Mecanismo de Busca em um Repositório de Artefatos de Software usando Algoritmos Genéticos

Beatriz Borsoi[1], Nathanyel Sandi[1], Vagner Lúcion[1]

[1]Departamento de Informática – Universidade Tecnológica Federal do Paraná(UTFPR)
CEP – 80.513-390 – Pato Branco – PR – Brasil

beatriz@utfpr.edu.br, nathan.sandi@gmail.com, vagnerlucion@gmail.com

Abstract. *The software reuse has as one of its most obvious objectives to reduce the time of software development projects through the use of artifacts (parts of the system) already ready. Reuse is also related to software quality because the artifacts are being reused more widely tested for use in different projects. In this work, artifacts are all products resulting carrying out the activities of the software lifecycle and that can somehow be stored. As a contribution to reuse by recovery software artifacts repository, a search engine using genetic algorithms was developed and is presented in this article.*

1. Introdução

O uso de código de software produzido em outros projetos e de funcionalidades implementadas para serem utilizadas em projetos distintos é a forma mais evidente de reuso. A técnica de modularização e o paradigma de orientação a objetos pela essência do seu conceito e pelo mecanismo de herança, possibilitam o reaproveitamento de código. O desenvolvimento baseado em componentes também está fundamentado no reuso de elementos de código. Porém, no desenvolvimento de software há outras possibilidades para reuso, como, por exemplo, de documentos de análise e de projeto, planos de teste, padrões de projeto e de experiências e conhecimento adquiridos com a realização das atividades.

Esse contexto permite identificar que reuso pode ser aplicado a todo o ciclo de vida de software. Justo (1996) argumentou que não há razões teóricas que impossibilitem a aplicação de reuso nas fases iniciais do desenvolvimento de um software.

Neste trabalho, reuso abrange todos os produtos que resultam direta e indiretamente da realização de atividades relacionadas ao ciclo de vida de software. Esses produtos são denominados artefatos e são representados por: componentes de código (como Dynamic Link Libraries (DLL), rotinas e funções), artefatos e documentos (diagramas, planos de testes, modelos e padrões para produzir documentação, dentre outros) e procedimentos (métodos, técnicas e orientações para realizar as atividades, que podem ser resultantes da experiência dos membros da equipe, por exemplo).

O resultado deste trabalho, para o repositório e para o mecanismo de busca, se destina à indústria de software no sentido de auxiliar para que produtos de software sejam produzidos em tempo e custo menores e com maior qualidade. Os artefatos que são objetos de reuso possivelmente já foram utilizados em outros projetos e, portanto, foram mais amplamente testados. Para que artefatos possam ser efetivamente reusados é

necessário que eles sejam adequadamente armazenados e que haja mecanismos eficientes para recuperá-los. Com o resultado da realização deste trabalho destacam-se as seguintes contribuições: o desenvolvimento de mecanismo de busca em repositório de artefatos de software utilizando algoritmos genéticos e a definição de metadados para o cadastramento dos artefatos no repositório.

O mecanismo de busca propostos implementa algoritmos genéticos e atua em um repositório de artefatos de software desenvolvido por Ariati (2012). Assim, será possível verificar algoritmos genéticos apresentam melhor efetividade na busca. A efetividade está relacionada ao atendimento dos requisitos indicados como critério de pesquisa aos artefatos resultantes de uma busca.

Este texto está organizado da seguinte forma: na Seção 2 são apresentados conceitos sobre artefatos, reuso e repositório de software; a Seção 3 apresenta o referencial teórico sobre algoritmos genéticos; na Seção 4 está a metodologia utilizada para implementação do mecanismo de busca proposto; e o resultado é apresentado na Seção 5. Por fim está a conclusão, seguida das referências bibliográficas.

2. Artefatos de Software

Para a Objetct Management Group (OMG) (2005), de acordo com o definido na especificação Reusable Asset Specification (RAS) para ativos de software reutilizáveis, um ativo reutilizável provê uma solução para um problema em um determinado contexto. Um ativo agrega um conjunto de artefatos. Um ativo pode ter pontos de variabilidade que possibilitam sua customização, possui regras de uso e um contexto de aplicação definido.

Um ativo de software, pela definição proposta por Ezran, Morisio e Tully (2002), é qualquer artefato de software passível de reutilização, incluindo documentação, padrões de análise e projeto, normas de codificação, dentre outros.

Um conceito bastante utilizado para componente de software e que pode aplicar-se ao conceito de artefato utilizado neste trabalho é apresentado por Sametinger (1997), citado em Peres (2006), que define componente de software como alguma parte do sistema de software que é identificável e reutilizável. Para Lucredio (2006) e Sametinger (1997) artefatos reutilizáveis são auto-contidos, explicitamente identificáveis, descrevem ou realizam funções específicas e têm interfaces bem definidas e documentação apropriada.

2.1. Reuso Artefatos de Software

Segundo Prieto-Diaz (1991), reuso está relacionado ao uso em situações novas de conceitos ou produtos previamente adquiridos ou construídos. Sendo que esses produtos devem ser definidos e/ou produzidos visando reuso, estarem armazenados de forma a facilitar a sua localização e a identificação de similaridade entre situações novas e antigas permitindo a adaptação e uso.

A reutilização de artefatos visa reduzir custo e tempo no desenvolvimento, minimizando a complexidade e melhorando a qualidade do software produzido. Baseado em estudo sobre reuso, a *Quantitative Software Management Associates* relatou que a utilização de componentes levou à redução de 70% no tempo do ciclo de

desenvolvimento e de 84% no custo do projeto [Lycett 2011].

2.2. Repositório de Artefatos de Software

Um repositório de artefatos de software é um sistema de software para o armazenamento de resultados da realização das atividades do ciclo de vida de software. Um repositório deve possibilitar que seus artefatos sejam catalogados de maneira que possam ser adequadamente caracterizadas visando facilitar a busca. A efetividade da busca pode estar relacionada aos metadados dos artefatos que devem permitir a definição de características que os descrevam, como: nome, descrição textual, tipo de artefato, data de criação e observações essenciais. Para definir os metadados para o cadastro dos artefatos no repositório desenvolvido por Ariati (2012) no qual atua o mecanismo de busca reportado neste artigo foi utilizado como base a especificação RAS (OMG, 2005) e o apresentado em Redolfi et al. (2005), que faz um apanhado do trabalho de diversos outros autores.

Frakes (2005) destaca que repositórios de componentes devem adotar metadados para representar componentes, como também prover mecanismos para certificação da qualidade desses componentes. Além disso, Frakes também sinaliza que repositórios devem incluir facilidades de gerência de configuração, provendo recursos para controle de mudanças, gerência de métricas de reuso e adotar mecanismos que viabilizem o retorno de investimento na negociação de componentes.

3. Algoritmos Genéticos

Algoritmo genético possui fundamentação na evolução biológica [Mitchell 1996], ou seja, o princípio da evolução das espécies e na genética, constituindo um modelo matemático que simula a teoria da evolução Darwiniana [Gonçalves 2013]. Esses algoritmos são baseados no princípio de sobrevivência dos mais aptos e na reprodução [Pacheco 1999]. Nesse modelo, existe um conjunto de indivíduos (população inicial) que representa as possíveis soluções para um determinado problema e a cada iteração (geração) os indivíduos são avaliados e os mais aptos são selecionados para gerar descendentes, produzindo a nova geração.

O algoritmo genético consiste na criação de uma população e na reprodução com suas respectivas mutações, descartando os indivíduos com menor nível de adaptabilidade ao meio [Lucas 2002]. Indivíduos com mais descendentes têm mais chance de perpetuarem seus códigos genéticos nas próximas gerações. Tais códigos constituem a identidade de cada indivíduo e estão representados nos cromossomos. Na construção de algoritmos computacionais que se baseiam nesse princípio, uma melhor solução para um determinado problema é buscada pela evolução de populações de soluções codificadas por cromossomos.

Muitos dos conceitos utilizados em algoritmos genéticos estão relacionados a processos naturais [Thengade, et al. 2012; Prebys 2007]:

a) Indivíduo – qualquer solução possível para o problema considerado.

b) População – grupo de indivíduos.

c) Espaço de busca – todas as possíveis soluções para o problema.

d) Cromossomo – a representação de um indivíduo por meio de suas características.

e) Gene – característica. Aspecto possível a um indivíduo.

f) Alelo – configurações possíveis para uma característica.

g) Locus – a posição de um gene em um cromossomo.

h) Genoma – coleção de todos os cromossomos de um indivíduo.

i) Fitness – é uma medida do quão bem o cromossomo resolve o problema. Para um algoritmo genético padrão, o fitness é uma função do conjunto de possíveis cromossomos que solucionam o problema.

j) Seleção – é o processo de escolha de pares de organismos para reprodução.

k) Crossover – é o processo de cruzamento em que há troca de genes entre pares de indivíduos em reprodução.

l) Mutação – é o processo de alterar aleatoriamente os cromossomos.

Ressalta-se que embora nem todos esses termos estejam explícitos nesse texto, os conceitos e as ideias envolvidas nos mesmos são utilizados no desenvolvimento no mecanismo de busca desenvolvido e reportado neste texto.

4. Metodologia de implementação

Na técnica de algoritmo genético inicialmente é realizada a definição dos genes. Uma série de genes forma um cromossomo que representa uma possível solução para o problema considerado. No repositório de artefatos, um gene é uma palavra-chave associada a um artefato. Assim, um artefato é um cromossomo. A Figura 1 apresenta esquematicamente o artefato como um cromossomo. Na representação dessa figura, 'PC', indica palavra-chave, e 'A', 'B', 'C', 'D', 'E', as palavras-chave distintas associadas ao artefato. Cada gene é uma palavra-chave que o usuário indicou como critério de busca.

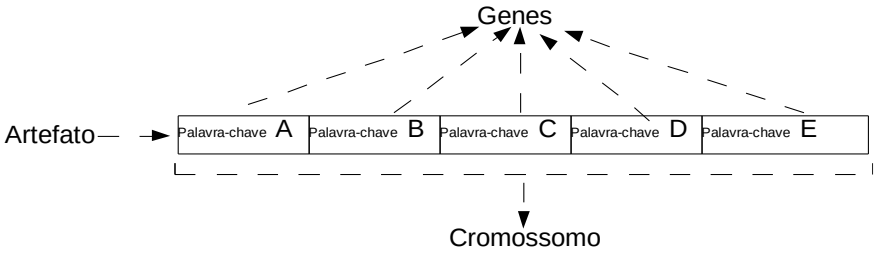

Figura 1 – Representação esquemática de um cromossomo para um artefato

Após a definição do gene e do cromossomo é realizada a criação da população inicial. No algoritmo proposto, essa população são os artefatos que possuem pelo menos uma palavra-chave associada que é igual ao indicado na busca, independentemente do grau de importância atribuído à mesma.

A partir da população inicial é realizado um processo iterativo de refinamento ou evolução das soluções. Novas soluções são criadas por combinação e refinamento

realizadas por seleção e cruzamento. São essas operações que produzem novas soluções que formam uma nova população, ou seja, uma geração. A geração é avaliada pelo *fitness*, que consiste na diferença do grau de importância da palavra-chave indicada na busca, com a que está associada ao artefato. O algoritmo proposto utiliza a técnica de cruzamento para combinação das palavras-chave e seus graus de importância (informado pelo especialista e pelo usuário).

A Figura 2 apresenta o esquema de funcionamento do algoritmo desenvolvido para a seleção dos artefatos no repositório que atendem aos critérios de busca.

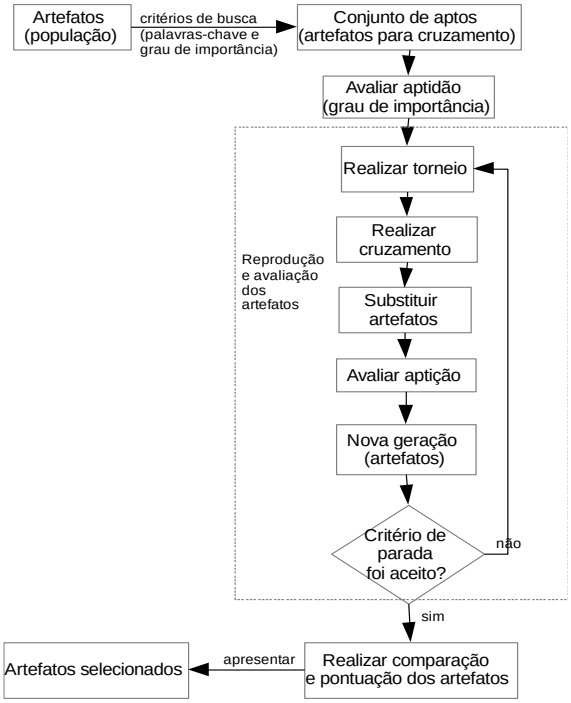

Figura 1 – Representação esquemática do algoritmo

O Quadro 1 mostra a representação da Figura 2 com a descrição das fases.

Quadro 1. Descrição das etapas do algoritmo (descritas na Figura 2)

Etapa	Descrição
Artefatos (população)	São todos os artefatos armazenados no banco de dados.
Critérios de busca	Palavras-chave e o respectivo grau de importância associado às mesmas informadas como critério de busca. O grau de importância define a relevância da palavra-chave associada ao artefato como critério de busca.
Conjunto de aptos	São os artefatos, dentre os armazenados no banco de dados, que atendem de alguma forma aos requisitos de busca.
Avaliar aptidão	Definir os artefatos que serão considerados na primeira iteração do algoritmo.
Nova geração	Realização do processo iterativo de geração de novas populações de artefatos que atendam aos requisitos indicados na busca, que são as palavras-chave e seus respectivos graus de importância.
Realizar torneio	Estabelecido como regra de manter a população para cruzamento com a mesma quantidade de indivíduos que o conjunto de aptos. O torneio visa obter os artefatos com melhor fitness, ou seja, mais aptos a competir com outros em futuras gerações. No conjunto gerado poderá haver artefatos repetidos.
Substituir artefatos	Verificar os artefatos obtidos a partir do cruzamento que podem substituir os seus pais, ou seja, que apresentam melhor aptidão.
Avaliar aptidão	Avaliar os artefatos obtidos.
Nova geração	São os artefatos resultantes do cruzamento compondo a população que passará pelo processo de reprodução e avaliação.

Critério de parada	Indica quando deve ser finalizado o processo de torneio, cruzamento, substituição de indivíduos e avaliação de fitness, ou seja, quando o conjunto obtido de descendentes é considerado satisfatório.

O Quadro 2 apresenta o algoritmo proposto para o mecanismo de busca utilizando algoritmos genéticos. Esse algoritmo se baseia diretamente no fluxograma da Figura 2.

Quadro 2. Descrição da busca por algoritmo genético

Conjunto de Aptos (Palavras Chave, Graus de Importância da busca)
Proximidade = Grau de importância do especialista – Grau de Importância da busca
Aptos = Palavras chaves com fitness maior (Menor proximidade)
Retorna artefatos aptos

Realizar Torneio (Artefatos Aptos)
Enquanto Torneio < Conjunto Apto
Escolha randômica de dois Artefatos
Seleciona melhor fitness
Retorna artefatos vencedores do torneio

Realizar Cruzamento (Artefatos vencedores do torneio)
Enquanto quantidade de cruzamento < (Quantidade de artefatos)/2
Seleciona dois indivíduos da lista em sequencia
Corta cromossomo (Vetor de palavras chaves e graus de importância)
Artefatos selecionados = Novos indivíduos

Obter descendentes (Artefatos selecionados)
Enquanto tem indivíduos do conjunto de artefatos
Realiza a avaliação do fitness, pela equação:

$$proximidade = \sum_{i=0}^{n-1} P_{informada} - P_{iartefato}$$

Onde n é a quantidade de palavras-chave e P é a importância associada à palavra-chave.
Se Fitness é alcançado
Verifica se atingiu o critério de parada
Se não
Realiza torneio e processo de cruzamento
Se Critério de parada é atendido
Avalia pontuação
Descendentes = Artefatos que possuem menor fitness
Pontuação = Compara (Descendentes, Artefatos do Banco)
Retorna Artefatos de Maior Pontuação

Para o cálculo da diferença das pontuações, são escolhidos somente os artefatos que apresentam diferença entre os graus de importância seja de 2, 1 ou 0 utilizando a seguinte especificação que define a pontuação do artefato: 0 - se não há palavra-chave semelhante; 1 - se diferença é de dois; 2 - se diferença é de um; 3 - se diferença é zero

A Figura 3 exemplifica a forma de realização do cruzamento, indicando um ponto de corte, implementada pelo algoritmo proposto para o mecanismo de busca utilizando algoritmos genéticos. O ponto de corte se refere à segmentação dos cromossomos para a geração de descentes, ou seja, a população para a próxima iteração. Na representação da Figura 3 ‘A1’, ‘B2’ e demais siglas representam as palavras-chave associadas aos artefatos e os novos artefatos obtidos a partir do cruzamento dessas palavras-chave.

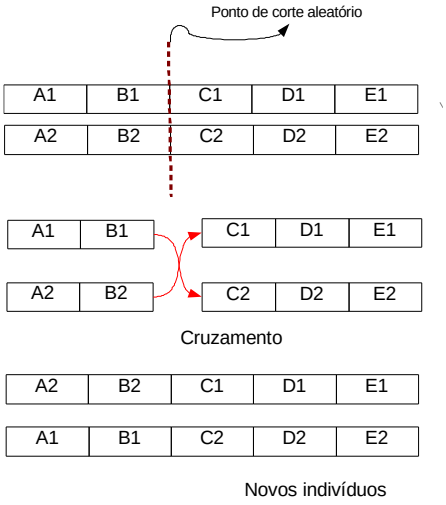

Figura 3 – Representação esquemática de cruzamento de indivíduos

A Figura 4 ilustra a forma de classificação dos artefatos obtidos do repositório com o mecanismo de busca utilizando algoritmos genéticos.

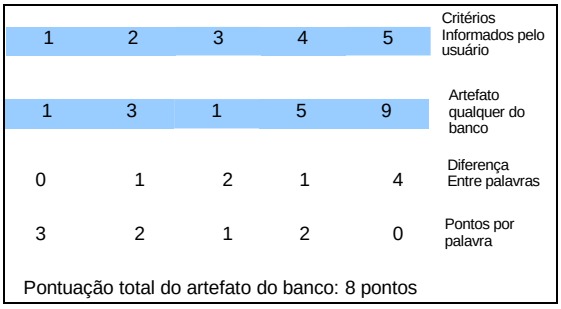

Figura 4 – Representação esquemática da classificação dos artefatos

Ressalta-se que na realização do torneio pode ocorrer perda de palavras-chave com grau de importância próximo ou igual ao indicado na busca. Isso porque um artefato pode ter fitness ruim (pontuação baixa) decorrente da existência de uma única palavra-chave igual ou pelas palavras-chave terem grau de importância muito distinto entre o informado na busca e o indicado no cadastro do artefato. Essa é considerada uma restrição do algoritmo.

5. Resultados Obtidos

O mecanismo de buscar foi implementado no repositório de artefatos de software proposto por Ariati (2012), a partir desse mecanismo foram realizados testes com diferente tamanhos de entradas (quantidade de artefatos cadastrados no banco de dados). A validação desses testes foi realizada em diferentes etapas: inicialmente em testes de mesa para um pequeno conjunto de artefatos, em seguida foram comparados resultados de buscas por algoritmos genéticos com busca binária usando SQL, após os testes de funcionalidade, o algoritmo. Essa busca foi desenvolvida por Ariati (2012) pela existência e não existência de palavras-chave indicadas na busca em comparação com palavras-chave cadastradas nos metadados dos artefatos. Assim, foram comparados os

artefatos retornados pelos dois mecanismos de buscas e o número de artefatos não identificados devido à precedência de máximos ou mínimos locais (descartados pelo algoritmo genético). Finalmente, um levantamento do tempo de execução foi realizado para diferentes conjuntos de testes, junto com a análise da complexidade do algoritmo.

A validação do algoritmo foi realizada pela avaliação em um conjunto pequeno de artefatos, verificando cada etapa realizada e averiguando com o que foi apresentado no método descrito na sessão anterior. Não houve anomalias durante as execuções, sendo em todas as execuções apresentados artefatos e valores de fitness esperados.

A Figura 5 apresenta os resultados obtidos quanto ao retorno dos artefatos pelo método de algoritmos genéticos comparado com uma busca SQL. O número de artefatos apresentáveis na busca SQL consiste no número de artefatos mostrados para o usuário no caso de uma busca binária.

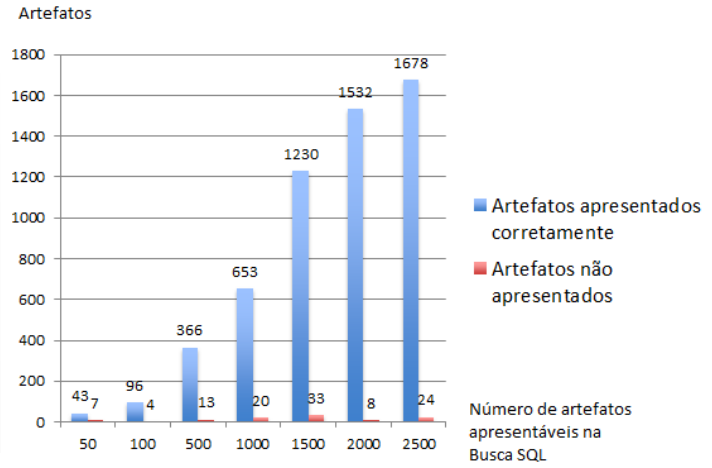

Figura 5 – Número de artefatos apresentados e não apresentados pelos mecanismos de busca SQL e com algoritmo genético.

O objetivo desse teste foi verificar o número de artefatos que foram apresentados corretamente (artefatos presentes nas duas buscas) e o número de artefatos descartados devido aos mínimos e máximos locais. Nesta análise são desconsiderados artefatos não apresentados devido a não satisfazer a pontuação mínima estabelecida. A partir do gráfico da Figura 5 é possível concluir que a perda de artefatos é pequena e o mecanismo de busca é eficiente.

A complexidade do algoritmo genético é linear, feita com base na análise do código (relacionando as condicionais e situações de iteratividade). A Figura 6 mostra a relação de tempo de execução por tamanho de entrada em três máquinas distintas para coletar diferentes tempos de resposta de acordo com o processamento. Os tempos de execução são relativamente altos quando o objetivo é tempo de resposta, em comparação com uma busca binária. Porém esse tempo é previsível, uma vez que algoritmos genéticos têm um alto número de iterações até que o critério de parada seja satisfeito.

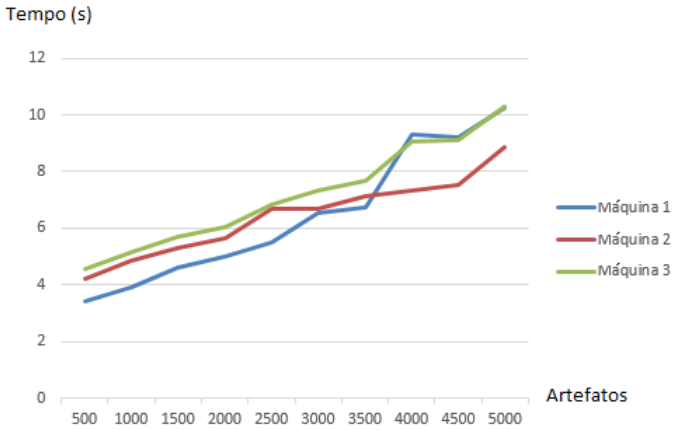

Figura 6 – Tempo de processamento em decorrência da quantidade avaliada de artefatos.

6. Conclusão

Este trabalho se refere a reuso de artefatos de software com a proposta de um mecanismo de busca utilizando algoritmos genéticos visando reuso. Reuso está vinculado ao armazenamento dos artefatos (implementação de repositórios, definição de metadados) e a forma de busca (mecanismos que permitam localizar os artefatos no repositório de acordo com critérios de busca indicados). O desenvolvimento deste trabalho permitiu verificar que a efetividade do reuso está diretamente relacionada aos mecanismos de busca destes artefatos e aos seus metadados.

Antes de definir e implementar os algoritmos de busca, os metadados propostos por Ariati (2012) foram revistos e complementados. Essas complementações permitiram realizar uma associação mais adequada entre artefatos e versões de artefatos. Os metadados foram associados às versões dos artefatos, porque alterações significativas podem ocorrer entre versões.

O algoritmo de busca utilizando algoritmos genéticos apresentou resultados satisfatórios, pois o conjunto de teste apresentou resultados coerentes com a base de dados, ordenados de acordo com o *fitness*. Vale lembrar que pode ocorrer em algum artefato a possibilidade do algoritmo cair em um extremo (mínimo ou máximo local) e assim não retornando a saída ótima, mas uma sub-ótima. Nos testes realizados não ocorreram casos de incoerência nos resultados.

O mecanismo de busca usando algoritmos genéticos consiste na definição dos genes, que neste caso são os artefatos que possuem associada pelo menos uma das palavras-chave indicadas na busca. As palavras-chave e seus graus de importância associados definem os genes utilizados nos cruzamentos e gerando os artefatos que melhor atendem aos critérios de busca. A maior dificuldade na implementação do mecanismo utilizando a técnica de algoritmos genéticos esteve centrada na adaptação dos conceitos de algoritmos genéticos para a recuperação de artefatos.

Como implementação futura, a otimização do algoritmo visando melhoria dos resultados e redução do tempo de execução. Também o desenvolvimento de outros mecanismos de busca utilizando outras técnicas de Inteligência Artificial.

Referências

Ariati, A. (2012) “Sistema web para armazenamento e recuperação de artefatos de software”, Trabalho de Conclusão de Curso (Graduação) – Tecnologia em Análise e Desenvolvimento de Sistemas. Universidade Tecnológica Federal do Paraná.

Ezran, M., Morisio, M. Tully, C. (2002) “Practical software reuse”, http://books.google.com.br/books?id=kIlpuK1GkLwC&pg=PA3&source=gbs_toc_r&cad=3#v=onepage&q&f=false, Fevereiro.

Frakes, B. and Pole, P. (1994) “An empirical study of representation methods for reusable software component”, IEEE Engin, v. 20, n. 8, p. 617-630.

Gonçalves, R. (2002) “Algoritmos genéticos”, http://www.scribd.com/doc/98647248/Algoritmos-Geneticos. Acessado em fevereiro.

Lucas, C. (2002) “Algoritmos genéticos: uma introdução”, Universidade Federal do Rio Grande do Sul – Porto Alegre.

Lucrédio, D. Fortes, M. S. Meira, S. (2006) “The Draco approach revisited: model-driven software reuse”. In: VI WDBC,p. 72–79.

Lycett, Mark. “Understanding variation in component-based development: case findings from practice”. Information and Software Technology Journal, v. 43, p. 203-213.

Michell, M. (1996) “An introduction to genetic algorithms (complex adaptive systems)”. London, England: The MIT Press.

OMG (2005). “Reusable asset specification OMG available specification”. Version 2.2.

Pacheco, A. M. (1999). “Algoritmos genéticos: princípios e aplicações.” Em: ICA: Laboratório de Inteligência Computacional Aplicada – Pontifícia Universidade Católica, Rio de Janeiro.

Prebys, E. K. (2007) “The genetic algorithm in computer science”. MIT Undergraduate Journal of Mathematics, p. 165-170.

Prieto-Diaz, R. (1991). “Implementing faceted classification for software reuse. Communications of the ACM”, v. 34, n. 5, p. 89-97.

Redolfi, G. et al. (2005) “A reference model for reusable components description”. In: 38th Hawaii International Conference on System Sciences, p. 282-291.

Sqmetinger, J. (1997) “Software engineering with reusable components”. Berlin Heidelberg: Springer-Verlag.

Thengade, A., Dondal, R. (2012) “Genetic algorithm – survey paper”. MPGI National Multi Conference. Recent Trends in Computing. Proceedings published by International Journal of Computer Applications, p. 25-29.