

Recuperação de Componentes em Bibliotecas de Software:

Uma Abordagem Conexionista

Márcio de Oliveira Barros¹
Cláudia Maria Lima Werner¹
Luiz Pereira Caloba²

¹ Depto. de Engenharia de Sistemas e Computação
{marcio, werner}@cos.ufrj.br

² Departamento de Engenharia Elétrica
caloba@coe.ufrj.br

COPPE / UFRJ

Universidade Federal do Rio de Janeiro
Cidade Universitária, RJ, Brasil
21945-970 Caixa Postal 68511

Abstract

This paper presents a reusable component retrieval approach that uses a neural network based on the Kohonen's self-organizing layer and the faceted classification. The network is trained with the objective of correcting similarity relationships between facet terms, enhancing the selection precision in an evolutive way (i.e., based on the results of previous selections). This technique was used on the implementation of the reusable components' search mechanisms of the Memphis environment.

KEY WORDS: software reuse, neural networks, classification.

I. Introdução

A reutilização de software é uma técnica de aproveitamento de informações produzidas durante desenvolvimentos de software anteriores, com o objetivo de reduzir o esforço necessário para o desenvolvimento de um novo projeto [8]. A idéia geral da reutilização é reaproveitar componentes de software previamente desenvolvidos em novos projetos, reduzindo o tempo de

desenvolvimento destes projetos e aumentando sua qualidade. Outros benefícios podem ser observados na manutenção e evolução de um produto de software.

Entretanto, a falta de ambientes de desenvolvimento que ofereçam ferramentas para facilitar a reutilização é um dos principais empecilhos tecnológicos a esta. A dificuldade de se selecionar um componente de uma biblioteca, de entender o seu funcionamento e de modificar este componente podem inibir sua utilização. Outros fatores inibidores são falhas na gerência das bibliotecas de componentes e fatores não-tecnológicos, ligados à psicologia, aspectos econômicos, gerenciais ou legais [7].

A seleção dos componentes a serem reutilizados é a primeira etapa do processo de reutilização. Esta etapa visa encontrar na biblioteca os componentes que atendam a certos requisitos funcionais, necessários em um produto de software em desenvolvimento. Se a biblioteca não possuir um componente que atenda exatamente estes requisitos, o algoritmo de seleção deve retornar componentes que atendam a requisitos similares.

Diversas abordagens já foram exploradas para a seleção de componentes reutilizáveis, entre elas a classificação por palavras-chave [4], a classificação facetada [14, 18, 5], o processamento automático de linguagem natural [10, 6] e a descrição formal do comportamento dos componentes [9]. Todas estas técnicas possuem uma característica em comum: atuam sobre um algoritmo pré-definido, não sendo capazes de evoluir e aprimorar a precisão da busca a medida que são utilizados.

Este artigo apresenta uma técnica de recuperação de componentes reutilizáveis que utiliza uma rede neural baseada na camada auto-organizável de Kohonen [17] e classificação facetada [14]. A rede é treinada com o objetivo de corrigir as relações de similaridade entre termos de facetadas, aprimorando a precisão do processo de seleção evolutivamente, de acordo com o resultado de buscas anteriores [2].

A principal motivação para a utilização de redes neurais em algoritmos de recuperação de componentes reutilizáveis vem da afinidade das redes neurais com problemas de reconhecimento de padrões. A recuperação de componentes reutilizáveis pode ser vista como um problema de reconhecimento de padrões: os algoritmos de recuperação buscam, dentre padrões de

componentes de uma biblioteca, os componentes que mais se assemelham a um padrão descrito pelo usuário.

Outras características de redes neurais que podem ser exploradas na recuperação de componentes reutilizáveis são sua capacidade de generalização e de aprendizado. A capacidade de generalização permite a seleção de componentes com pequenas diferenças do padrão especificado pelo usuário, caso não seja encontrado um componente idêntico ao desejado. A capacidade de aprendizado permite que a rede modifique as informações de busca de acordo com o comportamento do usuário, que indica se uma busca executada pela rede foi bem sucedida ou não.

Inicialmente, o artigo descreve a abordagem para a aplicação de redes neurais sobre o esquema de classificação facetada. Em seguida, uma visão geométrica do esquema de classificação é apresentada. O artigo discute, ainda, o método de treinamento da rede e os experimentos realizados com o modelo. Por fim, são traçadas considerações sobre a técnica descrita neste trabalho.

II. O Algoritmo de Recuperação

Redes neurais representam uma técnica do paradigma conexionista da Inteligência Artificial, baseada na simulação do cérebro humano através de estruturas computacionais. Elas são capazes de aprender, alterando seu comportamento de acordo com modificações nas informações, de generalizar e de trabalhar como memória associativa.

Redes neurais se baseiam no conceito de neurônio artificial, que possui um conjunto de sinapses de entrada e um único valor de saída. Cada sinapse possui um peso e recebe um valor de entrada. A saída do neurônio é calculada a partir do somatório dos produtos dos pesos das sinapses por seus valores de entrada. A organização dos neurônios define um modelo de rede neural.

O objetivo de se aplicar uma rede neural em um algoritmo de recuperação de componentes reutilizáveis é permitir que este algoritmo seja evolutivo, corrigindo seu comportamento de acordo com as buscas anteriores. A recuperação de componentes reutilizáveis está diretamente relacionada com a forma com que estes componentes são representados. Em nossa abordagem, os

componentes são classificados por um esquema de classificação facetada [14] e a rede neural é utilizada para corrigir o grafo conceitual de relações de similaridade entre os termos de facetas. Entre os diversos esquemas de classificação encontrados na literatura, selecionamos a classificação facetada por sua grande aceitação tanto na comunidade acadêmica quanto no meio industrial. O modelo de rede neural utilizado é apresentado na Figura 1.

Este modelo se baseia no mapa auto-organizável de Kohonen - camada de Kohonen - que é uma rede neural com apenas uma coluna de neurônios. Esta rede funciona no modelo "winner-takes-all", onde apenas um neurônio é ativado¹ para um conjunto de valores de entrada. Este modelo se aplica a problemas de classificação, onde um elemento, apresentado

na camada de entrada da rede, é inserido em uma classe, dentre um conjunto fixo de classes, representadas pelos neurônios.

A descrição do elemento que será classificado é apresentada na camada de entrada da rede, enquanto cada neurônio representa uma das classes em que o elemento pode ser classificado.

As sinapses são ligações entre a camada de entrada da rede neural e os neurônios. As sinapses contêm a representação das classes, podendo ser vistas como uma "tradução" destas para a representação utilizada na descrição dos componentes. O número de neurônios está diretamente associado ao número de classes existentes: a rede possui um neurônio para cada classe. O número de sinapses na camada de entrada depende da forma na qual são descritos os elementos que serão classificados.

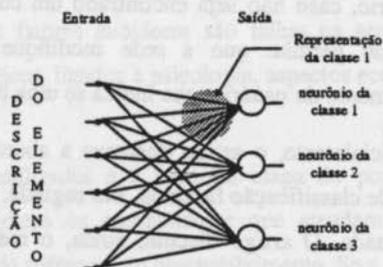


Figura 1 - A estrutura da rede neural

¹ O neurônio ativado tem seu valor de saída alterado para 1, enquanto os demais neurônios de sua coluna tem seus valores alterados para zero.

A rede neural indica a classe a que pertence um elemento apresentado em sua camada de entrada ativando um de seus neurônios. Na camada de Kohonen padrão, apenas um neurônio pode ser ativado para um elemento apresentado na camada de entrada da rede. Uma variante desta rede padrão pode ser construída, permitindo que mais de um neurônio seja ativado. Neste caso, a rede estaria classificando o elemento apresentado em sua camada de entrada em mais de uma classe. Esta modificação no modelo padrão nos permite a determinação não apenas do componente reutilizável mais próximo de uma descrição do usuário, mas também componentes similares a este.

Geometricamente, a camada de Kohonen representa cada classe como um vetor, que é disposto nas sinapses entre a camada de entrada e o neurônio que representa a classe. Pelo modelo do neurônio artificial, quando um vetor é apresentado na camada de entrada, cada neurônio terá como saída o somatório dos produtos de cada posição do vetor de entrada pela respectiva posição de seu vetor de sinapses. Assim, cada neurônio da camada de saída calculará o produto vetorial do vetor de entrada por seu vetor de sinapses. O produto vetorial pode ser equacionado pela seguinte fórmula:

$$\| u \cdot v \| = \| u \| \cdot \| v \| \cdot \cos \alpha$$

onde,

u e v são os vetores de entrada e de sinapses, respectivamente;

$\| u \cdot v \|$ é o produto vetorial dos vetores u e v ;

$\| u \|$ e $\| v \|$ são as normas dos vetores u e v , e

α é o ângulo entre os vetores u e v .

Se os vetores estiverem normalizados para o módulo unitário, cada neurônio calculará o cosseno entre seu vetor de sinapses e o vetor de entrada. Quanto maior o cosseno entre estes vetores, mais próximos eles estarão entre si. Assim, a classe com maior proximidade do vetor de entrada será aquela cujo neurônio possuir o maior valor de saída, ou seja, a classe representada pelo neurônio ativado pela rede.

Em nossa abordagem, a descrição do componente desejado é apresentada na camada de entrada da rede e cada componente reutilizável da biblioteca é representado por um neurônio. Neste caso, as classes em que a rede neural poderá classificar o componente desejado pelo usuário serão os próprios componentes da biblioteca. Assim, a rede neural indicará qual o componente da biblioteca está mais próximo da descrição do usuário.

Como os componentes da biblioteca são as classes da camada de Kohonen, o número de neurônios da rede será igual ao número de componentes reutilizáveis. Cada componente é representado pelo conjunto de sinapses ligando a camada de entrada ao neurônio que o representa. Para que esta rede possa ser construída, um problema deve ainda ser resolvido: a descrição dos componentes reutilizáveis, que define o tamanho da camada de entrada da rede. Este é o assunto da próxima seção.

III. Uma Visão Geométrica da Classificação Facetada

A classificação facetada, proposta por Prieto-Diaz [14], é utilizada para descrever os componentes reutilizáveis e o componente desejado pelo usuário. Este esquema de classificação considera que os elementos de uma área de conhecimento possuem características similares entre si, que os identificam neste contexto. Cada uma destas características é chamada de faceta, sendo os elementos descritos por um conjunto de facetar. Algumas vezes, as facetar são consideradas projeções ou pontos de vista de um domínio específico. Cada faceta possui um dicionário com os possíveis valores que pode assumir. Estes valores são chamados de termos de facetar. Desta forma, os componentes reutilizáveis e o componente desejado são descritos por um conjunto de termos de facetar.

No contexto de nossa abordagem, a descrição de termos de facetar deve ser traduzida para a forma de um vetor que possa ser utilizado na rede neural. O problema se resume a construir um vetor que represente um componente reutilizável, dados os termos de facetar que o classificam. Este vetor é utilizado para a programação da matriz de sinapses da rede neural, representando uma das classes em que a descrição do usuário pode ser classificada. O mesmo processo utilizado

para construção deste vetor deve ser utilizado para a construção de um vetor para o componente desejado pelo usuário, que também é descrito por um conjunto de termos de facetas.

Devemos ainda definir um espaço de representação dos vetores dos componentes reutilizáveis, considerando um conjunto com f facetas, cada qual com um determinado número t_x de termos. Cada termo de faceta t_x define uma dimensão neste espaço, ortogonal às dimensões definidas pelos outros termos. O número de dimensões deste espaço é igual ao número total de termos de facetas do esquema de classificação. A existência de uma relação entre dois termos t_1 e t_2 indica que, se um componente possui o termo t_1 em sua descrição, o vetor que o representa terá uma projeção no espaço definido pelo termo t_2 . Esta projeção está diretamente relacionada com o valor da relação de similaridade entre os dois termos.

Supondo uma biblioteca com X componentes reutilizáveis, teremos um espaço com X vetores de componentes. Quando um reutilizador estiver procurando por um componente, a descrição do componente desejado será criada como um novo vetor neste espaço. A rede neural calculará o produto vetorial do vetor descrição do usuário com os vetores de componentes disponíveis na base e decidirá o componente mais próximo da descrição, ativando o neurônio que representa este componente.

O processo de construção do vetor de um componente é bastante simples. Inicialmente as posições do vetor referentes a termos de facetas presentes na descrição do componente são preenchidas com 1. As posições do vetor referentes a termos que possuam relações de similaridade com os termos que descrevem o componente são preenchidas com os valores destas relações de similaridade. As demais posições do vetor são preenchidas com zero.

Para efeito de exemplo, suponhamos um esquema de classificação que contenha os termos apresentados na Tabela 1. Neste esquema, temos apenas duas facetas, **Agente** e **Linguagem**, com dois e três termos respectivamente. O espaço de representação de componentes reutilizáveis definido por este esquema de classificação possui 5 dimensões (R^5), uma para cada termo de faceta. Cada termo de faceta define uma dimensão do espaço de representação dos componentes reutilizáveis.

Faceta Agente	Faceta Linguagem
Árvore Binária	Eiffel
Lista	C
	C++

Tabela 1 - Um exemplo de esquema de classificação

Suponhamos um componente reutilizável descrito pelos termos **Lista** e **C++** e uma relação de similaridade entre os termos **C** e **C++**, no valor do **0,5**. A Figura 2 apresenta a representação vetorial e espacial do componente **Lista / C++**.

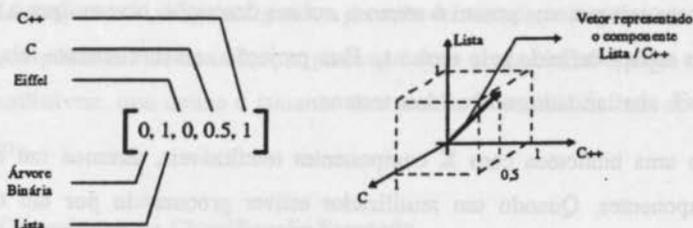


Figura 2 - O vetor de representação de um componente

III.1. Cálculo de Distância entre Componentes Reutilizáveis

Em muitos casos, não é suficiente que o método de recuperação de componentes reutilizáveis indique qual o componente mais próximo da descrição do usuário, devendo este método indicar também a distância entre o componente e a descrição.

O método de cálculo da rede neural apresentada considera a distância espacial entre o vetor que representa o componente e o vetor que representa a descrição do usuário. Esta distância é medida aplicando-se a lei dos cossenos sobre o resultado do produto vetorial do vetor do componente com o vetor da descrição do usuário, que indica o cosseno do ângulo entre os vetores.

$$a^2 = b^2 + c^2 - 2.b.c.\cos \alpha$$

$$d^2 = 1^2 + 1^2 - 2.1.1.\cos \alpha$$

$$d = \text{raiz} (2 - 2.\cos \alpha)$$

$$d = \text{raiz} (2 - 2.\|u \cdot v\|)$$

Esta medida de distância, que varia entre 0 e raiz (2), pode ser convertida para uma medida percentual, dividindo-se seu valor por raiz (2) e multiplicando-a por 100%. Esta medida reflete a distância espacial entre os vetores de componentes reutilizáveis e da descrição do usuário. É importante observar que esta medida é diferente daquela utilizada pelo método padrão do esquema de classificação facetada, que é a média aritmética das distâncias conceituais entre os termos de facetas [13].

IV.Método de Treinamento da Rede

O método de treinamento nesta abordagem permite que a rede neural seja capaz de reconhecer e aprender as heurísticas utilizadas por um reutilizador na seleção de um componente, tornando-a um real benefício em uma ferramenta de reutilização. A rede neural apresentada não precisa do treinamento de ajuste de sinapses, como normalmente acontece na camada de Kohonen padrão [17], porque as classes em que uma descrição do usuário pode ser classificada, ou seja, os componentes reutilizáveis, são previamente conhecidas. Neste caso, ao invés do treinamento da matriz de sinapses é necessária a sua programação, durante a qual as descrições dos componentes reutilizáveis são copiadas para as sinapses da rede.

O objetivo do treinamento no modelo de rede neural proposto é o ajuste dos valores das relações entre os termos de facetas, modificando o grafo conceitual de relações de similaridade entre estes termos. Este ajuste é efetuado de acordo com o componente indicado pela rede neural como o mais próximo da descrição do usuário, e o componente efetivamente selecionado por este.

É importante notar que a definição de um grafo conceitual de similaridade entre termos de facetas é uma tarefa difícil e que consome muito tempo, como observa Prieto-Díaz [14]. Nossa

abordagem permite que os valores destas relações sejam definidos subjetivamente, sendo corrigidos pelas próprias ações dos usuários durante as buscas por componentes.

A rede é treinada de modo supervisionado e de forma transparente ao usuário. Dada uma descrição do componente desejado, a rede calcula o grau de proximidade de cada componente da biblioteca com esta descrição. Em seguida, ela apresenta ao usuário os componentes mais próximos à descrição. O usuário, por sua vez, seleciona um destes componentes. Se o componente selecionado pelo usuário é o mesmo que a rede indicou como o mais próximo da descrição, o comportamento da rede neural está correto e nenhum treinamento é necessário. Se o usuário selecionar outro componente, a rede neural errou em sua seleção e seu comportamento deve ser corrigido. Esta correção ocorre nos valores das relações entre os termos de facetas.

O método de treinamento proposto é chamado de **ajuste fino**. Este método de treinamento corrige apenas as relações que influenciaram na diferença de distância entre os componentes e a descrição do usuário. O treinamento ajusta as relações dos termos que diferem entre a descrição do usuário e o componente selecionado por ele, somando um passo de treinamento² nestas relações. Um segundo ajuste é feito nas relações dos termos que diferem entre a descrição do usuário e o componente selecionado pela rede neural, subtraindo um passo de treinamento destas relações.

Quando não houver nenhuma relação entre estes termos, o método de treinamento executa um **ajuste global**. O ajuste global soma um passo de treinamento em todas as relações que envolvam os termos do componente selecionado pelo usuário e subtrai um passo de treinamento de todas as relações que envolvam os termos do componente selecionado pela rede neural.

Cada passo do processo de treinamento da rede neural é feito a cada seleção de componente, corrigindo um pequeno conjunto de relações. O grafo conceitual é corrigido de forma suave, considerando-se um passo de treinamento pequeno. Reutilizadores com mais experiência podem ser atribuídos com passos de treinamento maiores, refletindo maior peso no

² Um passo de treinamento é uma constante numérica

treinamento da rede. Esta diferenciação de níveis de usuários evita que o uso contínuo da rede por reutilizadores inexperientes desalinhe o comportamento da rede neural.

V. Experimentos Realizados

Diversos experimentos foram realizados com o modelo apresentado para a camada de Kohonen [1, 2], utilizando-se um esquema de classificação com 15 termos, organizados em 3 facetas. Nove componentes reutilizáveis foram utilizados e 12 relações entre termos de facetas foram definidas.

Este esquema de classificação relativamente pequeno foi utilizado com o objetivo de permitir a visualização geométrica dos resultados da rede neural. Com 15 termos de facetas, os componentes reutilizáveis são representados no R^{15} . Entretanto, a organização dos termos por faceta, torna possível trabalhar em um espaço tridimensional, que é mais tratável. Como a representação dos componentes da biblioteca e da descrição do usuário é baseada em uma visão geométrica, as regras validadas em um espaço continuam válidas em um espaço de maior dimensão.

Três formas de normalização dos vetores de sinapses e quatro métodos de treinamento, incluindo os ajustes fino e global, foram analisados durante estes experimentos. O treinamento por **aproximação do vencedor** soma um passo de treinamento em todas as relações dos termos que compõem a descrição do componente selecionado pelo usuário.

O treinamento por **ajuste fino do vencedor** ajusta as relações dos termos que diferem entre a descrição do componente desejado pelo usuário e o componente selecionado por ele, somando um passo de treinamento nestas relações. Se não houver nenhuma relação entre estes termos, o método de treinamento soma um passo de treinamento em todas as relações que envolvam os termos do componente selecionado pelo usuário.

A análise dos quatro métodos de treinamento teve como objetivo verificar a convergência da rede neural em três casos distintos: um conjunto de relações inicialmente zeradas, um conjunto

de relações incorretas e um conjunto de relações com margem de 20% de erro. A Tabela 2 apresenta os resultados destes experimentos:

Modelo de Treinamento	Grau de Convergência com as Relações		
	Zeradas	Incorretas	Erro Controlado
Aproximação do Vencedor	98 %	82 %	84 %
Ajuste Global	95 %	90 %	99,17 %
Ajuste Fino	99,17 %	95 %	100 %
Ajuste Fino do Vencedor	95 %	92 %	99,17 %

Tabela 2 - Resultados dos experimentos realizados com os métodos de treinamento

O **ajuste fino** foi o método de treinamento que apresentou melhores resultados durante os experimentos. Quando treinada por este método, a rede neural converge rapidamente e com alto grau de acerto (100 % com o conjunto de relações com margem de erro controlada e 99,17 % com o conjunto de relações zerado).

Através dos experimentos realizados, foi constatado que o modelo de cálculo de distância espacial entre os componentes funciona melhor quando as relações entre os termos de facetas estão mais completas, ou seja, quando o grafo conceitual de relações entre um conjunto de termos tende a um grafo completo, com arestas entre cada dois termos. Isto ocorre porque as relações entre os termos têm um papel fundamental na descrição dos vetores de componentes, indicando a projeção dos vetores nos espaços definidos por outros termos.

Outro resultado interessante foi a verificação de que a distância espacial entre os vetores tende a evitar empates entre componentes quando comparados a uma descrição do usuário. Por ser mais contínua que a medida de distância padrão, as diferenças entre alguns componentes podem ocorrer em casas decimais muito pequenas, desempatando componentes muito próximos.

A distância entre componentes reutilizáveis pode ser representada por termos de uma variável linguística fuzzy [20]. Estes termos mapeiam os valores da distância em palavras de linguagem natural, transformando um espaço contínuo em um espaço discreto. Este mapeamento faz com que diversos componentes sejam mapeados no mesmo termo fuzzy, ocultando pequenas diferenças em sua medida de distância com a descrição do usuário.

VI.A Implementação do Modelo

A técnica de recuperação apresentada foi implementada como parte do ambiente de desenvolvimento de software **Memphis**, que está sendo desenvolvido na **COPPE/UFRJ**. O projeto **Memphis** tem como objetivo construir um ambiente de desenvolvimento de software baseado em reutilização [19]. Este ambiente, baseado no modelo dos ambientes instanciados **TABA** [15], permite a reutilização das informações ao longo do processo de desenvolvimento de software, sendo implementado, em sua primeira versão, em estações de trabalho **Sun**, com sistema operacional **Unix**, utilizando-se a linguagem de programação orientada a objetos **Eiffel** [11]. O processo de desenvolvimento tem como base o método de **Booch** [3].

No ambiente **Memphis**, documentos de software são o produto das atividades do processo de desenvolvimento. Estes documentos podem ser compostos por outros documentos de software e, em última instância, são compostos por produtos de software, representados na forma de grafos. Estes grafos são a representação de diagramas de requisitos, esquemas de projeto, código ou qualquer outro produto gerado durante o processo de desenvolvimento. Um grafo é composto por nós e ligações. Os nós representam os vértices do grafo, podendo assumir os papéis de encapsuladores e organizadores. As ligações representam as arestas do grafo, podendo assumir os papéis de herança, composição, associação ou troca de mensagem.

Produtos de software são manipulados por ferramentas, que são executadas de acordo com a ordem das atividades do processo de desenvolvimento. Estas ferramentas são classificadas em ferramentas internas e externas. Todas as ferramentas do ambiente possuem um conjunto padrão de serviços, entre os quais podemos citar a criação de nós e ligações em um grafo, a seleção e apresentação de um grafo e a recuperação de componentes reutilizáveis de software. Todas as ferramentas têm acesso a uma base central de conhecimento, que pode estar representada de diversas formas, entre elas, sistemas de produção, *frames*, redes semânticas e redes neurais.

Qualquer produto de software pode ser um componente reutilizável devendo para tanto ser classificado pelo esquema de classificação facetada do ambiente. Neste esquema, cada faceta

possui um dicionário de termos, que podem estar conectados por relações de similaridades. Estas relações expressam a distância conceitual entre os termos, que é representada por termos de uma variável linguística fuzzy.

O conjunto padrão de serviços das ferramentas do ambiente **Memphis** implementa duas técnicas de recuperação de componentes reutilizáveis: a classificação facetada simples e classificação facetada apoiada por redes neurais. Na seleção de um componente reutilizável, o usuário pode optar por uma das técnicas de busca. Mais informações sobre o ambiente **Memphis** podem ser encontradas em [1, 16, 19].

VII. Considerações Finais

Neste artigo apresentamos uma técnica de recuperação de componentes reutilizáveis, que utiliza uma rede neural baseada no mapa auto-organizável de Kohonen para corrigir as relações de similaridade entre os termos de um esquema de classificação facetada.

A interação do usuário no processo de recuperação é a mesma que no processo de seleção padrão do esquema de classificação facetada: o usuário especifica os termos para um conjunto de facetas e o algoritmo apresenta os componentes similares a esta descrição. O usuário seleciona um destes componentes, disparando automaticamente o processo de treinamento, conforme descrito na seção IV.

O tempo de busca da técnica de recuperação de componentes reutilizáveis apresentada é linear. Todos os componentes são analisados para a decisão sobre os mais próximos à descrição do usuário. Este tempo de busca pode ser reduzido, utilizando-se uma outra camada de Kohonen para agrupar componentes semelhantes. Esta rede neural pode ser treinada pelo método padrão de treinamento da camada de Kohonen. Na busca por um componente reutilizável, os grupos de componentes são apresentados à camada de Kohonen do método de recuperação descrito, e somente os componentes do grupo vencedor são analisados.

Merkel et al [12] utilizam uma rede neural baseada no mapa auto-organizável de Kohonen para determinar a projeção, em um plano bidimensional, de um conjunto de componentes

reutilizáveis de software. Esta projeção permite a análise da formação de grupos de componentes com características similares ("clusters"). Os componentes são representados por palavras chave, extraídas de sua documentação textual através de um processo de recuperação automática de informação.

A abordagem apresentada neste artigo é diferente da abordagem de Merkl et al, embora utilize um modelo de rede neural similar. Enquanto Merkl et al utilizam o processo de treinamento padrão da camada de Kohonen para determinar grupos de componentes similares, o processo de treinamento do nosso modelo de rede neural ajusta as relações entre os termos das facetas que classificam os componentes reutilizáveis.

Agradecimentos

Os autores deste trabalho agradecem ao **CNPq** e à **CAPES** pelo apoio financeiro.

Bibliografia:

- [1] Barros, M. "Recuperação de Componentes em Bibliotecas de Software: uma Abordagem Conexionista", Tese de Mestrado, Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ, 1995
- [2] Barros, M. et al "Retrieving Reusable Components from Libraries: a Conexionist Approach", Communication presented on the Fifth International Conference on Software Reuse, Orlando, USA, 1996
- [3] Booch, G. "Object-Oriented Analysis and Design with Applications", Second Edition, The Benjamin/Cummings Publishing Co., 1994
- [4] Diab, Mahmoud. "Software Reuse Repository", 4th Annual Workshop on Software Reuse, Syracuse, NY, 1991
- [5] Faget, J. "The REBOOT Approach to the Concept of a Reusable Component", 5th Annual Workshop on Software Reuse, USA, 1992
- [6] Girardi, M.R.; Ibrahim B. "Automatic Indexing of Software Artifacts", Third International Conference on Software Reuse, Rio de Janeiro, Brasil, 1994
- [7] Griss, M.L. et al. "Managerial and Organizational Reuse - Starting and Running a Software Reuse Program", em "Software Reusability", Ellis Horwood, 1994

- [8] Isoda, Sadahiro. "Experience Report on Software Reuse Project: Its Structure, Activities and Statistical Results", 14th Conference on Software Engineering, Melbourne, Australia, 1992
- [9] Katz, Samuel et al. "PARIS: A System for Reusing Partially Interpreted Schemas", 9th International Conference on Software Engineering, 1987
- [10] Maarek, Y. et al. "Automatic Generating Software Libraries without Pre-Encoded Knowledge", IBM Research Division, 1989
- [11] Meyer, B. "Applying 'Design by Contract'", IEEE Computer, 1992
- [12] Merkl, D. et al. "Learning the Semantic Similarity of Reusable Software Components", Third International Conference on Software Reuse, Rio de Janeiro, Brasil, 1994
- [13] Prieto-Díaz, R. "A Software Classification Scheme", Tese de Doutorado, University of California, 1985
- [14] Prieto-Díaz, R. "Classification of Reusable Modules", IEEE Software, Vol. 4, No. 1, 1987
- [15] Travassos, G.H. "O Modelo de Integração de Ferramentas da Estação TABA", Tese de Doutorado, Programa de Sistemas e Computação, COPPE/UFRJ, 1994
- [16] Travassos, G.H. et al. "Extensões do Projeto TABA para a Definição de: Controle de Processos, Ferramenta Interna, Conhecimento e Editor de Grafos", Relatórios Técnicos do Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ, 1995
- [17] Wasserman, Roger. "Neurocomputing: Theory and Practice", Van Nostrand Reinhold, 1989
- [18] Werner, C.M.L. "Reutilização de Software no Desenvolvimento de Software Científico", Tese de Doutorado, Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ, 1992
- [19] Werner, Cláudia M.L. et al. "Memphis: Um Ambiente para Desenvolvimento de Software Baseado em Reutilização", Relatório Técnico do Projeto Memphis, COPPE/UFRJ, 1996
- [20] Zadeh, Lotfi A. "Making Computers Think Like People", IEEE Spectrum, 1994