

Reutilização de Software: Classificação e Seleção de Artefatos Reutilizáveis

MAC5701

Tópicos em Ciência da Computação

Orientadora: Ana Cristina V. de Melo

Claudia de Oliveira Melo

claudia@ime.usp.br

25/06/2004

*“A indústria de software não é industrializada”
(P. Naur and B. Randell, 1968)*

Índice

1.	Introdução	4
2.	Reutilização	5
2.1.	Definições.....	5
2.2.	Objetivos	5
2.3.	Pré-Condições para Reutilização	7
2.4.	Abordagens de Reutilização	8
2.5.	Suporte à Reutilização	9
3.	Métodos de Classificação dos Artefatos	11
3.1.	Baseados em Palavras-Chave	11
3.1.1.	Reuso com Bibliotecas de Documentos de Propósito Geral	12
3.1.2.	Reuso com bibliotecas feitas sob medida	13
3.1.3.	Reuso com acesso guiado por menu	14
3.1.4.	Outros tipos de reuso	14
3.2.	Baseados em Facetas	15
3.3.	Baseados em Orientação a Objetos.....	17
3.3.1.	Relacionamentos de artefatos em redes semânticas.....	18
3.3.2.	Taxonomia de Componentes com Categorias	19
3.4.	Desempenho dos Métodos de Classificação	20
4.	Trabalhos Recentes.....	20
4.1.	Abordagem Semântica para Recuperação de Componentes	20
4.2.	Abordagem Semântica com Web Services.....	21
5.	Comparação entre os Métodos de Classificação	22
6.	Conclusão.....	23
7.	Bibliografia	24

1. Introdução

A produtividade de software aumentou nas últimas décadas, mas não o suficiente para acompanhar a exigente demanda da indústria. Esse fato, comumente chamado de "crise de software", continua a existir, mesmo após muitos estudos na área de Engenharia de Software e Inteligência Artificial.

A reutilização apareceu como uma das poucas abordagens reais para tratar o problema, aumentando a produtividade e a qualidade que a indústria necessitava. Vários avanços foram conseguidos em bibliotecas de software, técnicas de classificação, criação e distribuição de artefatos reutilizáveis, ambientes de apoio à reutilização, entre outros. Mesmo assim o grau de reutilização manteve-se muito aquém das expectativas.

O objetivo desse trabalho é descrever sucintamente a teoria da Reutilização de Software e apresentar as pesquisas correlatas encontradas. Analisando suas vantagens, desvantagens e limitações será possível definir qual a melhor maneira de contribuir para o avanço da tecnologia para Reutilização.

2. Reutilização

2.1. Definições

Para evitar problemas com as diversas e semelhantes nomenclaturas à cerca de reutilização, reuso, reusabilidade etc é interessante definir os seguintes termos[9]:

- ✓ *Reuso* é o uso de conceitos ou produtos previamente adquiridos ou construídos em uma nova situação. Isso envolve a representação desses produtos em vários níveis de abstração, o armazenamento dos mesmos para futuras referências, a identificação de similaridades entre situações novas e antigas, a recuperação de produtos já desenvolvidos (ou parte deles) e sua adaptação na nova situação.
- ✓ *Reusabilidade* é uma medida da facilidade em se utilizar os conceitos e produtos em novas situações.

2.2. Objetivos

A noção de reutilização de software é antiga e consiste em utilizar software existente para construir novos sistemas. Reutilização não é só aplicável a fragmentos de código fonte, mas a todo o trabalho gerado durante o processo de desenvolvimento de software, como dados, arquitetura e projeto. Portanto, a informação susceptível à reutilização inclui a análise de requisitos, especificações do sistema, estruturas de desenho, e qualquer informação que seja necessária ao processo de desenvolvimento. Estes produtos do desenvolvimento são chamados *artefatos de software*.

McIlroy propôs a criação de uma indústria de componentes normalizados com os quais pretendia construir aplicações complexas através de pequenos blocos disponíveis em catálogo. Esperava-se que os programadores reutilizassem tudo o que estivesse disponível, desde pedaços de código a experiências anteriores, mas a reutilização não fazia explicitamente parte do processo de desenvolvimento. Com a criação de aplicações de maiores dimensões, a partir do início da década de 80, a reutilização de software sofreu um impulso decisivo.

Os principais objetivos da reutilização são claros e bem definidos, além de aceitos pela maioria dos autores. Dentre eles, podem ser destacados:

- Aumento da Produtividade: reutilizando-se artefatos o esforço no desenvolvimento diminui gerando um produto final de igual ou superior qualidade do que um gerado sem reutilização
- Aumento da Qualidade: como os artefatos foram previamente testados, a probabilidade de sua correteude é ainda maior. Se a parte tem qualidade, o todo também terá.
- Redução dos custos: o custo de construção de artefatos reutilizáveis é diluído entre os vários projetos onde será reutilizado.
- Redução no tempo de entrega: se o esforço no desenvolvimento diminui, consequentemente o tempo de entrega também diminui.
- Padronização: como os artefatos seguem uma padronização pré-definida, sua reutilização em um sistema causará consequente padronização.
- Interoperabilidade: a padronização garante que os sistemas se comportem de maneira comum, aumentando a interoperabilidade.
- Previsibilidade/Confiabilidade/Redução de Riscos: artefatos bem testados e reutilizados várias vezes têm alto grau de confiabilidade, diminuindo os riscos de erro.

Como toda a tecnologia nova, além dos vários benefícios, a reutilização apresenta também problemas que precisam ser ajustados de forma que possa ser realmente adotada e amplamente utilizada.

2.3. Pré-Condições para Reutilização

O desenvolvimento de aplicações com base em artefatos reutilizáveis deve observar algumas características que podem constituir tanto o sucesso, quanto uma barreira à reutilização:

- 1) **Empenho:** a principal razão da baixa taxa de reutilização que se verifica deve-se à inexistência de qualquer esforço para reutilizar. A indiferença em reutilizar tem origem em razões como a falta de incentivo, limitações temporais, considerar os benefícios da reutilização duvidosos, falta de educação, falta de recursos técnicos, falta de suporte na organização ou falta de apoio por parte da gestão, entre outros.
- 2) **Existência:** o fato de não existirem partes isoladas, sob a forma de artefatos, quer tenham sido desenhados para reutilização ou não, representa o principal impedimento. A falta de incentivo econômico para produzi-lo ou o fato de exigir uma tecnologia mais recente para a sua produção também são razões.
- 3) **Disponibilidade:** a falta de repositórios e a limitação do seu uso são as principais razões da falta de disponibilidade do artefato. A falta de acesso ao código fonte ou a dificuldade em analisar em profundidade o artefato são também razões que conduzem a que o artefato não seja considerado para reutilização.
- 4) **Acessibilidade:** a má ou insuficiente representação e classificação do artefato ou a fraca qualidade das ferramentas de busca são as principais razões para não encontrá-lo. A dificuldade ou incapacidade para especificar o que procurar pode também fazer com que o artefato desejado nunca seja encontrado.
- 5) **Legibilidade:** a dificuldade em compreender o artefato, seja por documentação insuficiente ou devido a uma excessiva complexidade do mesmo, estão na origem da sua desconsideração para reutilização.

- 6) Validação: a falta de teste do artefato, o baixo desempenho, a não adesão aos padrões de normalização ou a falta de suporte por parte do produtor do artefato são consideradas as principais razões para sua rejeição na validação.
- 7) Integrabilidade: a existência de incompatibilidade de hardware e ambiente de integração são as razões mais frequentes que impedem a integração do artefato. A necessidade de modificações muito extensas e trabalhosas, bem como a dependência de software exterior pode também conduzir à eliminação do artefato no desenvolvimento de determinada aplicação.

2.4. Abordagens de Reutilização

Se considerarmos o que precisa ser utilizado e em que fase do desenvolvimento ele deve ser reutilizado, podemos identificar duas abordagens principais de Reutilização:

- i. Geradores de Aplicação: a partir de uma descrição/especificação de alto nível o programa é gerado. Trabalham dentro de apenas um domínio de aplicação. O processo de reutilização é transparente ao usuário.
- ii. Reutilização por Composição: a partir da seleção de um componente é possível construir ou compôr um programa. Isso implica que o usuário é parte ativa do processo de reutilização.

Dentro do conceito de Orientação a Objetos, a Reutilização por Composição possibilita um tipo de reuso mais geral e menos acoplado ao domínio da informação. Optando por essa linha de reutilização, existem alguns requisitos mínimos que os componentes devem satisfazer: i) serem bem definidos, ii) terem interface simples e clara, iii) serem bem documentados.

Esses requisitos são importantes uma vez que o mecanismo de reutilização depende da interação do reutilizador, que deverá entender, avaliar e incorporar o componente em seu software sem um alto custo.

Existem outras questões técnicas fundamentais para a viabilização do reuso: como construir um ambiente que auxilie o processo de reuso? Como organizar os componentes em uma biblioteca? Como torná-los realmente acessíveis ao reutilizador de software? Nesse momento torna-se clara a necessidade de um mecanismo de apoio à reutilização, assunto discutido na próxima seção.

2.5. Suporte à Reutilização

De acordo com um relatório Departamento de Defesa Americano (DoD/SRI) sobre Reutilização de Software, as funções e atributos mais importantes de uma biblioteca de reutilização são quase que inteiramente determinadas pela escolha cuidadosa de algumas opções do repositório, tais como:

1. Plataforma de Representação que pode incluir: sistemas baseados em papel, sistemas de gerenciamento de bases de dados, sistemas de armazenamento e busca de informações, sistemas baseados em conhecimento e hipertexto.
2. Métodos de Indexação e Classificação, dos quais os mais utilizados na prática são aqueles estudados pela comunidade científica biblioteconômica, como:
 - ✓ **Classificação baseada em palavras-chave**, onde os artefatos de software são indexados por palavras-chave extraídas automaticamente de textos dos próprios artefatos;
 - ✓ **Classificação por facetas**, onde os artefatos são categorizados pela síntese de valores das facetas;
 - ✓ **Classificação por Enumeração ou Classificação Orientada a Objetos**, onde os artefatos reutilizáveis são atribuídos a classes hierárquicas mutuamente exclusivas

Outros importantes métodos de indexação de componentes reutilizáveis são aqueles extraídos de:

- ✓ Inteligência Artificial, particularmente de linguística computacional, sistemas baseados em conhecimento e sistemas especialistas;
- ✓ Hipertexto
- ✓ Métodos de Especificação Formal

3. **Escalabilidade do Repositório** entre plataformas de diferentes tamanhos e complexidade, o que é de grande importância na incorporação de políticas de reuso em grandes empresas de desenvolvimento, como a NASA por exemplo.

As questões consideradas como parte da Escalabilidade de Reuso são: interoperabilidade das bibliotecas, projeto de interface das bibliotecas, bases de dados distribuídas, segurança da base de dados, garantia de qualidade, gerenciamento de mudança, suporte automático à indexação por vocabulário controlado, enfim, melhores representações de coleções de bibliotecas para ajudar os usuários a achar e entender as partes que precisam.

Nas próximas seções serão resumidos os principais métodos e técnicas desenvolvidos recentemente, que foram adotados pelos grupos de pesquisa e desenvolvimento para a construção e gerenciamento de bibliotecas de software. Será dada ênfase para os **métodos de indexação e classificação** dos artefatos reutilizáveis.

3. Métodos de Classificação dos Artefatos

Nessa seção serão apresentados os principais tipos de classificação existentes, citando seus exemplos clássicos de aplicação. Trabalhos mais recentes na área serão considerados na próxima seção.

3.1. Baseados em Palavras-Chave

As antigas bibliotecas de software forneciam pouca automatização para acessar e recuperar suas funções, módulos e estruturas de dados. Em bibliotecas de software matemáticas como IMSL [1], os fornecedores de biblioteca entregavam uma sólida documentação, com descrição detalhada de cada componente reutilizável, um sumário e um índice analítico. Anos atrás, por exemplo no Unix ou no Microsoft Foundation Classes (MFC), a documentação era acessível na forma de manuais online com busca. Somente nos últimos anos que as bibliotecas incorporaram métodos sofisticados de classificação, baseados no processo de identificar e aglomerar strings de palavras-chave encontradas no texto do próprio artefato de software.

Salton provê um levantamento completo de técnicas baseadas em palavras-chave aplicadas a recuperação de texto [2]. Em sua revisão ele sugere uso de:

- ✓ índices invertidos para melhor acesso aos registros do texto;
- ✓ restrições de distância para avaliar mais precisamente a proximidade de dois registros;
- ✓ pesos e frequências para distingüir a importância de palavras-chave;
- ✓ listas de parada para eliminar palavras sem grande importância, mas que geralmente são utilizadas (preposições, por exemplo).
- ✓ listas de sinônimos e enciclopédias para ampliar as consultas de recuperação de texto;
- ✓ similarização de palavras (word stemming) a truncamento de termos para extrair as raízes e padronizar o uso de palavras;
- ✓ Procura por nível de quorum para controlar o tamanho da saída da recuperação;
- ✓ técnicas de procura em listas parciais para considerar subconjuntos de termos da pesquisa;

- ✓ técnicas de formação de frases para controlar o contexto de co-ocorrência de palavras-chave.
- ✓ indexação estatística de texto para automatizar a classificação de documentos;
- ✓ aglomeração (*clustering*) de documentos para melhorar a recuperação de documentos relacionados;

Em um trabalho posterior de Salton [3], ele também investigou várias abordagens lingüísticas para indexação de documentos, como por exemplo a geração de identificadores de conteúdo complexo, uso de termos semânticos obtidos de dicionários lidos por máquina, e a utilização de bases de conhecimento especialmente construídas para a classificação.

Ele eventualmente destaca a imprecisão inaceitável dos métodos sintáticos utilizados em seus experimentos e indica o poder de métodos estatísticos mais simples. Quase todas as técnicas aplicáveis na recuperação de texto foram tentadas e usadas com sucesso variável na classificação e recuperação de artefatos de software textuais. Muitos dos métodos identificados também foram estendidos ou combinados com outros tipos de métodos de classificação do software, como Facetas ou Enumerativa (seções seguintes).

3.1.1. Reuso com Bibliotecas de Documentos de Propósito Geral

Como a maioria dos artefatos de software são textuais, a abordagem mais óbvia para a classificação e busca de software está na adoção de sistemas de processamento de texto pré-existent. Frakes e Nejmeb empregaram o sistema baseado em palavras-chave denominado CATALOG (AT&T) para criar uma biblioteca pequena dos módulos de software [49].

O sistema de CATALOG contém um gerador de base de dados, uma ferramenta interativa para criar, modificar, adicionar, e remover registros, além de uma interface de busca. Essa interface permite combinações booleanas dos termos e conjuntos de registros recuperados, e as perguntas são resolvidas com técnicas de casamento parcial. A busca é realizada usando um índice invertido de termos significativos e uma lista de palavras que devem ser descartadas (lista de parada).

Para promover clareza e a exatidão da informação reusável, Frakes e Nejmeah a estruturaram usando moldes predefinidos (*templates*) que tratavam diferentes tipos de artefatos de software, como por exemplo: módulos e funções. A aproximação feita neste sistema experimental mostrou que a tecnologia de recuperação de informação (IR) padrão (não construída especificamente com o propósito de classificar e buscar artefatos de software) poderia ser usada eficazmente na organização de bibliotecas simples.

Mostrou também que a organização de grande repositórios necessitam de tecnologias especializadas que vão além de IR. Ela poderia ajudar na interpretação, indexação e estruturação dos artefatos de software, mas necessitaria de técnicas de processando de informação tais como processamento de linguagem natural (NLP), representação do conhecimento e uso de enciclopédias inteligentes.

3.1.2. Reuso com bibliotecas feitas sob medida

Um sistema especializado em reuso de software, mais conhecido pelo seu mecanismo de recuperação baseado em palavras-chave, é o STARS (Software Technology for Adaptable, Reliable Systems), do Departamento de Defesa Americano (DoD).

A experiência com a construção e o uso do sistema conduziu os desenvolvedores às seguintes questões sobre a adoção de padrões de recuperação de texto em reutilização de software:

- ✓ similarização de palavras às vezes reduz palavras não correlatas;
- ✓ algumas palavras aparentemente de ruído são de fato palavras que têm um conteúdo em alguns contextos;
- ✓ o uso automático de sinônimos às vezes enlarga o espectro da pergunta de maneira imprópria;
- ✓ determinadas combinações de palavras devem ser tratadas como frases ao invés de palavras de busca;
- ✓ a expansão de perguntas sem acertos nem sempre é útil.

Em vista destes problemas, o DoD desenvolveu e incorporou ao sistema as seguintes facilidades :

- i. Similarização de palavras mais exata baseada no uso pretendido de palavras;
- ii. Controle maior sobre o uso de palavras ruidosas, sinônimos, e de expansão da pergunta;
- iii. Casamento de frases no refinamento de perguntas;
- iv. Ajuda no processo de reformulação da pergunta;
- v. Maior poder no tratamento de expressões técnicas.

3.1.3. Reuso com acesso guiado por menu

O sistema REUSE (REUsing Software Efficiently) foi construído para classificar e recuperar eficazmente a informação existente do software [5], isto é moldes, módulos, pacotes e programas executáveis. Similarmente ao DoD, ele usa o esquema de palavras-chave para classificar seus componentes da biblioteca. Ao mesmo tempo, entretanto, o acesso de usuário ao sistema é organizado não por meio das perguntas, mas através de uma interface customizável, guiada por menu.

O software usa as palavras-chave, que refletem as necessidades de uma organização, para construir um sistema hierárquico de menus que refletem os padrões e as metodologias organizacionais. Tais menus e palavras-chave fornecem ferramentas para classificar e recuperar componentes de software reutilizáveis. O sistema REUSE mantém também uma enciclopédia para reduzir diferenças de terminologia dentro da comunidade de usuários.

3.1.4. Outros tipos de reuso

Sistemas mais recentes estendem a recuperação fundamental por palavras-chave, incluindo facilidades que realçam o acesso e a manutenção dos repositórios de reutilização. No sistema SIB (Software Information Base), as palavras-chave que caracterizam os artefatos reutilizáveis são organizados em descritores da palavra-chave. São determinados pesos para determinar sua importância relativa e então relacioná-los pela similaridade semântica. Os descritores e seus relacionamentos

formam uma rede elaborada des nós e ligações que podem mais tarde ser buscadas pela facilidade da busca do sistema [4].

No sistema CART (Computer-Aided Reuse Tool), a classificação é executada usando palavras-chave geradas automaticamente de modelos de especificação. Os usuários podem então buscar no sistema usando uma língua muito mais rica do que aquela usada no modelo. Isso é possível com o auxílio de uma enciclopédia, normalização da terminologia, e alguma ajuda do usuário sempre que a resolução for muito complicada para o sistema [7].

No sistema CodeFinder, o sistema de recuperação baseado em palavras-chave é sustentado pela reformulação interativa das perguntas e por uma aproximação da recuperação usando uma rede neural, com um algoritmo capaz de recuperar eficazmente os componentes de software relacionados à pergunta, seja feita por palavras-chave, frases ou afinidades lexicais [6].

3.2. Baseados em Facetas

A classificação baseada em facetas foi proposta originalmente por um aluno e uma bibliotecária, Shiyali Ramamrita Ranganathan, como uma técnica eficaz para a gerência de informação da biblioteca [8]. É a principal concorrente do popular sistema decimal de Dewey para classificação de coleções de biblioteca. Prieto-Diaz e Freeman foram os primeiros a sugerir a possibilidade de adotar o esquema facetas na classificação e recuperação dos artefatos reutilizáveis de software [9, 10].

Tradicionalmente, a técnica de classificação por facetas confia na existência de um grande número de termos do domínio organizados em diversos conjuntos distintos, mutuamente exclusivos e ortogonais nomeados **facetas**.

Uma faceta pode então ser definida como uma coleção homogênea de termos de domínio, na qual pode ser incluída todos os possíveis valores de atributo. É importante notar que uma faceta não é nada mais que um par atributo-valor, onde temos um atributo fixado pela faceta e vários valores associados ao atributo. O interessante

desse esquema é que as facetas podem variar de número, tamanho e natureza, o que permite sua vasta utilização e combinação com outras tecnologias de classificação.

Cada artefato adquirido é descrito nos termos de um vetor de descrição, onde cada valor do vetor é escolhido de uma das facetas predefinidas. Tais vetores de classificação são armazenados então em uma tabela relacional, onde as colunas representam facetas e as linhas denotam descrições dos artefatos. A tabela de classificação pode mais tarde ser procurada para encontrar e recuperar as descrições necessárias do artefato. Procurá-los seria simples e eficaz utilizando uma linguagem de pergunta relacional, tal como o SQL.

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 (*clustering*) 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. Outros mecanismos de busca podem utilizar métodos mais sofisticados, baseados nos perfis estatísticos das descrições do artefato, métricas de afinidade, espaços vetoriais ou lógica fuzzy [10, 2].

Sua principal vantagem é a facilidade da classificação do artefato, a simplicidade da representação e armazenamento das descrições, da uniformidade de atributos da classificação e da facilidade da automatização [19]. A principal deficiência é o custo elevado da manutenção do repositório, pois a classificação é essencialmente manual [12,13].

Os seguintes aspectos são característicos para muitos sistemas de classificação baseados em facetas:

- a) as facetas especificam um vocabulário controlado usado na classificação do artefato;
- b) as facetas são geralmente combinadas para fazer a classificação de assunto específico;
- c) os termos de classificação são geralmente ampliados com o uso de listas de sinônimos e uma enciclopédia;

- d) as facetas são ordenadas por sua relevância aos usuários da coleção;
- e) a similaridade de termos da faceta é avaliada arranjando-as em um grafo conceitual de distância;
- f) os artigos recuperados são classificados por sua proximidade semântica à pergunta;
- g) a classificação baseada em facetas é extensível, porque os termos novos da classificação e as novas facetas podem ser adicionadas a qualquer momento.

3.3. Baseados em Orientação a Objetos

Uma maneira completamente diferente de organizar artefatos de software pode ser conseguida com o uso da classificação enumerativa [9], que acredita que artefatos podem ser incluídos em uma hierarquia predefinida de categorias. Uma das aplicações mais bem sucedida deste método é classificação de títulos de livro por assunto via método decimal de Dewey. Uma outra forma de classificação hierárquica é representada pelos sistemas de conceitos orientados a objetos [14, 15, 16].

Orientação a Objetos consiste tipicamente em muitos conceitos relacionados, geralmente referenciados como objetos. Eles são descritos nos termos de seus valores de propriedade que podem incluir referências a outros objetos. Os objetos podem também definir métodos, que produzem computações úteis no processamento de pedidos recebidos de outros objetos do sistema ou de interfaces do sistema. Os objetos novos com todas as suas propriedades e métodos são criados (instanciados) de acordo com as descrições abstratas fornecidas por classes. As classes são organizadas em hierarquias de herança. Como resultado, as classes mais específicas herdam as propriedades e o comportamento de suas superclasses.

Diferentes abordagens de classificação podem ter de ser feitas para os diferentes tipos de objetos no sistema. Nós podemos desejar construir uma taxonomia diferente para especificações e projetos da classe, alternativas de projeto, peculiaridades arquiteturais ou detalhes de implementação da classe [20].

Os defensores de orientação a objetos [21] pregam que as estruturas e os mecanismos empregados em sistemas de software orientados a objetos facilitam extremamente a reutilização de componentes de software, pois:

- ✓ A programação orientada a objetos promove o desenvolvimento *bottom-up* usando classes pré-existentes;
- ✓ Um sistema orientado a objetos pode ser visto como uma coleção de classes inter-relacionadas e com comunicação, o que é melhor que um bloco monolítico de programação;
- ✓ As classes combinam dados e procedimentos, são modulares e, em geral, suas relações são claras e abstratas;
- ✓ As classes novas podem ser derivadas das classes existentes a partir de especialização e instanciamento;
- ✓ As características e o comportamento de classes existentes estão disponíveis nas classes derivadas através do mecanismo de herança e herança múltipla;
- ✓ É possível desenvolver especialmente para reuso, pois podem incluir um número de características "adiadas" (por exemplo: métodos virtuais), que serão implementadas na especialização da classe;
- ✓ As classes combinam elementos do projeto e implementação em uma única unidade reutilizável.

3.3.1. Relacionamentos de artefatos em redes semânticas

As redes semânticas são os sistemas mais recentes para descrição estruturada de objeto [22] - uma coleção de conceitos e valores-propriedade, formando uma rede dos nós relacionados por suas propriedades.

Embora tais sistemas não sejam projetados para serem repositórios de componentes de software, muitas técnicas desenvolvidas inicialmente para a

manipulação de conhecimento em redes semânticas são aplicáveis no processamento de artefatos e suas descrições. As redes semânticas são estruturadas em planos de nós relacionados a um único conceito. Cada plano, entretanto, não tem nenhuma estrutura interna, uma vez que um conceito semântico é representado como por uma rede de todos os nós alcançáveis naquele plano. Um conceito pode também estar relacionado com os conceitos de outros planos. Tais ligações representam inferências e deslocamentos de atenção.

Na rede semântica não há nenhuma hierarquia predeterminada de classes e superclasses: cada nó define sua própria hierarquia de conceitos alcançáveis do seu ponto na rede.

3.3.2. Taxonomia de Componentes com Categorias

A organização taxonômica de artefatos de software é muito popular em sistemas que armazenam componentes OO reutilizáveis. Muitos deles foram parcialmente classificados na fase de projeto e implementação. Devanbu, Brachman, Selfridge e Ballard [17] descrevem o sistema de informação do software **LaSSIE** que integra visões arquiteturais, conceituais e de código de um grande sistema.

As facilidades de pesquisa e navegação do sistema permitem que os programadores procurem em um grande sistema de software possíveis componentes reutilizáveis. A base de conhecimento de LaSSIE armazena os componentes de software em uma estrutura taxonômica que representam ações (por exemplo ações externas e internas), objetos (por exemplo: dispositivo, recurso, hardware e software), atores (por exemplo: usuários, processos, grupos e processadores), e dos estados (por exemplo: da linha, do recurso, dos dados e da rede).

Sistemas baseados em conhecimento como LaSSIE representam e classificam componentes de software na estrutura com lógica. Isso permite:

- ✓ Agregação da informação sobre programas e seus componentes;
- ✓ Recuperação semântica de componentes de software;
- ✓ Uso da classificação e da herança para dar suporte aos updates do software;

- ✓ Uso de uma base de conhecimento como índice inteligente de artefatos do software.

3.4. Desempenho dos Métodos de Classificação

Experimentos conduzidos por Frakes e Pole em 1994 mostraram que não há uma vantagem clara de um método sobre outro, se considerarmos a eficácia de cobertura (*recall*) e a precisão na busca do artefato. Porém, o método de Enumeração superou os outros métodos quanto à velocidade de busca. Também foi determinado que a classificação por palavras-chave é menos cara, pois não requer intervenção humana no processo de indexação de documentos.

4. Trabalhos Recentes

Os trabalhos recentes na área de classificação são, em geral, combinações dos métodos clássicos com tecnologias novas nas áreas de Recuperação de Informação, Processamento de Linguagem Natural, Redes Semânticas e Grafos Conceituais.

4.1. Abordagem Semântica para Recuperação de Componentes

Em 2003, Sugumaram e Storey [24] apresentaram uma solução baseada em semântica para a recuperação de componentes. Essa abordagem utiliza uma ontologia do domínio para prover a semântica necessária ao refinamento de pesquisas feitas pelos usuários em linguagem natural, para assim compará-las aos componentes do repositório.

Os três principais passos dessa abordagem são:

1. Geração da Pesquisa Inicial a partir da pergunta digitada em Linguagem Natural: através de processamento de linguagem natural, as palavras-chave são abstraídas. Esse módulo do sistema foi inspirado no trabalho de Girardi e Ibrahim [26];

2. Refinamento da Pesquisa: mapeamento da pergunta inicial pela ontologia do domínio, garantindo o uso dos termos corretos na pesquisa;
3. Recuperação dos Componentes: uma medida de distância conceitual é empregada para trazer os componentes mais relevantes encontrados.

4.2. Abordagem Semântica com Web Services

Em 2004, Yao e Etzkorn [25] estenderam a idéia de Sugumaram e Storey, inovando em sua abordagem semântica nos seguintes pontos:

1. Extensão do Repositório de Artefatos para o *World Wide Web*, ou seja, a Internet agora é vista como um repositório de artefatos de software;
2. Os componentes de software são tratados como serviços descritos por um formato de representação semântica de serviço;
3. Melhoria na Recuperação de Informação através do casamento semântico entre a representação semântica da pergunta do usuário contra uma ontologia do domínio;
4. Armazenamento dos componentes relevantes em um repositório baseado em infra-estrutura UDDI (Universal Description, Discovery and Integration).

O interessante nesse trabalho é a utilização de linguagens e protocolos recentemente desenvolvidos na área de Web Services. Conceitualmente, os web services não são muito diferentes dos outros modelos de componentes que encapsulam regras de negócio, tais como Java Beans, COM ou CORBA. Os web services têm:

- ✓ Um protocolo de transporte bem definido, baseado em XML - SOAP (Simple Object Access Protocol);

- ✓ Uma linguagem para definir as interfaces entre os web services - WSDL (Web Services Definition Language);
- ✓ Um padrão para se criar serviços de diretório - UDDI (Universal Description, Discovery and Integration)

Portanto, Web Services é uma tecnologia que apóia a reutilização de software, uma vez que os componentes são encarados como serviços e os clientes querem encontrá-los e contratá-los. Toda essa arquitetura serve para ajudar o processo de classificação e publicação de serviços em uma rede. Essa rede será organizada e acessada como uma rede semântica.

5. Comparação entre os Métodos de Classificação

Nesta seção é mostrada uma tabela comparativa dos métodos de classificação de artefatos vistos anteriormente. As principais considerações sobre cada método dizem respeito ao pré-processamento, tipo de repositório, esquema de classificação e mecanismo de pesquisa.

Método	Pré-processamento	Repositório	Classificação	Pesquisa
Palavras-chave	Análise Léxica Sufixo Variação Fonética Listas de Parada Formação Frasal Análise Sintática Dicionários com Sinônimos	Índice Invertido Base de dados de texto completo	Indexação Estatística de texto Vocabulário Livre Grupo de Termos (Clusters) Grupo de Documentos	Restrição de distâncias Pesquisa por Lista Parcial
Facetas		Base de Dados Relacional Vetores de Descrição Facetas Grafos de Distâncias Conceituais	Manual Vocabulário Controlado Proximidade Conceitual Fuzzy	Reformulação de pesquisa Menus
Orientado a Objetos		Hierarquia Classe e Superclasse Atributo-valor	Especialização Generalização Instanciação	Taxonomia Transversal Índices Padrões

6. Conclusão

Para aumentar a produtividade no desenvolvimento de software é necessário encontrar alternativas de escrever e reescrever menos código a cada novo projeto. A reutilização de artefatos - produtos gerados durante o desenvolvimento de um projeto ou produto de software - viabiliza essa idéia. Estudar as tecnologias criadas para implementar essa solução (como bibliotecas de software, técnicas de classificação e recuperação de artefatos) e suas limitações é essencial para a implantação de um processo de reutilização.

Este trabalho introduziu brevemente alguns aspectos teóricos relativos à Reutilização de Software, dando ênfase maior às técnicas de Classificação dos artefatos. Por ser muito importante, mas também extensa, a área de Classificação, aspectos sobre como armazenar artefatos após sua classificação serão tratados em um trabalho posterior.

Estudando um pouco mais a classificação de artefatos é possível perceber a complexidade de sua implementação, além dos vários problemas inerentes a esse tipo de atividade. O maior objetivo das pesquisas nesse campo está relacionado com o aumento de precisão na descrição de um artefato, aumentando conseqüentemente as chances do mesmo ser encontrado nas buscas e finalmente reutilizado.

7. Bibliografia

- [1] IMSL (1995): The IMSL Product Family, WWW Document <http://www.vni.com/index.html>, Visual Numerics, Inc: Houston, Texas.
- [2] Salton, G. (1989): Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Readings, Massachusetts: Addison-Wesley Pub.
- [3] Salton, G., C. Buckley, and M. Smith (1990): On the application of syntactic methodologies in automatic text analysis. Information Processing & Management. 26(1): p. 73-92.
- [4] Fugini, M.G. and S. Faustle (1993): Retrieval of reusable components in a development information system, in Advances in Software Reuse: Selected Papers from the Second International Workshop on Software Reusability,
- [5] Arnold, S.P. and S.L. Stepoway (1988): The reuse system: Cataloging and retrieval of reusable software, in Software Reuse: Emerging Technology, W. Tracz, Editor. Computer Society Press: Washington, D.C. p. 138-141.
- [6] Henninger, S. (1994): Using iterative refinement to find reusable software. IEEE Software. 11(5): p. 48-59.
- [7] Liao, H.-C. and F.-J. Wang (1993): Software reuse based on a large object-oriented library. ACM SIGSOFT, Software Engineering Notes. 18(1): p. 74 -80.
- [8] Ranghanathan, S.R. (1957): Prolegomena to Library Classification. Letchworth, Hertfordshire, UK: The Garden City Press Ltd.
- [9] Prieto-Diaz, R. (1989): Classification of reusable modules, in Software Reusability: Concepts and Models, T.J. Biggerstaff and A.J. Perlis, Editors. Addison-Wesley Pub. Co.: New York, NY. p. 99-123.
- [10] Prieto-Diaz, R. and P. Freeman (1987): Classifying software for reusability. IEEE Software. 4(1): p. 6-16.
- [11] Frakes, W.B. and T.P. Pole (1994): An empirical study of representation methods for reusable software components. IEEE Transactions on Software Engineering. 20(8): p. 617-630.
- [12] Mili, H., E. Ah-Ki, R. Godin, and H. Mcheick (1997): Another nail to the coffin of faceted controlled-vocabulary component classification and retrieval. Software Engineering Notes. 22(3): p. 89-98.
- [13] Sorumgard, L.S., G. Sindre, and F. Stokke (1993): Experiences from application of a faceted classification scheme, in Advances in Software Reuse: Selected Papers from the Second International Workshop on Software Reusability, P.-D. Ruben and B.F. William, Editors. IEEE Computer Society Press: Los Alamitos, California. p. 116-124.
- [14] Dillon, T.S. and P.L. Tan (1993): Object-Oriented Conceptual Modeling. Sydney: Prentice-Hall.

- [15] Goldberg, A. (1984): The influence of an object-oriented language on the programming environment, in *Interactive Programming Environments*, D. Barstow, H. Shrobe, and E. Sandewall, Editors. McGraw-Hill. p. 141-174.
- [16] Graham, I. (1994): *Object Oriented Methods*. Wokingham, England: Addison-Wesley Pub. Co.
- [17] Devanbu, P., R.J. Brachman, P.G. Selfridge, and B.W. Ballard (1991): LaSSIE: A knowledge-based software information System. *Communications of ACM*. 34(5): p. 34-49.
- [18] Prieto-Diaz, R. and G.A. Jones (1987): Breathing new life into old software. *GTE Journal of Science and Technology*. Spring: p. 23-31.
- [19] Prieto-Diaz, R. (1991): Implementing faceted classification for software reuse. *Communications of ACM*. 34(5): p. 88-97.
- [20] D'Alessandro, M., P.L. Iachini, and A. Martelli (1993): The generic reusable component: an approach to reuse hierarchical OO designs, in *Advances in Software Reuse: Selected Papers from the Second International Workshop on Software Reusability*, P.-D. Ruben and B.F. William, Editors. IEEE Computer Society Press: Los Alamitos, California. p. 39-46.
- [21]. Meyer, B. (1987): Reusability: the case for object-oriented design. *IEEE Software*: p. 50-64.
- [22] Quillian, M.R. (1968): Semantic memory, in *Semantic Information Processing*, M. Minsky, Editor. MIT Press: Cambridge, MA. p. 227-270.
- [23] Minsky, M. (1975): A framework for representing knowledge, in *The Psychology of Computer Vision*, P. Winston, Editor. McGraw-Hill: New York. p. 211-280.
- [24] Sugumaran, Vijayan; Storey, Veda C. (2003). "A Semantic-Based Approach to Component Retrieval", *The DATA BASE for Advances in Information Systems* – Summer 2003, Vol. 34, N° 3, pg 8-24.
- [25] Yao, Haining; Etkorn Letha (2004). "Towards A Semantic-Based Approach to Reusable Component Classification and Retrieval".
- [26] Girardi, M. R.; Ibrahim, B. (1993). "A Software Reuse System based on Natural Language Specifications", *Proceedings of the 5th International Conference on Computing and Information (ICCI'93)*, Sudbury, MA, May 27-29, 1993, pp. 507-511.