura ou como uma classificação alfabética. O *thesaurus* contém termos

preferenciais (descritores) para a indexação e termos não preferenciais para sinônimos correspondentes aos termos preferenciais. O termo preferencial é o que classifica um conceito.

Um *thesaurus* pode ser construído antes do componente ser classificado, porém isto exige grande conhecimento do conteúdo que a biblioteca irá conter, ou pode também ser construído após a classificação do componente, quando uma certa quantidade de termos já estiver cadastrada. A desvantagem deste esquema é o grande esforço necessário para construir e manter um *thesaurus*.

3.3.3.1 ESQUEMA DE CLASSIFICAÇÃO ENUMERADA

Conforme [FRA94], neste esquema de classificação os componentes são organizados em níveis hierárquicos e são divididos em classificações exclusivas. A área de domínio é dividida em mutualidade única, normalmente de acordo com a disciplina científica individual e sua taxonomia. Este esquema é representado através de uma estrutura de árvore.

A classificação enumerada também é citada por [LUN95], como o esquema que divide o universo do conhecimento em classes que incluem todas as possíveis classes combinadas, ou seja, subdivisões de cada classe. Estas sub-classes são organizadas em um critério padrão, normalmente de acordo com sua taxonomia e disciplina científica.

O esquema de classificação enumerada é difícil de se montar para áreas com domínio muito dinâmicas, bem como novos desenvolvimentos em uma área podem requerer uma revisão de todo o esquema de classificação[HAL94].

Segundo [RAM98], este tipo de classificação é difícil de ser alterado, uma vez que ele é dividido em classificações exclusivas. Outra desvantagem é que a ambigüidade de componentes pode existir dentro de diferentes hierarquias além da sua inflexibilidade para a inclusão de novos componentes, que faz com que este esquema seja difícil de ser escolhido como classificador de componentes, pois para software, uma classificação flexível é requerida para que possa encontrar as demandas de expansões contínuas e evoluções do software.

3.3.3.2 ESQUEMA DE CLASSIFICAÇÃO FACETA

O esquema de classificação faceta é mais flexível que o enumerado, segundo [HAL94]. A teoria da classificação faceta foi desenvolvida por Ranganathan, baseado na idéia de estrutura de conhecimentos universais, de acordo com o ponto de vista do conhecimento organizacional.

O sistema de classificação faceta pode ser considerado como uma representação conceitual do conhecimento do programador sobre o software é portanto, uma excelente base para construção de uma avançada interface para o usuário ou um mecanismo intermediário “inteligente” para um sistema de software de informações com facilidades, por exemplo, *browsing* em conhecimentos [HAL94].

Segundo [MCC97], num esquema de classificação faceta, a classificação se dá através de um conjunto de facetas ordenadas. Uma faceta descreve várias propriedades de um componente e possui um conjunto de termos e valores finitos que pode assumir. Segundo [HAL94], o esquema de classificação faceta de Prieto-Dias foi projetado de acordo com a teoria de classificação de faceta de Ranganathan e consiste de seis facetas, porém [MCC97] descreve um sistema de classificação com apenas cinco facetas.

Independente da quantidade de facetas, [LUN95] descreve que neste esquema de classificação, componentes são descritos por um grupo de termos padrões ou facetas, onde cada faceta descreve um aspecto chave do software. Este esquema caracteriza-se por ser simples porém bastante eficiente.

Segundo [FRA94], no esquema de classificação faceta, uma área de domínio é analisada em termos básicos que são organizados como facetas. O desenvolvimento de facetas é usualmente feito pela identificação de vocabulários relevantes em um domínio que são então agrupados como termos em facetas.

Um esquema de classificação completo, baseado no esquema faceta, segundo [MCC97], possui cinco facetas e cada faceta possui um conjunto de termos e valores finitos que pode assumir. Apesar do componente ser classificado por várias facetas, uma procura na biblioteca de reuso pode ser feita simplesmente especificando o nome do componente a ser

buscado. Isto reduz o esquema de classificação de facetas para uma palavra-chave de pesquisa.

Baseando-se no esquema proposto por [MCC97], as facetas podem ser:

- a) nome: é o nome do componente;
- b) tipo de componente: é o tipo de componente como um pacote de aplicação, estrutura da aplicação ou classe de objeto, função, procedimento, manual;
- c) área de aplicação: é a área de aplicação do componente, a qual o componente pertence, domínio ou classe. Pode ser geral, para qualquer tipo de aplicação ou específica para determinado domínio de aplicação;
- d) agente: é o formato ou forma do componente, como código, projeto, especificação, documentação, testes;
- e) linguagem: é o ambiente tecnológico em que o componente pode ser usado, como hardware, software, protocolo.

Os autores [FRE87], [HAL94] e [KUT97] descrevem uma sexta faceta, chamada objeto ou assunto, que, conforme [MCC97], não tem sido mais usada por sistemas de classificação mais flexíveis, desde que o objeto a ser classificado seja um parâmetro genérico para qualquer faceta simples.

No trabalho de [FRE87] cita-se a faceta “função”, como de grande relevância para a classificação de um componente, pois descreve o que o componente faz, juntamente com as facetas “objeto” que é o objeto manipulado e a faceta “agente”, que se refere à forma do componente.

Segundo [LUN95], para produtos de software em uma fase inicial do ciclo de vida, o uso somente do esquema de classificação faceta pode dificultar a determinação e seleção das facetas apropriadas para a complexidade e indeterminismo.

Outra desvantagem, segundo [FRE87], ocorre quando este esquema é utilizado em sistemas com grande quantidade de componentes e que crescem continuamente, onde há grandes grupos de componentes similares em classes bem específicas.

3.3.3.3 ESQUEMA DE CLASSIFICAÇÃO FACETA COM A APLICAÇÃO DE UM THESAURUS

Segundo [FRE87], a aplicação de um *thesaurus* ao esquema faceta provêm o agrupamento de termos válidos de determinada faceta que possuem significados ou funções semelhantes, com o intuito de melhorar a performance de recuperação dos componentes.

Primeiramente deve-se escolher as facetas onde será aplicado o *thesaurus*. Pode-se aplicá-lo em uma ou mais facetas. A faceta onde é aplicada o *thesaurus* passa a ser um termo *thesaurus*. Os termos válidos da faceta, que tiverem significados ou funções semelhantes, são chamados de sinônimos. Estes sinônimos são agrupados em um único termo, chamado de termo preferencial. O termo preferencial é o termo que será utilizado para a recuperação de todos os componentes classificados por um dos sinônimos inclusos no termo preferencial.

Na recuperação, a pergunta para a pesquisa é feita através do esquema de classificação faceta e de um *thesaurus*. Os termos válidos para facetas são listados em formato *thesaurus*, para cada faceta que tenha sido denominada como termo *thesaurus*. Para formular a pergunta é necessário entrar com um termo preferencial do *thesaurus* para a faceta.

Com isto serão recuperados todos os componentes que estiverem classificados com um dos sinônimos pertencentes ao termo preferencial escolhido na pesquisa. Por exemplo, se um *thesaurus* referente a faceta função conter como termo principal *add* e sinônimos *increment* e *sum* a recuperação será de todos os componentes que estiverem classificados ou com a função *increment* ou *sum* ou *add*. Ao contrário do esquema faceta sozinho, que iria recuperar somente os componentes que estiverem classificados com a função escolhida na pesquisa, neste exemplo, *add*. Com isto recupera-se componentes que executam a mesma função, porém foram classificados com diferentes valores para uma mesma faceta.

Uma desvantagem deste esquema é o esforço despendido para a construção e a manutenção do *thesaurus*, pois o cadastramento e escolha dos termos preferenciais e sinônimos é feita manualmente.

4 COMPARATIVO ENTRE OS ESQUEMAS DE CLASSIFICAÇÃO

Neste capítulo serão definidos os critérios de comparação a serem utilizados para efetuar a comparação entre os esquemas de classificação bem como os comentários referente aos resultados das comparações feitas.

4.1 CRITÉRIOS DE COMPARAÇÃO

Foram escolhidos alguns esquemas de classificação estudados no capítulo terceiro, com o objetivo de fazer um comparativo entre os esquemas baseado em alguns critérios. Trabalhos como dos autores [MCC97], [HAL94] e [FRA94] não definem um esquema de classificação ideal, ou mais adequado para ser utilizado na classificação de componentes reusáveis. Cada esquema possui suas características, vantagens e desvantagens.

O quadro 1 demonstra alguns diferenciais entre os esquemas de classificação faceta, valor-atributo, palavra-chave, enumerado e o esquema faceta com a aplicação de um *thesaurus*. Estes esquemas foram escolhidos por serem considerados os mais utilizados segundo os trabalhos pesquisados. Os critérios utilizados para o comparativo foram:

- a) flexibilidade: para a avaliação da flexibilidade foi considerada a flexibilidade que o esquema fornece para a inclusão de novos componentes na biblioteca. Para os esquemas em que há dificuldade, ou, que conforme a literatura, exigem muito esforço para a manutenção da biblioteca, foi dado conceito baixo; e para os demais foi dado conceito médio ou alto segundo o grau de dificuldade;
- b) efetividade: para mensurar a efetividade de recuperação de componentes baseou-se num estudo feito com o PROTEUX, mencionado no trabalho de [FRA94], que é um ambiente que suporta os esquemas de classificação enumerado, palavra-chave, faceta e valor-atributo. Foi verificada a capacidade de recuperação de componentes através da avaliação da quantidade de itens relevantes na base de dados versus quantidade relevante de itens recuperados.

- c) ambigüidade: foi avaliada a tendência do esquema de classificação em conter itens ambíguos na classificação e recuperação dos componentes. São considerados como itens, os atributos, valores de atributos, termos de facetas e as palavras-chave. Os esquemas com maior tendência à ambigüidade foram conceituados como *alta*, e os com menor tendência, como *média*. Os esquemas que, conforme a literatura, podem ser eliminados com a aplicação de um *thesaurus* foram conceituados como *baixa*;
- d) amigabilidade: foram considerados como *alta*, os esquemas que se utilizam de termos indexados através de semânticas bem conhecidas, o que facilita a classificação do componente;
- e) esforço necessário para a construção da biblioteca: foi considerado como *grande* para os esquemas que, conforme literatura, exigem grande esforço para a construção da biblioteca e considerado como *médio* para os esquemas que não exigem tanto esforço para a construção da biblioteca.

ESQUEMAS:	FACETA	FACETA C/ <i>THESAURUS</i>	VALOR ATRIBUTO	ENUMERA DO	PALAVRA- CHAVE
Flexibilidade	Média	Média	Média	Baixa	Média
Ambigüidade	Média	Baixa	Alta	Alta	Alta
Amigabilidade	Alta	Alta	Alta	Média	Alta
Efetividade	Boa	Boa	Boa	Boa	Boa
Esforço necessário P/construção da biblioteca	Grande	Grande	Grande	Grande	Médio
Facilidade de compreensão	Boa	Boa	Boa	Boa	Boa

Quadro 1 – Comparativo entre os Esquemas de Classificação

4.2 COMENTÁRIOS

Nesta sessão serão feitos comentários referente a cada critério adotado na comparação dos esquemas de classificação.

4.2.1 FLEXIBILIDADE

Conforme [HAL94], com o objetivo de prover autonomia no desenvolvimento, uma significativa quantidade de pesquisas tem sido feitas sobre o método de classificação faceta.

Conforme [MCC97], métodos por facetas são mais fáceis para desenvolver e ampliar do que o método enumerado, sendo também mais flexíveis, mais precisos, mais apropriados para a ampliação da biblioteca. No esquema de classificação enumerado a inclusão de novos componentes é bastante complexa, visto que a classificação é feita em classes e sub-classes.

Já o método de classificação faceta é mais fácil de ser modificado do que o método enumerado, segundo [FRA94], pois uma faceta pode ser alterada sem afetar as outras facetas, no método de classificação. Os esquemas palavra-chave e valor-atributo também mostraram-se flexíveis para novas inclusões.

4.2.2 AMBIGÜIDADE

Segundo [FRA94], o esquema de classificação valor-atributo é similar para pesquisas em facetas, em que facetas são equivalentes a atributos e termos de facetas são equivalentes para valores. As diferenças são:

- a) no esquema faceta um domínio é descrito utilizando de cinco a sete, ou menos facetas, já para a definição de atributos no esquema valor-atributo, não há limites;
- b) o esquema faceta usualmente provê mais facilidade para manusear sinônimos, visto que termos de faceta são controlados, o que não ocorre em pesquisas de valor-atributo, onde os valores podem ser designados pela pessoa que classifica o componente, assim como também no esquema palavra-chave.

Dentre o esquema valor-atributo e faceta, apesar de o esquema valor-atributo ter maior tendência para a ambigüidade de termos, as pesquisas mencionadas em [FRE87] e [MCC97] propõe soluções somente para o esquema de classificação faceta: a junção de um *thesaurus* com o esquema faceta. Desta forma, acabaria a ambigüidade encontrada no esquema faceta, pois termos de uma faceta são agrupados como sinônimos e é determinado um termo preferencial que melhor represente os sinônimos.

O esquema de classificação por palavra-chave também tem problemas com a ambigüidade, pois as palavras-chave utilizadas na classificação dos componentes não são controladas, podendo o indexador ou a pessoa que classifica, designar a palavra que melhor lhe convêm para a representação de determinado componente.

No esquema de classificação enumerado, também há o problema da ambigüidade, pelo fato de um componente poder existir dentro de diferentes hierarquias.

4.2.3 EFETIVIDADE

Segundo [FRA94], com base num projeto experimental com o ambiente PROTEUX, que é uma biblioteca de reuso que suporta múltiplos esquemas de classificação, como o valor-atributo, enumerado, palavra-chave e faceta, chegou-se a conclusão de que:

- a) não há diferenças muito significantes entre os esquemas, em se tratando de efetividade (a capacidade de recuperação de componentes, que é avaliada pela quantidade de chamadas(número relevante de itens em uma Base de dados) e pela precisão (número relevante de itens recuperados)).
- b) todos os esquemas recuperam a mesma quantidade de componentes, porém eles encontram itens diferentes.

4.2.4 AMIGABILIDADE

Todos os esquemas comparados, exceto o enumerado, são bastante amigáveis, pois pode-se utilizar termos do vocabulário para a classificação do componente. Na classificação enumerada as classes e sub-divisões de classes são organizadas em um critério padrão, normalmente de acordo com sua taxonomia e disciplina científica, o que torna os termos a serem utilizados menos amigáveis.

4.2.5 FACILIDADE DE COMPREENSÃO

Todos os esquemas de classificação proporcionam um bom entendimento dos componentes recuperados, desde que sejam classificados de acordo com suas características.

A facilidade na compreensão do componente é um dos objetivos da utilização dos esquemas de classificação, portanto, estando o componente classificado já facilita a sua compreensão na recuperação.

4.2.6 ESFORÇO NECESSÁRIO PARA A CONSTRUÇÃO DA BIBLIOTECA

Praticamente todos os métodos utilizados neste comparativo exigem grande esforço para a construção da biblioteca. No esquema faceta, é necessário primeiramente que se determine quais as facetas que serão utilizadas para classificar os componentes. Após, devem ser incluídos em cada faceta todos os termos válidos para a mesma. Este é um trabalho bastante complexo quando se trata de uma grande base de componentes a serem reusados.

No esquema de classificação enumerado, o esforço é ainda maior, visto que os componentes são classificados primeiramente em áreas específicas e após em sub-áreas.

No esquema valor-atributo, deve-se primeiramente escolher os atributos que melhor irão classificar o componente. Apesar da quantidade de atributos ser ilimitada, os mesmos devem expressar as características mais importantes de um componente, para que não haja redundância ou atributos que não irão caracterizar adequadamente um componente.

No esquema palavra-chave a construção da biblioteca é moderadamente fácil, pois os termos e a quantidade de termos para a classificação é ilimitada, bastando haver um certo conhecimento do componente por parte do classificador.

5 ESPECIFICAÇÃO E IMPLEMENTAÇÃO DO PROTÓTIPO

Neste capítulo será apresentada a especificação e implementação de uma biblioteca de componentes reusáveis onde serão classificados e recuperados componentes utilizando os esquemas de classificação faceta, valor-atributo, palavra-chave e faceta com a aplicação de um *thesaurus*. Para a especificação do sistema foi utilizada a ferramenta Power Design 6.1 da Sysbase e a implementação foi desenvolvida no ambiente NetExpress 2.0, que integra todas as ferramentas necessárias para escrever, editar, compilar e executar aplicações em linguagem COBOL, sendo utilizados arquivos indexados para a armazenagem dos dados. Optou-se pelo ambiente NetExpress pelo fato do autor deste trabalho ter conhecimentos e experiência profissional neste ambiente e pelo fato de que esta linguagem se aplica a este problema.

5.1 DIAGRAMA DE CONTEXTO

O Diagrama de Contexto do sistema de classificação de componentes foi desenvolvido na ferramenta case PowerDesigner 6.1 e é representado pela figura 3. O mesmo define as entidades externas ao sistema e a integração destas entidades com o sistema.

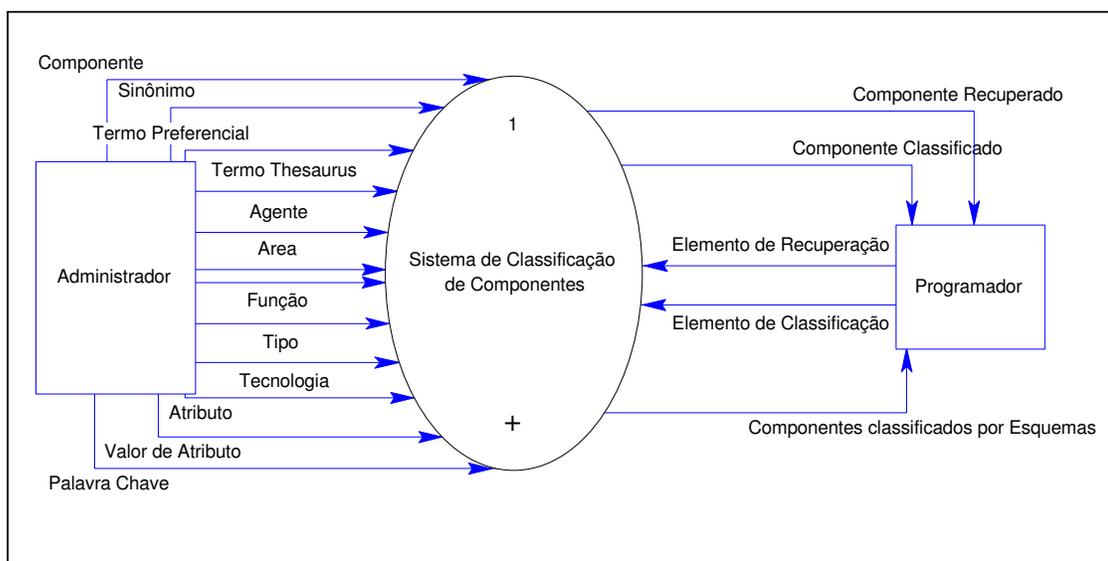


Figura 3 – Diagrama de Contexto

5.2 DFD NÍVEL 0

O DFD nível 0 do sistema foi desenvolvido na ferramenta case PowerDesigner 6.1 e é representado pela figura 4, 5 e 6.

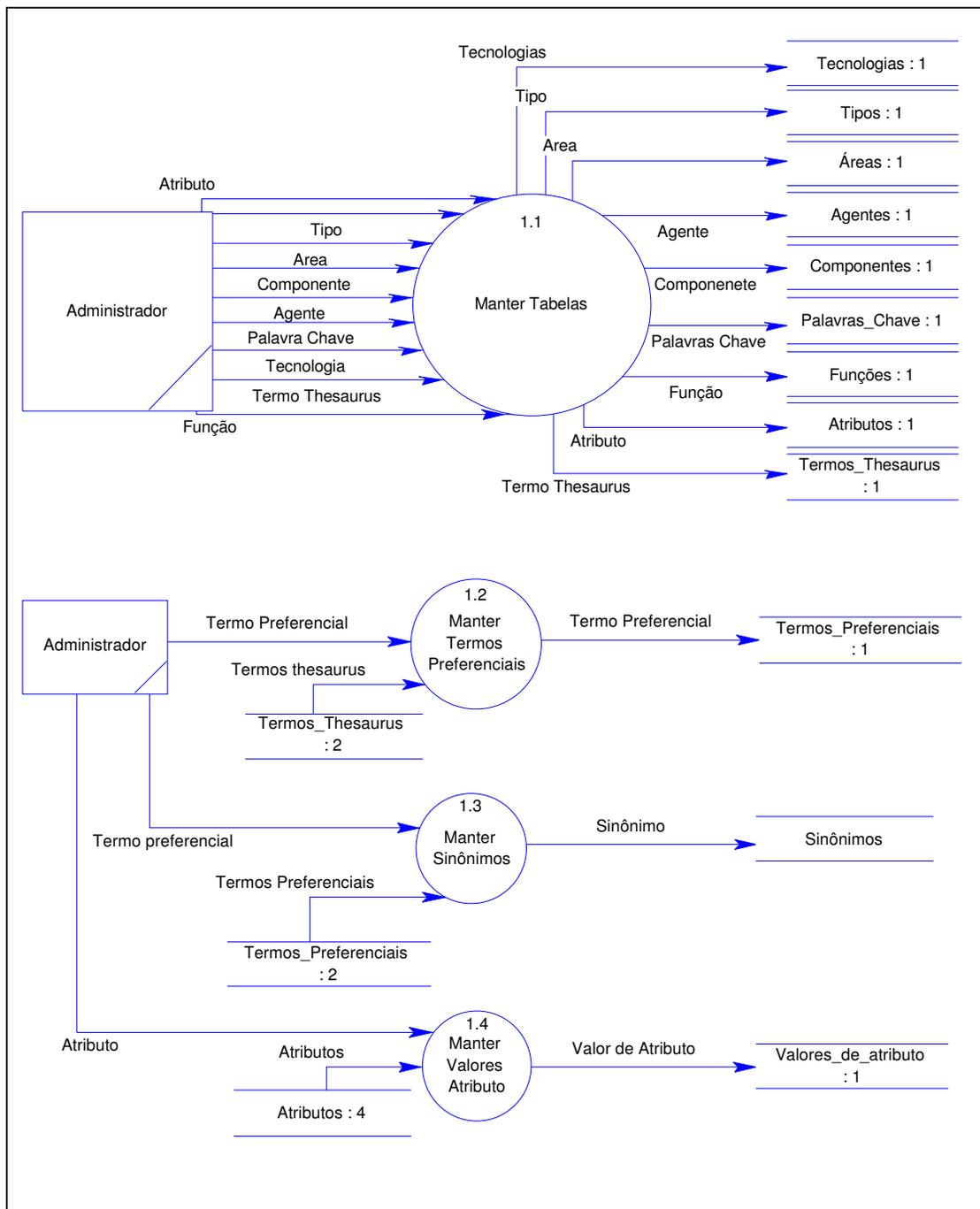


Figura 4 – DFD Nível 0

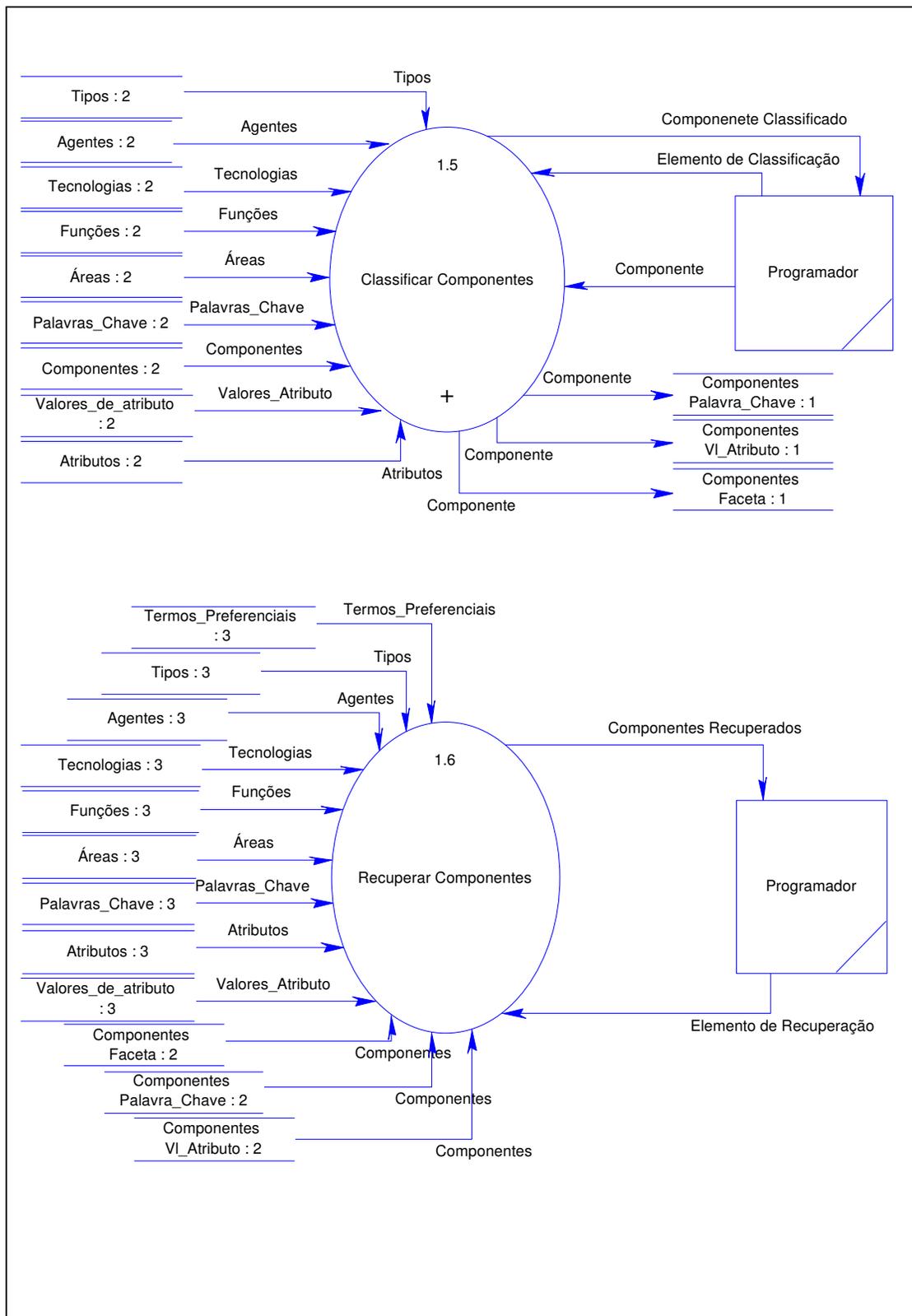


Figura 5 – Continuação DFD Nível 0

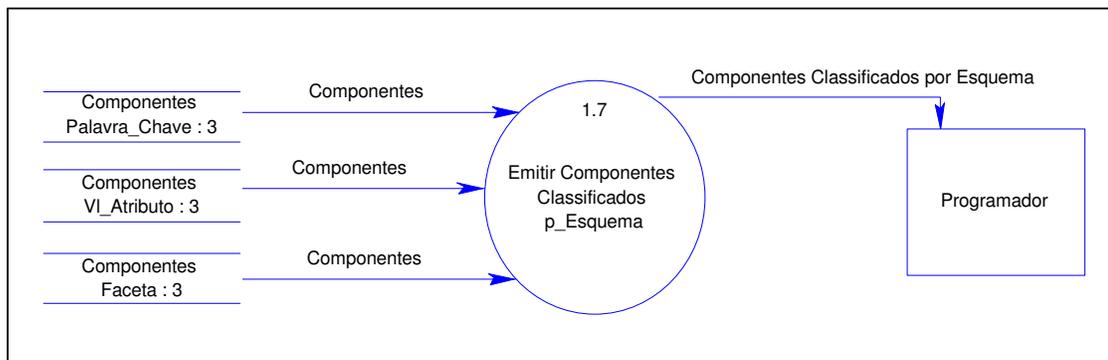


Figura 6 – Continuação DFD Nível 0

5.3 MODELO ENTIDADE-RELAÇONAMENTO

O Modelo Entidade-Relacionamento do sistema foi desenvolvido na ferramenta case PowerDesigner 6.1 e é representado pela figura 7.

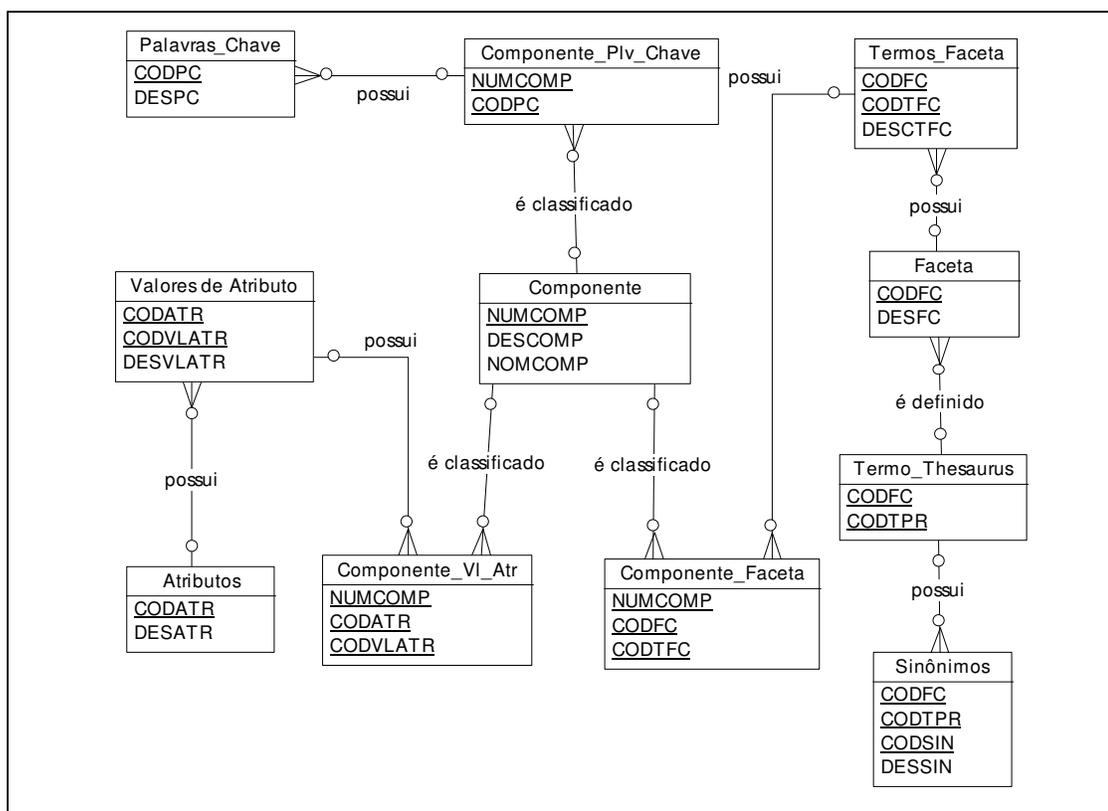


Figura 7 – Modelo Entidade-Relacionamento

5.4 DICIONÁRIO DE DADOS

O Dicionário de Dados do sistema de classificação de componentes foi desenvolvido na ferramenta case PowerDesigner 6.1 e é demonstrado no quadro 2.

Name	Code	Type
CODATR	CODIGO_DO_ATRIBUTO	I
CODFC	CODIGO_DA_FACETA	I
CODPC	CODIGO_PALAVRA_CHAVE	I
CODSIN	CODIGO_SINONIMO	I
CODTFC	CODIGO_TERMO_FACETA	I
CODTPR	COD_TERMO_PREFERENCIAL	I
CODVLATR	CODIGO_VALOR_ATRIBUTO	I
DESATR	DESCRICAO_DO_ATRIBUTO	A30
DESCOMP	DESCRICAO_DO_COMPONENTE	A300
DESCTFC	DESCRICAO_TERMO_FACETA	A30
DESFC	DESCRICAO_DA_FACETA	A30
DESPC	DESCRICAO_PALAVRA_CHAVE	A30
DESSIN	DESCRICAO_DO_SINONIMO	A30
DESVLATR	DESCRICAO_DO_VL_ATRIBUTO	A30
NOMCOMP	NOME_DO_COMPONENTE	A8
NUMCOMP	NUMERO_COMPONENTE	I

Quadro 2 – Dicionário de Dados

5.5 APRESENTAÇÃO DO SISTEMA

O sistema de classificação de componentes foi implementado no ambiente Net Express 2.0, que possui as ferramentas necessárias para a edição, compilação e execução de aplicações em linguagem COBOL, conforme [FOC97]. Este sistema pode ser incluído num modelo de ciclo de vida do software que seja voltado para o reuso. No trabalho de [SUR97] é apresentado um modelo baseado na reutilização que poderá facilitar a inclusão deste sistema ao modelo de ciclo de vida do software. O sistema tem na tela principal as opções de cadastramento, classificação, consulta, relatórios e help, conforme ilustrado na figura 8.

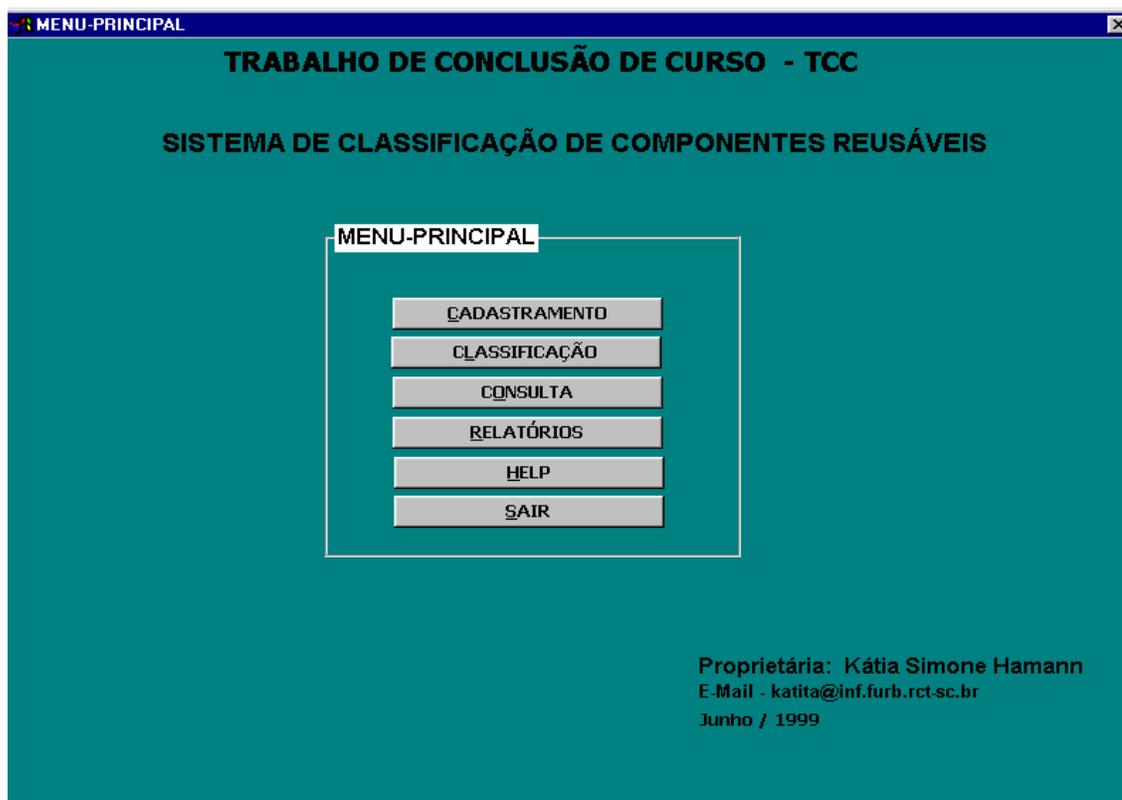


Figura 8 – Menu Principal do Sistema de Classificação de Componentes

Na opção cadastramento, classificação ou consulta é apresentada uma tela para a escolha do esquema de classificação em que se deseja cadastrar itens necessários para a classificação de um componente (veja figura 9). Esta tela também é utilizada para a escolha do esquema na classificação ou consulta de um componente. Visto que cada esquema de classificação apresenta características próprias, foram desenvolvidos diferentes telas para cada esquema, tanto na consulta quanto na recuperação de componentes.



Figura 9 – Menu Escolha de Esquema de Classificação

Todos as opções serão detalhadas nos próximos capítulos.

5.5.1 CADASTROS

No sistema proposto deve-se efetuar o cadastramento de alguns valores e termos necessários para a classificação e recuperação de componentes, tais como:

termos válidos de facetas, para a classificação de componentes no esquema faceta,
atributos, para a classificação através de valor-atributo,
termo *thesaurus*, termo preferencial e sinônimos para a utilização de um *thesaurus* na
recuperação de componentes através do esquema de classificação faceta.

Para os esquemas de classificação faceta e valor-atributo, é necessária a inclusão de facetas e valores válidos para cada faceta quando se tratar do esquema de classificação faceta e a inclusão de atributos válidos para a classificação de um componente através do esquema

de classificação valor-atributo. Além destes, para a classificação de um componente através do esquema faceta e *thesaurus*, será necessário a inclusão dos termos que irão ser incluídos no *thesaurus*. Para a inclusão de qualquer um destes termos deve-se selecionar o botão Cadastramento na tela Menu Principal e pelo esquema em que se deseja incluir os itens.

Na escolha do esquema de classificação faceta é apresentada a tela ilustrada na figura 10. Nela deve-se cadastrar os valores válidos para cada uma das facetas.

CADASTRO DE TERMOS DE FACETAS

CADASTRO

ESQUEMA DE CLASSIFICAÇÃO FACETA

ITENS PARA CADASTRAMENTO

<u>FACETAS</u>	<u>TERMOS DE FACETAS</u>
TIPOS DE COMPONENTE	
AGENTES	
FUNÇÕES	
TECNOLOGIAS	ADD ADD DEL DELETE INCREMENT
ÁREAS DE APLICAÇÃO	

SAIR **HELP**

Mensagem
Digite os termos válidos para cada faceta. Tecele enter para gravar e PGUP para pular para próxima faceta
Para cancelar um termo, teclar ESC

Figura 10 – Tela de Cadastro de Valores Válidos para Facetas

Caso tenha-se optado pelo cadastramento e o esquema de classificação valor-atributo, será apresentada a tela para a inclusão dos atributos válidos a serem utilizados para a classificação e recuperação de componentes, conforme mostrado na figura 11. Poderão ser cadastrados no sistema no máximo cinco atributos.

CADASTRAR ATRIBUTOS

CADASTRO DE ATRIBUTOS
ESQUEMA DE CLASSIFICAÇÃO VALOR-ATRIBUTO

NOME ATRIBUTO... :

Atributos Designados

- TIPO
- FUNCAO
- LINGUAGEM
- AREA

HELP SAIR

Mensagem

Digite o nome dos atributos e tecle enter para gravar ou esc para cancelar

Figura 11 – Tela Cadastro de Atributos no Esquema de Classificação Valor-Atributo

Na escolha do esquema de classificação faceta/*thesaurus*, será apresentada a tela conforme figura 12. Nela, deve-se optar pela faceta que se deseja incluir como termo *thesaurus*. Deve-se também escolher o termo preferencial através da seleção de um dos termos válidos da faceta escolhida como termo *thesaurus*. Os sinônimos devem ser escolhidos através de duplo clique sob os termos válidos da faceta.

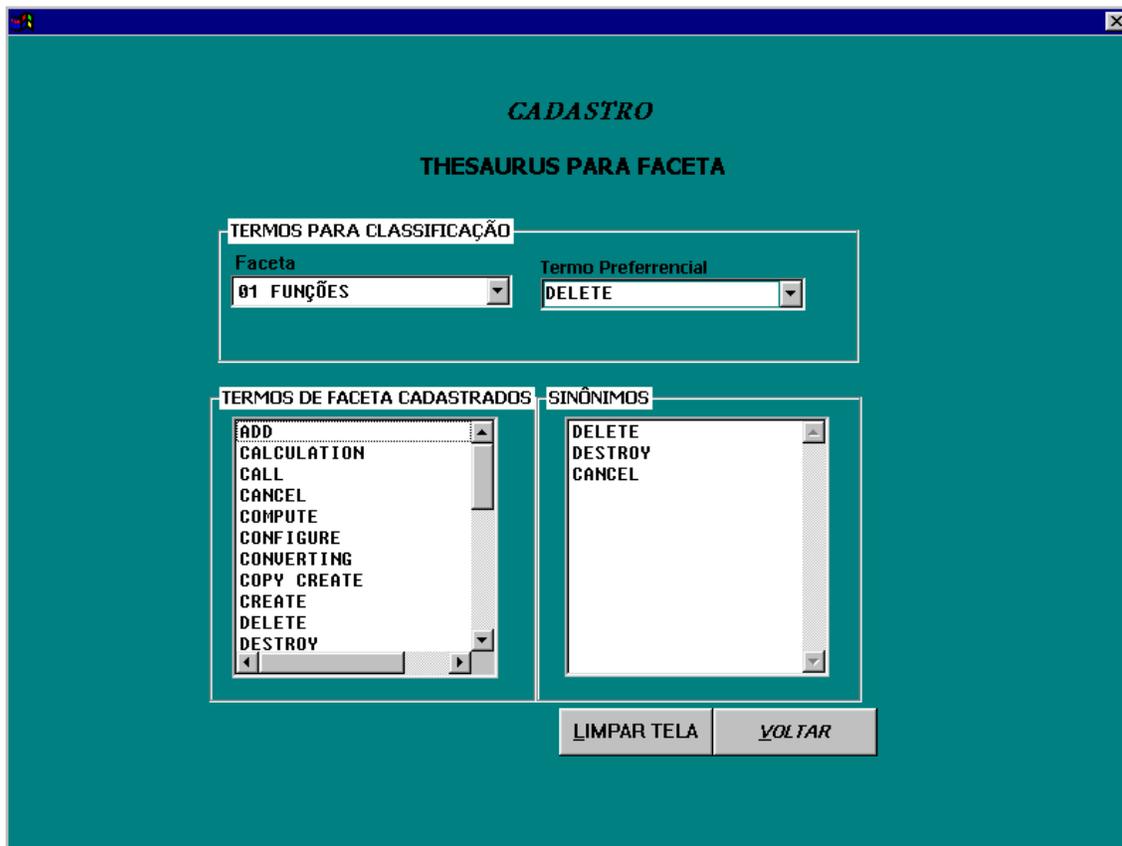


Figura12 – Tela Cadastro do Esquema de classificação Faceta com *Thesaurus*

5.5.2 CLASSIFICAÇÃO DE COMPONENTES

Assim como na tela de cadastramento, nesta opção também primeiramente é solicitado ao usuário a escolha do esquema de classificação ao qual se deseja classificar o componente a ser reusado. Para cada esquema de classificação foram desenvolvidas telas específicas, que solicitam os dados necessários referente o componente a ser classificado.

Para escolher um componente, basta clicar sobre o botão Componente p/Classificar, para que seja apresentada uma janela contendo uma lista com código e nome de todos os componentes possíveis de serem reusados, conforme tela apresentada na figura 13. Estes componentes já se encontram armazenados em um repositório e nesta etapa eles são apresentados para que possam ser classificados e incluídos em uma biblioteca de componentes reusáveis. A escolha do componente deve ser feita através de duplo clique sob o mesmo.

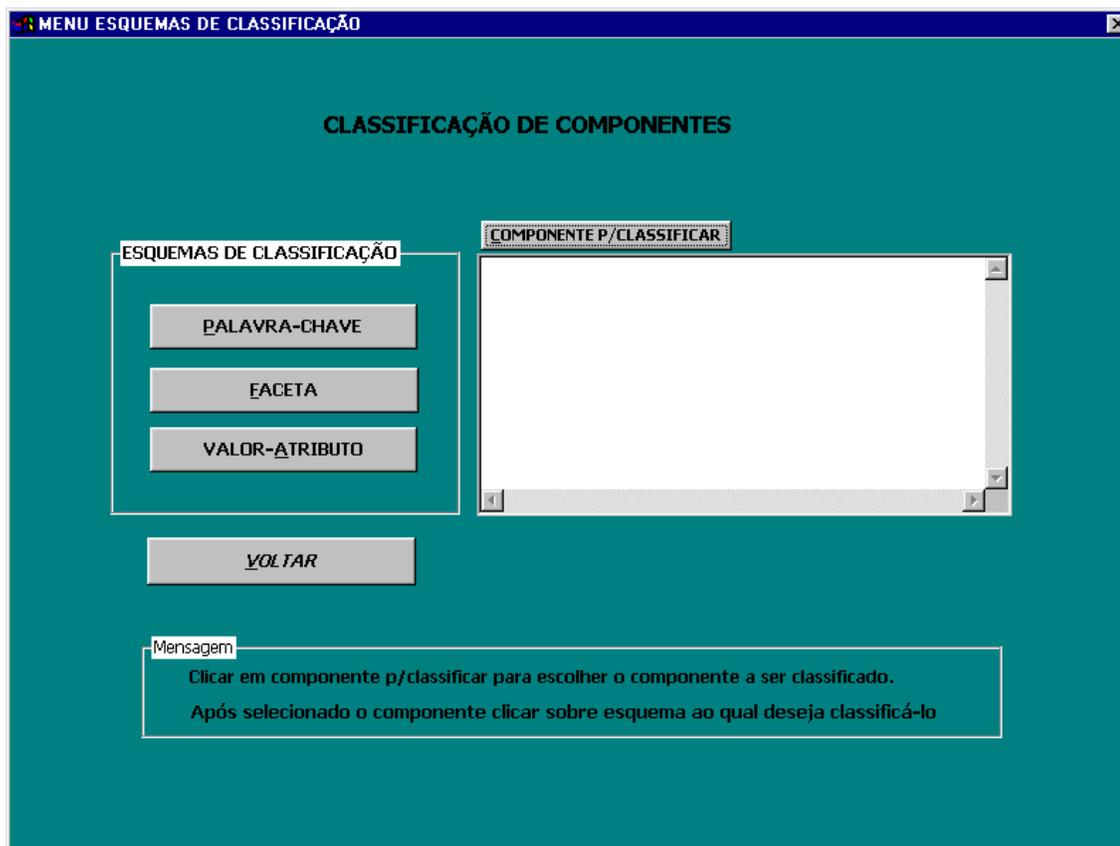


Figura 13 – Tela para Escolha do Componente a ser Classificado e do Esquema de Classificação

Após escolhido o componente, deve-se escolher o esquema de classificação, onde será apresentada nova tela para cada esquema. Através desta janela também pode-se consultar e alterar dados de componentes já classificados e inclusos na biblioteca anteriormente. O processo é o mesmo. Após o componente estar classificado o mesmo é armazenado em uma biblioteca de componentes reusáveis. Foram criadas bibliotecas para armazenar componentes de cada um dos esquema de classificação, ou seja, um componente quando classificado por determinado esquema é armazenado na biblioteca de componentes reusáveis classificados através do mesmo esquema de classificação.

Para a classificação de um componente através do esquema de classificação valor-atributo, é apresentada a tela da figura 14, onde deverão ser incluídos os valores de atributos válidos para um determinado componente.

CLASSIFICAÇÃO POR VALOR-ATRIBUTO

CLASSIFICAÇÃO DE COMPONENTES
ESQUEMA DE CLASSIFICAÇÃO VALOR-ATRIBUTO

Componente ++

001 MOV_DT_SIST

ATRIBUTOS	VALOR DO ATRIBUTO
TIPO	CODIGO
FUNCAO	MOVE
LINGUAGEM	COBOL
AUTOR	KATIA S. HAMANN
AREA	COMERCIAL

GRAVAR VOLTAR

Figura 14 – Tela de Classificação de Componente no Esquema Valor-Atributo

Para a classificação do componente, é necessário que seja escolhido um valor de atributo para cada atributo. Os atributos já foram definidos no cadastramento, conforme figura 11. Cada valor de atributo é armazenado em um arquivo que conterà todos os valores de atributos já utilizados para a classificação de outros componentes. Para a classificação de um componente pode-se escolher um dos valores já incluídos ou ainda pode-se incluir um novo valor de atributo. Todos os atributos devem conter valores.

Para a opção do esquema de classificação palavra-chave, será apresentada a tela conforme figura 15.

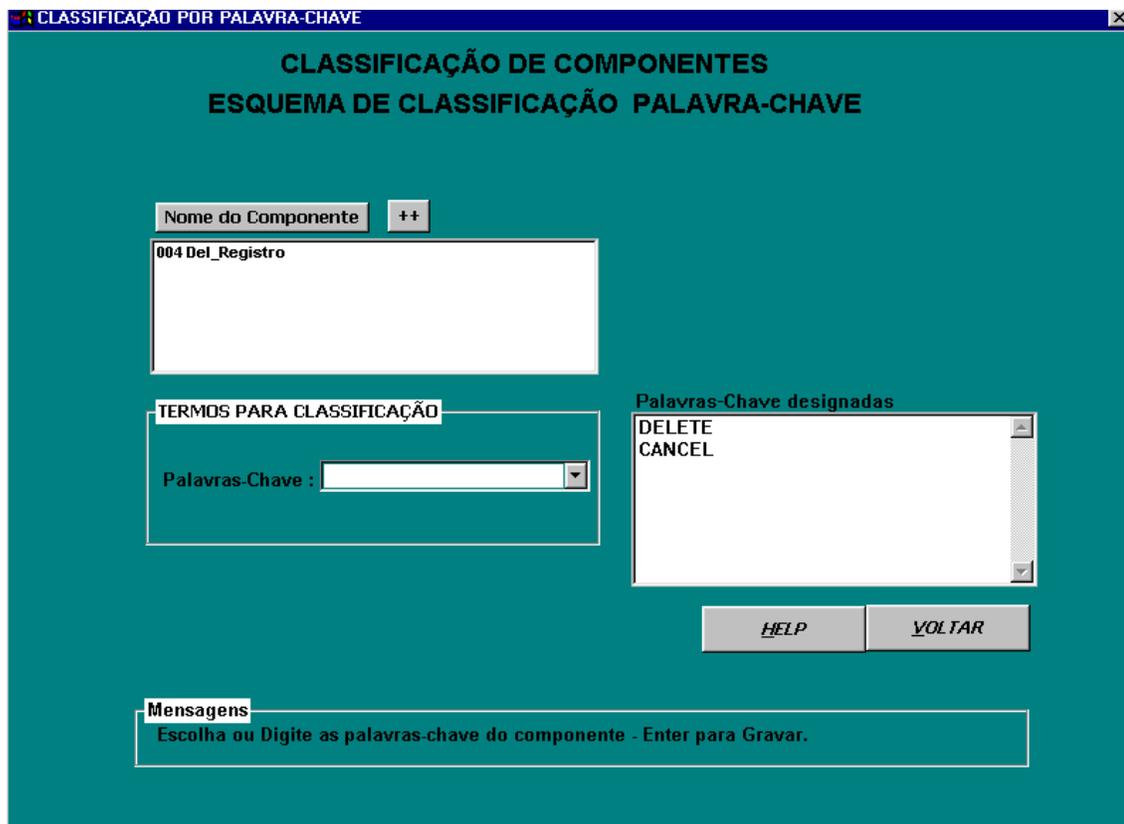


Figura 15 – Tela de Classificação de Componentes no Esquema Palavra-Chave

Para classificar um componente através do esquema palavra-chave, basta digitar a ou as palavras-chave referente ao componente a ser classificado. Assim como no esquema de classificação valor-atributo, neste esquema, as palavras chaves designadas são incluídas em um arquivo que conterà todas as palavras chaves já utilizadas para classificar componentes.

Neste caso, pode-se também classificar um componente com a utilização de uma palavra-chave já utilizada anteriormente. Para isto basta dar duplo clique sob a palavra-chave desejada.

Para a classificação de um componente através do esquema de classificação faceta, é apresentada a tela conforme figura 16.

CLASSIFICAÇÃO FACETA

CLASSIFICAÇÃO DE COMPONENTES
ESQUEMA DE CLASSIFICAÇÃO FACETA

Componente ++

004 Del_Registro

FACETAS E TERMOS

Nome: deletar_reg

Tipo: CODIGO

Agente: ARQUIVO

Função: DELETE

Tecnologia: NETEXPRESS

Área: CONTAS A RECEBER

HELP ARMAZENAR VOLTAR

Figura 16 – Tela de Classificação de Componentes no Esquema Faceta

Neste esquema, as facetas e os termos válidos para as facetas já foram definidas no cadastramento. Portanto, para classificar um componente, deve-se escolher um termo dentre os termos válidos determinados para cada faceta. Esta escolha deve ser feita clicando-se sob o termo válido escolhido de cada faceta.

5.5.3 CONSULTA E RECUPERAÇÃO DE COMPONENTES

Após classificados e armazenados na biblioteca, os componentes estão prontos para serem recuperados e reutilizados em novos projetos. Primeiramente, o usuário precisa optar pela opção Consulta, no Menu-Principal e após escolher o esquema de classificação ao qual deseja efetuar a recuperação de componentes. O sistema permite que se recupere somente componentes que estiverem classificados pelo esquema de classificação escolhido nesta etapa.

Para o esquema de classificação valor-atributo, será apresentada a tela mostrada na figura 17, onde, ao se escolher um atributo, serão apresentados na janela Lista de Valores do Atributo, os valores já cadastrados referentes a este atributo. Deve-se escolher um dos valores e optar por um dos operadores booleanos que estão ao lado da janela Lista de Valores de Atributos. Os atributos, valores dos atributos e o operador escolhidos serão apresentados nas janelas Atributos para Pesquisa, Valores para pesquisa e Operadores respectivamente.

Os componentes encontrados serão relacionados na janela Componentes Recuperados.



Figura17 – Tela de Recuperação de Componentes no Esquema Valor-Atributo

Para a recuperação de componentes através do esquema de classificação palavra-chave a tela apresentada é a mostrada na figura 18.

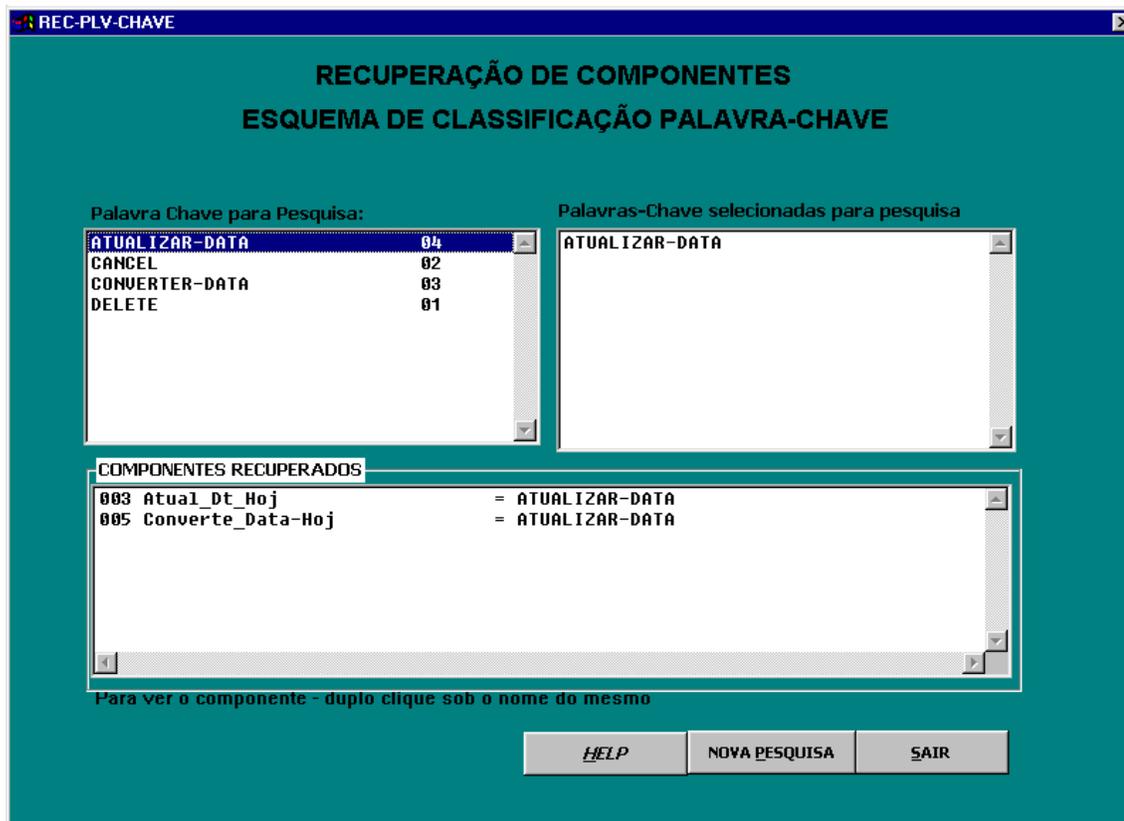


Figura 18 – Tela de Recuperação de Componentes no Esquema Palavra-Chave.

Para recuperar componentes através do esquema de classificação palavra-chave, deve-se escolher a ou as palavras-chaves contidas na base de dados. As palavras-chaves selecionadas para efetuar a pesquisa serão listadas na janela Palavras-Chaves Selecionadas para Pesquisa. Todos os componentes que estiverem classificados com uma das palavras-chaves selecionadas para pesquisa irão ser recuperados e apresentados na janela Componentes Recuperados. Além do código e nome do componente, também é apresentada a palavra-chave cujo o componente foi classificado.

Para a recuperação de um componente através do esquema de classificação faceta será apresentada a tela ilustrada na figura 19.

**RECUPERAÇÃO DE COMPONENTES
ESQUEMA DE CLASSIFICAÇÃO FACETA**

Seleção de Facetas e Termos para Pesquisa

Tipo:

Tecnologia:

Agente:

Área:

Função:

Facetas e Termos selecionados para pesquisa

Tipo:

Tecnologia:

Agente:

Área:

Função:

COMPONENTES RECUPERADOS

001 File_descrip_Pa
002 Conf_impresora
003 ATUALIZAR-DATA

Para ver o componente - duplo clique sob o nome do mesmo

Figura 19 – Tela de Recuperação de Componentes no Esquema Faceta.

Para recuperar componentes através do esquema de classificação faceta, deve-se optar por um dos valores válidos de cada faceta, que estão contidos na base de dados. Os valores selecionados para pesquisa são mostrados na janela Facetas e Termos Selecionados para a Pesquisa.

Não é necessário preencher os valores de todas as facetas. Após selecionados os termos de facetas deve-se clicar no botão Pesquisar e os componentes recuperados serão apresentados na janela Componentes Recuperados. Para a recuperação de componentes através do esquema de classificação faceta juntamente com um *thesaurus*, a tela é idêntica, porém os termos válidos de uma faceta que tenha sido determinada como termo *thesaurus*, irão aparecer em forma de *thesaurus*, ou seja, somente os termos preferenciais. Escolhendo o termo preferencial serão recuperados todos os componentes que estiverem classificados com um dos sinônimos relacionados ao termo preferencial.

5.5.4 RELATÓRIOS

A figura 20 demonstra a tela de opções para emissão de relatórios. Podem ser emitidos relatórios dos componentes que foram classificados em um determinado esquema de classificação. No relatório consta o número e nome do componente classificado e os termos utilizados para a sua classificação, conforme ilustrado na figura 21 com o esquema de classificação faceta.

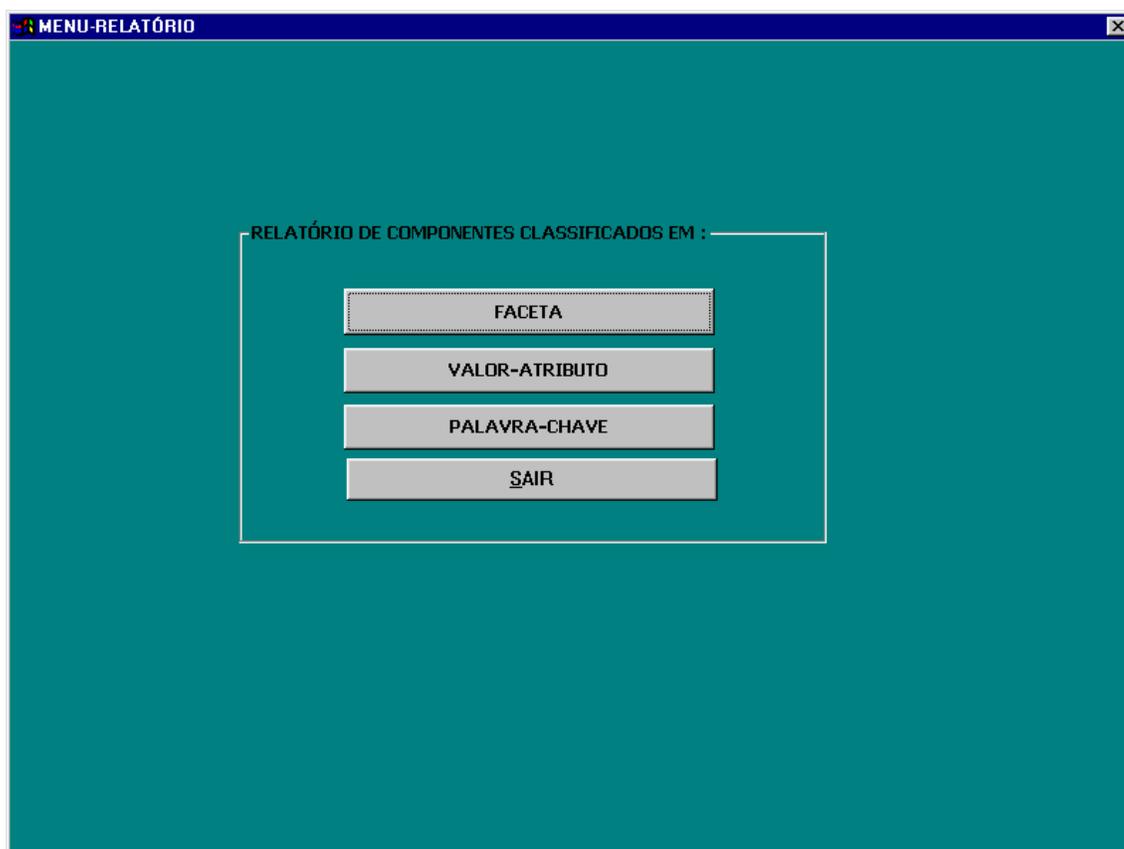
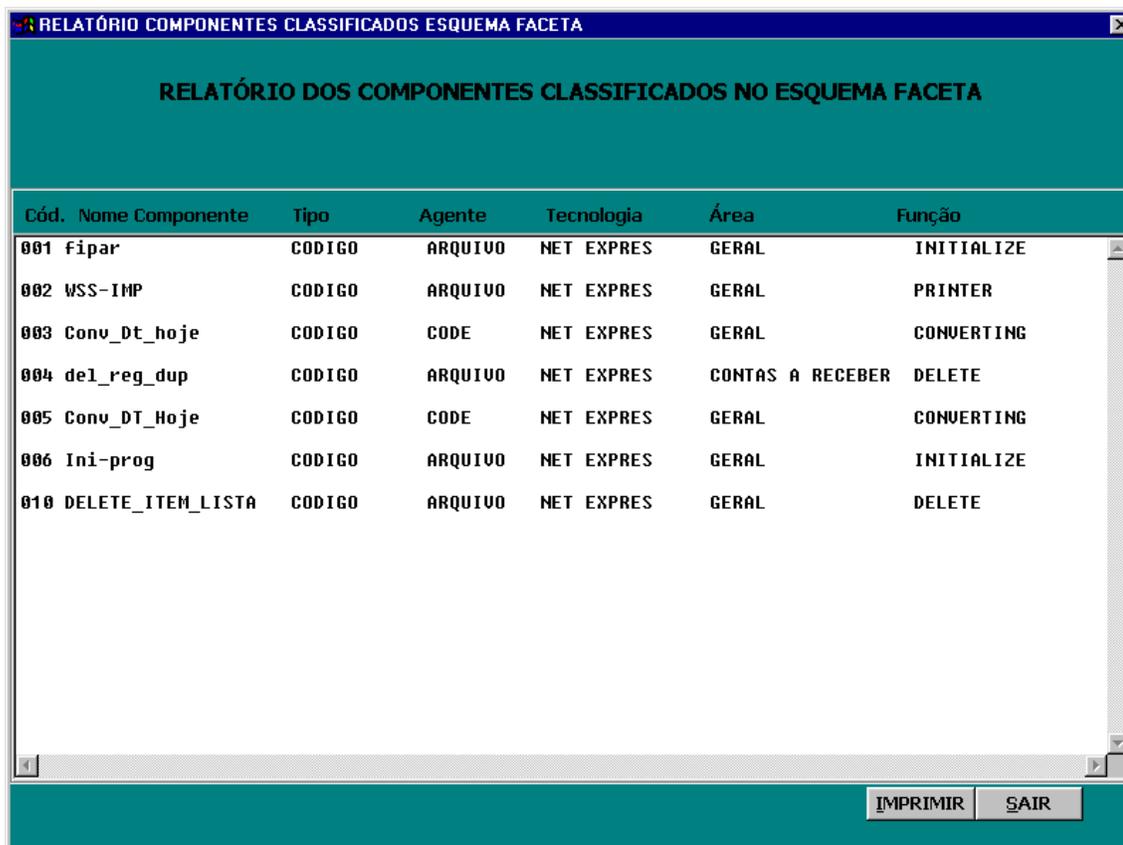


Figura 20 – Tela Menu Relatórios



Cód.	Nome Componente	Tipo	Agente	Tecnologia	Área	Função
001	fipar	CODIGO	ARQUIVO	NET EXPRES	GERAL	INITIALIZE
002	WSS-IMP	CODIGO	ARQUIVO	NET EXPRES	GERAL	PRINTER
003	Conv_Dt_hoje	CODIGO	CODE	NET EXPRES	GERAL	CONVERTING
004	del_reg_dup	CODIGO	ARQUIVO	NET EXPRES	CONTAS A RECEBER	DELETE
005	Conv_DT_Hoje	CODIGO	CODE	NET EXPRES	GERAL	CONVERTING
006	Ini-prog	CODIGO	ARQUIVO	NET EXPRES	GERAL	INITIALIZE
010	DELETE_ITEM_LISTA	CODIGO	ARQUIVO	NET EXPRES	GERAL	DELETE

Figura 21 – Tela Relatório de Componentes Classificados por Faceta

6 CONCLUSÃO

CONSIDERAÇÕES FINAIS

A reusabilidade de software é um importante conceito que uma empresa desenvolvedora de software precisa implantar para que possa obter maior produtividade e qualidade no desenvolvimento de sistemas.

Com uma organização cuja cultura esteja voltada para o reuso, de forma que todos os integrantes da organização utilizem o reuso no seu cotidiano, pode-se obter melhores resultados em um curto espaço de tempo.

Estudos voltados a esta área têm seu enfoque voltado a facilitar a implantação do reuso em uma organização. Esquemas de classificação utilizados para classificar componentes reusáveis são, sem dúvida, de grande importância para a reusabilidade de software como um todo, visto que facilitam a recuperação e entendimento do componente.

De acordo com os testes realizados no sistema, o esquema de classificação palavra-chave é o que oferece maior facilidade de uso, pois não há restrições nos termos nem limitações para a classificação de um componente.

Já o esquema de classificação faceta limita os termos a serem utilizados para a classificação. O componente é classificado através dos termos válidos para cada faceta, que anteriormente já foram definidos e não podem ser alterados pelo classificador. Isto significa que o cadastramento de termos válidos para cada faceta deve contemplar todas os possíveis termos para classificar um componente.

No esquema valor-atributo, além de se poder classificar um componente através de quantos atributos forem desejados, os valores de cada atributo são determinados pelo próprio classificador, o que pode resultar em ambigüidade de valores, dificultando a recuperação de componentes.

Durante os testes efetuados na recuperação de componentes pôde-se observar que o esquema de classificação faceta torna o trabalho mais prático, pois os termos válidos são limitados, o que não quer dizer que este seja o esquema mais adequado. Ao contrário do esquema faceta, nos esquemas palavra-chave e valor-atributo, têm-se a opção de escolher quantos valores forem necessários para a recuperação de determinado componente. No esquema valor-atributo tem-se a vantagem de limitar ou estender a pesquisa com a utilização dos operadores booleanos *and* e *or*, tornando a pesquisa mais complexa que no esquema palavra-chave que utiliza somente o operador booleano *or* e o esquema faceta que utiliza somente o operador booleano *and*.

A recuperação através do esquema faceta mostrou-se mais eficiente com a utilização de um *thesaurus*, pois a junção destes esquemas possibilitou uma melhor performance na recuperação de componentes.

O protótipo do sistema de classificação de componentes reusáveis demonstrado neste trabalho, está limitado a classificação e recuperação de componentes do tipo código.

Considerando que o objetivo do protótipo é didático, pode-se dizer que o objetivo foi alcançado, visto que foram demonstradas a forma de classificação e forma de recuperação de alguns esquemas de classificação, tornando possível identificar as principais características de cada esquema de classificação.

6.1 SUGESTÕES

Algumas sugestões para futuros trabalhos nesta área são:

desenvolvimento de uma ferramenta para auxiliar na classificação dos componentes.

Este processo ainda é feito quase que em sua totalidade manualmente pela pessoa que classifica os componentes através de seus conhecimento e experiência;

estudo e a implementação de uma biblioteca utilizando a junção de mais de um esquema de classificação para classificar um determinado componente. Pesquisas a este respeito já vem sendo feitas, com a finalidade de melhorar a performance, eliminar a ambigüidade e tornar a classificação simples, porém muito pouca literatura tem-se encontrado sobre este assunto;

implementação de uma ferramenta de gerenciamento de configuração dos componentes criados e incluídos na biblioteca de componentes.

implementação de um sistema de classificação de componentes que utilize vários esquemas de classificação onde os dados utilizados para classificar os componentes são compartilhados entre os esquemas e que permita a classificação e recuperação de diversos tipos de componentes, como código, especificação.

REFERÊNCIAS BIBLIOGRÁFICAS

- [BIA95] BIANCHI, Fábio Campos; NASCIMENTO, Maria Elenita M.; **Workshop Qualidade de Software. IX Simpósio Brasileiro de Engenharia de Software.** Recife, 1995
- [FOC97] FOCUS, Micro. **NetExpress – Getting Started.** Copyright : Micro Focus, 1997.
- [FRA94] FRAKES, William B.; POLE, Thomas P.. **International Conference on Software Reuse: Advances in Software Reusability. An Empirical Study of Representation Methods for Reusable Components.** Los Alamitos : IEEE -Computer Society Press, 1994.
- [FRE87] FREEMAN, Peter. DIAZ-PRIETRO, Rubén. **Classifying Software for Reusability.** Los Alamitos : Institute of Electrical and Electronics Engineers - IEEE – Computer Society Press, 1987.
- [HAL94] HALL, P.V. & Chapman. **Software Reuse and Reverse Engineering in Practice.** London : Chapman & Hall, 1994.
- [KUT97] KUTOVA, Marcos André Silveira; Macedo, Alessandra Alaniz. **Reuso de Software.** Trabalho de Conclusão de Curso – Ciências Exatas, Universidade de São Paulo. São Carlos, 1997.
- [LUN95] LUNG, Chung-Gorng; URBAN, Joseph E.. **An Approach to the Classification of Domain Models in Support of Analogical Reuse.** Department of Computer Science and Engineering. Arizona : ACM, 1995.
- [MAF93] MAFFEO, Bruno. **Engenharia de Software e Especificações de Sistemas.** Rio de Janeiro : Campus, 1992.
- [MCC93] MCCLURE, Carma. **The Three Rs Of The Software Automation: Reengineering, Reusability, Repository.** New Jersey : Prentice Hall, 1993.

- [MCC94] MCCLURE, Carma. **Reuse Engineering: Extending Information Engineering to Enable Software Reuse.** 02/03/99 Endereço Eletrônico: <http://www.reusability.com/paper4.htm>.
- [MCC97] McCLURE, Carma. **Software Reuse Techniques – Adding Reuse to the System Development Process.** New Jersey : Prentice Hall, 1997.
- [RAD96] RADA, Roy. **Software Reuse: Principles, Methodologies and Practices.** Oxford : Intellect, 1996.
- [RAM98] RAMOS, Débora Cristina Leira; **Ferramenta para Gerenciamento de Componentes Reutilizáveis em ACCESS.** Blumenau, 1998. Trabalho de Conclusão de Curso (Centro de Ciências Exatas e Naturais). Universidade de Blumenau - FURB.
- [SUR97] SURDI, Luciano Rodrigo; **Um Modelo de Ciclo de Vida do Software Baseado na Reutilização.** Blumenau, 1997. Trabalho de Conclusão de Curso (Centro de Ciências Exatas e Naturais). Universidade de Blumenau - FURB.