

UMA PESQUISA SOBRE FERRAMENTAS CASE PARA ENGENHARIA REVERSA ESTÁTICA

Juliana da Silva Cindra

Especialista em Análise de Projeto e Gerência de Sistemas de Informação/IFF
Quali-EPT/ IFF
jcindra@iff.edu.br

Mara Regina dos Santos Barcelos

Graduada em Tecnologia em Desenvolvimento de Software/IFF
Quali-EPT/ IFF
mrbarcelos@iff.edu.br

Jonivan Coutinho Lisboa

Mestre em Computação Aplicada e Automação/UFF
Quali-EPT/ IFF
jlisboa@iff.edu.br

RESUMO

Este artigo apresenta uma síntese de resultados obtidos na pesquisa de uma ferramenta para auxílio no processo de engenharia reversa do código do Projeto SIGA-EDU. A engenharia reversa de software auxilia o desenvolvedor a entender como o software funciona internamente, permitindo estudar e aprender sua estrutura e lógica, além de melhorar a qualidade do software, tanto em segurança quanto em desempenho. Assim, este trabalho tem por objetivo estudar a engenharia reversa de software realizada por algumas ferramentas selecionadas de acordo com determinados critérios, analisando seus resultados e verificando seus pontos fortes e fracos.

ABSTRACT

This article presents a synthesis of results achieved in the research of a tool to help performing reverse engineering of code for Project SIGAEDU. Reverse Engineering of Software helps the programmer to understand how software works internally, allowing to study and to learn its structure and logic, and also improve the quality of software considering security and performance issues. Thus, this work aims to study the reverse engineering of software performed by some selected tools, according to determined criteria, analyzing their results and verifying their strengths and weaknesses.

1. INTRODUÇÃO

Há tempos, a engenharia reversa é utilizada no universo do hardware para desmontagem de equipamentos, com o intuito de descobrir os segredos de projeto e manufatura de concorrentes. A engenharia reversa de software possui semelhança com a engenharia reversa de hardware, com a principal diferença de que o programa que passa pelo processo não é de um concorrente, mas sim um trabalho próprio. O objetivo principal é entender os “segredos” do sistema através da análise do programa, buscando criar uma representação do mesmo em um nível de abstração mais elevado [6]. A engenharia reversa, como mencionada, é uma técnica empregada em várias partes da Engenharia de Software, sendo bastante utilizada no auxílio à manutenção de sistemas. Seu propósito é re-documentar o sistema e descobrir informações ocultas no projeto, proporcionando uma melhoria na compreensão do sistema.

Para programas orientados a objetos, partindo-se do nível de implementação para o de projeto, é possível a extração de informações com o auxílio de ferramentas CASE (*Computer-Aided Software Engineering*), que podem ser categorizadas como ferramentas de análise estática ou de análise dinâmica [6]. As ferramentas estáticas são denominadas ferramentas de visualização de código [9], pois possibilita ao

engenheiro de software visualizar o programa extraindo-se informações do código, o que permite melhorar enormemente a qualidade das mudanças e a produtividade das pessoas que as fazem [6]. Já as ferramentas dinâmicas monitoram a execução do software, e usam as informações obtidas durante a monitoração para construir um modelo comportamental do programa [6].

Este trabalho possui foco na análise estática, apresentando resultados de uma pesquisa sobre ferramentas para a extração da estrutura estática de software. Tal estrutura pode ser representada em digramas de classe da UML (*Unified Modeling Language*), apresentando de forma gráfica os elementos estáticos, como classes e relacionamentos de um programa. Sendo assim, são apresentadas algumas ferramentas CASE para a atividade de engenharia reversa que foram testadas para auxílio à recuperação do modelo de domínio do código do Projeto SIGA-EDU (Sistema Integrado de Gestão Acadêmica – Módulo Educacional).

O SIGA-EDU é um dos dois sub-módulos do módulo SIGA-EPT (Sistema Integrado de Gestão Acadêmica da Educação Profissional e Tecnológica) do projeto RENAPI (Rede Nacional de Pesquisa e Inovação). O módulo SIGA-EPT visa o desenvolvimento de um sistema integrado de gestão acadêmica, utilizando-se software livre, que possa tanto beneficiar as Instituições Federais de Ensino no apoio à sua gestão, quanto prover o MEC (Ministério da Educação) de informações e indicadores institucionais. O projeto é mantido pelo MEC e está sendo desenvolvido por diversos núcleos de pesquisa e desenvolvimento instalados em instituições da Rede Federal de Educação Profissional e Tecnológica em diversas regiões do país [7]. O sistema SIGA-EPT é desenvolvido em Java e encontra-se sob licença GNU-GPL.

Neste artigo, são apresentadas algumas ferramentas para a engenharia reversa do sistema SIGA-EDU, com suas respectivas descrições e características. O objetivo é identificar o tipo de suporte oferecido ao processo de engenharia reversa e verificar quais das ferramentas recuperam o maior número de informações relevantes do sistema SIGAEDU através da análise estática, verificando qual das ferramentas avaliadas é melhor adequada para o trabalho. A pesquisa foi conduzida pelo Quali-EPT – Núcleo de Garantia da Qualidade de Software da RENAPI.

Além desta Seção introdutória, o artigo contém mais três Seções: a Seção 2 trata da metodologia da pesquisa; a Seção 3 apresenta as ferramentas CASE selecionadas para engenharia reversa, e os testes realizados com as ferramentas para o estudo de caso; a Seção 4 faz uma análise dos resultados obtidos; finalmente, a Seção 5 traz uma conclusão do trabalho.

2. METODOLOGIA DA PESQUISA

O grande número de ferramentas CASE disponíveis no mercado, a complexidade de avaliação e escolha dessas ferramentas, bem como a importância que tais ferramentas têm na engenharia de software, foram fatores fundamentais para a criação de uma norma para orientar a avaliação e seleção de tais ferramentas.

A norma ISO/IEC 14102 [2], publicada pela ISO (*International Organization for Standardization*), auxilia o mercado comercial a identificar requisitos organizacionais para as ferramentas CASE e estabelecer uma relação entre tais requisitos e as características das ferramentas a serem avaliadas.

Além disso, a norma também descreve um processo para selecionar as ferramentas mais apropriadas dentre diversas candidatas, seguindo-se quatro processos genéricos (Figura 1), adaptando-os de acordo com as necessidades de cada caso. Os processos são: iniciação, estruturação, avaliação e seleção.

operacional Linux, distribuição Ubuntu 8.04.

3.1. ArgoUML

O ArgoUML é uma ferramenta CASE desenvolvida pela comunidade de desenvolvedores de código livre Tigris, vinculada à Universidade da Califórnia em Berkeley. A ferramenta é distribuída através da licença BSD (*Berkeley Software Distribution*), possui código aberto e está disponível em diversos idiomas. Este software é desenvolvido em Java e pode ser usado nas plataformas que possuam Java nas versões 5 ou 6. Para os testes, essa ferramenta foi uma boa opção, pois importa código Java, exporta no formato XMI (*XML Metadata Interchange*), e é livre.

A versão utilizada da ferramenta foi a 0.26, para a realização da engenharia reversa do sistema SIGA-EDU. Ao realizá-la, foi possível recuperar a estrutura de pacote do sistema, e os seguintes elementos de UML: classes, métodos, atributos, interface e componente. Os relacionamentos recuperados foram: abstração, generalização e dependência. A ferramenta não conseguiu recuperar os outros tipos de relacionamentos contidos no código do sistema SIGA-EDU, como associação uni e bidirecional, agregação e composição.

Um diferencial da ferramenta é a visualização de propriedade dos elementos UML quando selecionados, exibindo uma espécie de rastreabilidade entre elementos. Além disso, a ferramenta é portátil, funcionando em todo sistema operacional que tenha a máquina virtual Java. Esta ferramenta não gera automaticamente diagramas de classe. Para se visualizar os diagramas deve-se criar um novo diagrama e arrastar os elementos para ele.

3.2. Jude Community

Jude Community é uma versão gratuita da ferramenta Jude (*Java and UML Developers' Environment*), com suporte básico para diagramas UML, realizando importação e exportação de código na linguagem Java. A redistribuição de Jude Community é permitida a professores e estudantes de instituições de ensino, sendo restrito às suas instalações. A ferramenta permite trabalhar com oito tipos de diagramas, tais como: classes, caso de uso, seqüência, desenvolvimento, dentre outros. É possível realizar exportação em Java, HTML ou exportar os diagramas em forma de imagem.

A versão da ferramenta utilizada nos testes foi a 5.3. A importou normalmente o código Java do SIGA-EDU, gerando a estrutura de pacotes. Foram recuperados: classes, métodos, atributos, interfaces e estruturas de dados como listas (*lists*), conjuntos (*sets*), mapas (*maps*), e coleções (*collections*). Os relacionamentos recuperados foram: uni- e bidirecionais e herança.

A ferramenta não gera automaticamente os diagramas UML. Para conseguir isso, é necessário criar um novo diagrama e arrastar os elementos para ele. A ferramenta possui versões para sistemas Windows e Linux. Uma desvantagem da versão Community de Jude é o não-suporte ao formato de arquivo XMI, característica esta encontrada na versão Jude Professional, que não é gratuita.

3.3. NetBeans

O projeto de código aberto NetBeans é um IDE (*Integrated Development Environment*) estabelecido pela Sun Microsystems em junho de 2000. A ferramenta é escrita em Java, mas suporta outras linguagens de programação, como por exemplo C, C++, Ruby e PHP, e também suporta linguagens de marcação como XML e HTML. A ferramenta possui uma vasta documentação bem organizada, e uma de suas vantagens é a portabilidade do sistema, por ser escrito em Java, funcionando em qualquer sistema operacional que suporte JVM (*Java Virtual Machine*). O projeto NetBeans está sob as condições da licença SPL (*Sun Public License*), uma variação da MPL (*Mozilla Public License*).

Os recursos de modelagem UML para NetBeans são fornecidos pelo NetBeans UML Project, um *plug-in* que permite aos usuários da IDE modelar seus projetos, gerar código através de seus modelos ou realizar a engenharia reversa de projetos Java. O NetBeans UML Project já vem integrado no IDE NetBeans desde a versão 6.0. A versão utilizada nos testes foi a 6.1. A ferramenta recuperou a estrutura de pacotes,

classes, métodos e atributos do código do SIGA-EDU, e também a

implementação de classe e relacionamentos de agregação e generalização.

A ferramenta não gera automaticamente os diagramas, tendo que arrastar os elementos para um novo para poder visualizá-lo. O *plug-in* UML também não dá suporte ao formato de arquivo XMI. Porém, um diferencial da ferramenta é a opção de *plug-ins* para controladores de versão como o CVS (*Concurrent Version System*) e Subversion, podendo-se baixar o código que passará pelo processo de engenharia reversa diretamente de seu repositório. Outro diferencial é a opção de geração de relatório. A ferramenta funciona em qualquer sistema operacional que possua máquina virtual Java.

3.4. Umbrello

O Umbrello UML Modeler é um software de modelagem UML que faz parte do projeto KDE. Atualmente é desenvolvida e mantida por um grupo de programadores de diferentes partes do mundo. É licenciado pela licença GNU e é suportada na plataforma GNU/Linux. O Umbrello também gera código automaticamente nas linguagens Java, PHP, JavaScript, ActionScript, C++, SQL, Ada, IDL, XMLSchema, Python, Perl e Ruby. Além disso, também gera arquivos gráficos do tipo PNG, realiza a engenharia reversa de classes, exporta arquivos no padrão XMI, dentre outras funcionalidades.

O Umbrello utilizado nos testes encontra-se na versão 2.0. Na importação do código, foi possível identificar a estrutura de pacotes, classes, métodos, atributos e interfaces. Em relação a relacionamentos recuperados, foi possível identificar composição e associação simples bidirecional.

A ferramenta, como as anteriormente citadas, não gera automaticamente diagramas a partir da importação. Para isso, deve-se arrastar os elementos para um novo diagrama, para que seja possível visualizar seus relacionamentos. Apesar de esta ferramenta funcionar apenas em ambientes KDE, um ponto a seu favor é o suporte oferecido a arquivos XMI.

4. ANÁLISE DOS RESULTADOS OBTIDOS

Após a realização dos testes, foram verificados e comparados os tipos de elementos e relacionamentos recuperados a partir da engenharia reversa em cada ferramenta. As ferramentas analisadas obtiveram resultados parecidos em termos de quantidade de elementos UML e relacionamentos recuperados na engenharia reversa do código do sistema SIGA-EDU.

Em relação a isso, o diferencial entre elas acabou sendo a identificação de algum elemento ou relacionamento por uma delas que outra não pôde identificar. Isto demonstra que, embora haja no mercado diversas ferramentas para realizar esta atividade, nenhuma pode ser considerada “perfeita”. Mesmo que numa ferramenta se mostre superior a outra no conjunto de resultados, esta outra pode se apresentar melhor em um resultado específico.

A Tabela 1 mostra os elementos UML recuperados por cada ferramenta analisada. Conforme colocado anteriormente, para exemplificar a classificação relativa entre as ferramentas, cite-se o exemplo de Jude Community, que apresentou um total de oito tipos de elementos diferentes identificados no processo de engenharia reversa, porém não identificou o elemento “componente” identificado pela ferramenta ArgoUML.

Tabela 1: Elementos UML recuperados pelas ferramentas CASE do código do sistema SIGA-EDU

ELEMENTOS DO CÓDIGO SIGA-EDU RECUPERADOS PELA ENGENHARIA REVERSA										
FERRAMENTA	Classe	Atributo	Método	Interface	Componente	List	Set	Map	Collection	Total de Elementos recuperados
ArgoUML	X	X	X	X	X	-	-	-	-	5
Jude Community	X	X	X	X	-	X	X	X	X	8
NetBeans	X	X	X	X	-	-	-	-	-	4
Umbrello	X	X	X	X	-	-	-	-	-	4

Legenda: X (elemento recuperado pela engenharia reversa);

- (elemento não recuperado pela engenharia reversa).

A Tabela 2 mostra os relacionamentos entre os elementos identificados no processo de engenharia reversa realizado por cada uma das ferramentas analisadas, sobre o código do sistema SIGA-EDU.

Tabela 2: Relacionamentos recuperados pelas ferramentas CASE do código do sistema SIGA-EDU

RELACIONAMENTOS DO CÓDIGO SIGA-EDU RECUPERADOS PELA ENGENHARIA REVERSA								
FERRAMENTA	Associação Simples Unidirecional	Associação Simples Bidirecional	Generalização / Herança	Agregação	Composição	Abstração / Implementação de classe	Dependência	Total
ArgoUML	-	-	X	-	-	X	X	3
Jude Community	X	X	X	-	-	-	-	3
NetBeans	-	-	X	X	-	X	-	3
Umbrello	-	X	-	-	X	-	X	3

Legenda: X (elemento recuperado pela engenharia reversa);

- (elemento não recuperado pela engenharia reversa).

Na Tabela 2, pode-se observar que todas as ferramentas obtiveram a mesma quantidade de tipos de relacionamentos identificados na engenharia reversa. No total dos testes, as ferramentas juntas identificaram sete tipos diferentes de relacionamento, mas individualmente, cada uma identificou um total de três.

Para que uma escolha de uma ferramenta seja feita com uma certa margem de satisfação, a Tabela 3 apresenta um quadro comparativo criado através dos critérios estabelecidos anteriormente na metodologia, auxiliando na recomendação da seleção.

Tabela 3: Comparativo entre as ferramentas CASE testadas

FERRAMENTA	AVALIAÇÃO				
	Quantidade de tipos diferentes de elementos recuperados na engenharia reversa do código	Quantidade de tipos diferentes de Relacionamentos recuperados na engenharia reversa do código	Suporte ao formato de arquivo XMI	Portabilidade da ferramenta	Total de pontos
<i>ArgoUML</i>	5	3	1	1	10
<i>Jude Community</i>	8	3	0	1	12
<i>NetBeans</i>	4	3	0	1	8
<i>Umbrello</i>	4	3	1	0	8

Legenda da Pontuação:

Cada tipo de elemento e relacionamento diferente recuperado pelas ferramentas equivale a 1 (um) ponto;

O suporte ao formato XMI equivale a 1 (um) ponto;

A portabilidade do sistema equivale a 1 (um) ponto.

5. CONCLUSÃO

Este artigo apresentou uma pesquisa sobre ferramentas CASE para engenharia reversa estática, de acordo com alguns critérios definidos na norma ISO/IEC 14102:2008. Com este trabalho, pôde-se perceber que não existem ferramentas de código aberto e/ou livres de análise estática que proporcionem condições para o trabalho de engenharia reversa de forma mais simples.

Para que se possam obter resultados mais completos ou precisos da engenharia reversa de um sistema, talvez haja a necessidade da utilização de mais de uma ferramenta. Isso aponta para um campo que pode ser melhor explorado, em relação ao desenvolvimento de uma ferramenta livre que recupere os vários elementos UML e relacionamentos de um código, que importe e exporte no formato XMI, e que seja um sistema portátil, proporcionando maior liberdade no seu uso.

REFERÊNCIAS

ArgoUML. <http://argouml.tigris.org>.

ISO/IEC 14102:2008, “Information technology – Guideline for the evaluation and selection of CASE tools”. <http://www.iso.org>.

Jude Community. <http://jude.change-vision.com>.

NetBeans. <http://www.netbeans.org>.

Oman, P. W. e Cook, R. C., “The book Paradigm for Improved Maintenance”. IEEE Software, vol. 7, n. 1, janeiro de 1990.

Pressman, R.S., “Engenharia de Software”, 6a. ed., McGraw Hill. New York, 2001.

Projeto SIGA-EPT. <http://www.sigaept.org>.

Umbrello. <http://uml.sourceforge.net>.

Weinrich, J., “Software de Apoio à Avaliação e Seleção de Ferramentas CASE Baseado na Norma ISO/IEC 14102”. Trabalho de Conclusão de Curso, Universidade Regional de Blumenau. Blumenau, junho de 1999.