

Métodos Formais e Engenharia Reversa aplicados à Manutenção e Usabilidade de Sistemas Interactivos

J. C. Silva*

Departamento de Informática, Universidade do Minho
Braga, Portugal

SDDI'2006
III Simpósio Doutoral do DI

22, 23 e 24 de Fevereiro de 2006

*Doutorando em Informática, área de conhecimento em Fundamentos da Computação.

Conteúdo

1	Identificação	3
2	Resumo da Tese de Doutoramento	3
2.1	Área de Investigação e Desenvolvimento (I&D)	3
2.2	Resumo	3
2.3	Objectivos Estratégicos	4
3	Enquadramento	4
3.1	Enquadramento Científico	4
3.2	Motivação	5
3.3	Objectivos Detalhados	6
3.4	Trabalhos Alternativos	7
3.5	Bibliografia Principal	7
4	Macro-planeamento das Actividades	9
4.1	Fase 1: Estudo das Características dos Sistemas Interactivos e Metodologias de Desenvolvimento Associadas	9
4.2	Fase 2: Desenvolvimento da Tese	10
4.3	Fase 3: Caso de Estudo	11
4.4	Fase 4: Conclusões e Revisão do Texto da Tese	12
5	Contribuições	12
5.1	Um Modelo e uma Arquitectura Aplicacional	12
5.1.1	O Modelo	12
5.1.2	A Arquitectura applicacional	14
5.2	Relacionando todos os objectos gráficos	14
5.3	Caso de Estudo: JTurma	15
5.4	Publicação	17

1 Identificação

- **Nome:** João Carlos Cardoso da Silva
- **E-mail:** *jcsilva@ipca.pt*
- **Orientadores:**
 - Prof. João Alexandre Saraiva
 - Universidade do Minho, Departamento de Informática
 - *jas@di.uminho.pt*
 - Prof. José Creissac Campos
 - Universidade do Minho, Departamento de Informática
 - *jose.campos@di.uminho.pt*
- **Data Início:** Abril 2005
- **Data Término Prevista:** Abril 2008

2 Resumo da Tese de Doutoramento

2.1 Área de Investigação e Desenvolvimento (I&D)

A área de investigação e desenvolvimento (I&D) desta tese de Doutoramento enquadra-se na área de conhecimento em Fundamentos da Computação.

2.2 Resumo

O trabalho que se propõe efectuar neste projecto de doutoramento consiste em definir uma abordagem para a aplicação de métodos formais no contexto da engenharia reversa de sistemas interactivos. Assim, vai ser explorada a possibilidade da adopção de métodos formais na engenharia reversa de sistemas interactivos de forma a validar posteriormente a usabilidade do sistema aquando da sua adaptação e evolução.

2.3 Objectivos Estratégicos

A manutenção e evolução dos sistemas interactivos mantendo um elevado nível de usabilidade constituem problemas importantes que se levantam hoje em dia ao engenheiro de software. Pelas suas características este tipo de sistema é bastante vulnerável aos problemas referidos. As metodologias e técnicas actuais não abordam de forma satisfatória estes graves problemas. Os métodos formais têm merecido elevada atenção da comunidade científica. A sua aplicação na tentativa de resolução de diversos problemas tem resultado num número significativo de experiências positivas. Com esta proposta de trabalho pretende-se validar a tese de que a aplicação de métodos formais no processo de engenharia reversa de sistemas interactivos (suportadas por um modelo adequado) permite melhorar significativamente o grau de flexibilidade e suporte à manutenção e evolução do sistema. Com o plano de trabalho apresentado pretende-se atingir esse objectivo de forma rigorosa e credível. Acredita-se que o texto resultante da elaboração desta tese se poderá constituir numa referência original e significativa nesta área científica.

3 Enquadramento

3.1 Enquadramento Científico

Um dos problemas que afectam o sucesso dos projectos de engenharia de software que envolvam uma componente de interacção com o utilizador é a usabilidade [Cam04]. Por outro lado, hoje em dia é comum falar-se dos custos na manutenção de sistemas interactivos. Acontecem sempre situações em que existem problemas com software já desenvolvido. Por exemplo, os requisitos iniciais estão em constante mudança o que implica a alteração do software.

Nos primeiros desenvolvimentos de soluções informáticas, a disciplina da engenharia de software concebia o processo de desenvolvimento dos sistemas através de metodologias pouco flexíveis e orientadas fundamentalmente para projectos novos. Rapidamente se observou que os sistemas informáticos têm um período de vida bastante extenso, sendo necessário adaptar os referidos sistemas às diversas evoluções, tanto funcionais como tecnológicas. A engenharia de software tem respondido a estas necessidades adaptando e promovendo metodologias e processos que visam o desenvolvimento de soluções informáticas num processo em espiral [Boe88], tentando aumentar a rapi-

dez com que são incorporados novos requisitos (ou alterações de requisitos) nas soluções informáticas. Segundo [eES80], 70 por cento do esforço de programação concentra-se na manutenção.

Nestas situações a utilização de engenharia reversa pode contribuir para a resolução de tais problemas [MBN03, ESS03]. Tendo em vista explorar este aspecto, pretende-se com este projecto de Doutoramento desenvolver metodologias e ferramentas baseadas em métodos formais para a engenharia reversa de sistemas interactivos. No âmbito do projecto IVY, o trabalho desenvolvido neste projecto de Doutoramento será aplicado no sentido de avaliar a usabilidade de sistemas interactivos legados.

3.2 Motivação

Hoje em dia é comum falar-se da complexidade e dos custos no desenvolvimento e na manutenção de sistemas que envolvam uma componente de interacção com o utilizador, i.e. os sistemas interactivos. Os custos referidos não se medem simplesmente em tempo e dinheiro. A utilização de sistemas interactivos em áreas onde a segurança é um factor crítico, por exemplo no ramo automóvel ou na aviação, implica que as consequências da má qualidade podem chegar, em casos extremos, à perda de vidas humanas. Torna-se então importante assegurar a qualidade dos sistemas interactivos desde cedo no processo de desenvolvimento. O sucesso de tais sistemas depende muito da sua usabilidade. Vários estudos têm sido realizados nesta área [dSGD98, BA95]. A norma ISO DIS 9241-11 define a usabilidade de um sistema como a eficácia, eficiência e satisfação com que utilizadores determinados atingem objectivos determinados em ambientes específicos. Eficácia diz respeito a possibilidade (ou não) de o utilizador poder atingir os seus objectivos utilizando o sistema num dado contexto. Eficiência tem a ver com o maior ou menor esforço que o utilizador deverá aplicar para atingir esse objectivo. Satisfação é uma medida subjectiva do grau de agradabilidade na utilização do sistema. Avaliar a qualidade de um sistema interactivo implica avaliar a facilidade de utilização do mesmo. Esta tarefa complica-se dada a necessidade de incluir considerações sobre factores-humanos no processo de avaliação. No desenvolvimento de sistemas interactivos é, portanto, necessário cruzar duas comunidades: a Interacção Humano-Computador (IHC) e a Engenharia do Software [CH03]. As metodologias de desenvolvimento de software ainda estão bastante limitadas a este nível.

O projecto IVY, no qual se enquadra este projecto de Doutoramento, tem

como objectivo desenvolver uma ferramenta que suporta uma abordagem ao desenvolvimento de sistemas interactivos em que se procura facilitar a comunicação entre as duas comunidades. O trabalho já realizado na abordagem é baseada em modelos e pretende possibilitar aos engenheiros de software uma maior autonomia na consideração de aspectos de usabilidade relacionados com o comportamento do sistema, bem como identificar os pontos em que é necessário recorrer ao auxílio de peritos em IHC.

3.3 Objectivos Detalhados

As abordagens clássicas desenvolvidas para os problemas apresentados propõem intervenções ao nível das metodologias e das ferramentas de forma a validarem a usabilidade e suportarem uma mais fácil e rápida incorporação de alterações.

O trabalho que se propõe efectuar neste projecto de doutoramento consistirá em definir uma abordagem para a aplicação de métodos formais no contexto da engenharia reversa de sistemas interactivos. Assim, vai ser explorada a possibilidade da adopção de métodos formais na engenharia reversa de sistemas interactivos de forma a validar posteriormente a usabilidade do sistema aquando da sua adaptação e evolução.

Os objectivos principais deste projecto de Doutoramento são:

- definir um modelo para a engenharia reversa de sistemas interactivos;
- desenvolver a tecnologia e ferramentas de suporte ao modelo;
- validar o modelo e a tecnologia propostos através de casos de estudo baseados em soluções reais.

Como caso de estudo pretende-se estudar a aplicação de engenharia reversa a código Java2/Swing [LEW⁺02]. Um aspecto que interessa referir prende-se com o nível de abstracção a que se pretende trabalhar. O objectivo não é obter um modelo da estrutura do código que implementa a interface. Quais as classes/objectos envolvidas(os) e suas relações. Em vez disso, o que se pretende é obter um modelo da interface que esse código implementa. Que informação está presente na interface, e que acções podem ser executadas sobre ela. Trata-se, portanto, de um salto de abstracção maior que o necessário para obter um modelo do código. O que se pretende deduzir por análise do código, não é quais os objectos presentes na interface (campos

de texto, botões, etc.) e como estão organizados, mas antes que informação está presente na interface (o número, o nome, etc.) e as acções disponíveis na mesma (adicionar, consultar, etc.). O processo de engenharia reversa assenta num certo número de suposições e heurísticas que permitem tornar o processo viável. Estas abordam basicamente dois aspectos: aspectos relacionados com a arquitectura da aplicação a reverter e aspectos relacionados com o nível de abstracção a utilizar.

Entre as muitas questões que irão surgir, alguns dos problemas envolvidos são:

- Como reverter o código para um nível adequado de abstracção e como identificar esse nível adequado de abstracção;
- Como modelar a interacção;
- Como modelar interfaces com elevado grau de dinamismo.

3.4 Trabalhos Alternativos

Nos últimos anos, vários autores investigaram a usabilidade dos sistemas interactivos. Várias técnicas têm sido apresentadas. Por exemplo, Chen e Subramaniam [CS01] desenvolveram uma técnica baseada num modelo para testar as interfaces gráficas. O modelo permite manipular testes representados por máquinas de estados finitas.

Moore [Moo96] descreve uma técnica para parcialmente automatizar o processo de engenharia reversa de sistemas interactivos. O resultado deste processo é um modelo para testar a funcionalidade das interfaces gráficas. O trabalho desenvolvido permite com base num conjunto de regras detectar componentes interactivos de código legado.

3.5 Bibliografia Principal

A seguir apresenta-se uma listagem das principais referências bibliográficas que suportam o trabalho de Doutoramento.

Referências

- [BA95] Peter Bumbulis and P.S C. Alencar. A framework for prototyping and mechanically verifying a class of user interfaces. *IEEE*, 1995.

- [Boe88] B. Boehm. *A Spiral Model for Software Development and Enhancement*, volume 21, pages 61–72. Computer edition, Maio 1988.
- [Cam04] José C. Campos. The modelling gap between software engineering and human-computer interaction. In Rick Kazman, Len Bass, and Bonnie John, editors, *ICSE 2004 Workshop: Bridging the Gaps II*, pages 54–61. The IEE, 2004.
- [CH03] J. C. Campos and M. D. Harrison. From HCI to Software Engineering and back. In Rick Kazman, Len Bass, and Jan Bosch, editors, *Bridging the Gaps Between Software Engineering and Human-Computer Interaction, ICSE '2003 workshop*, pages 49–56, Portland, Oregon, USA, May 2003. IFIP.
- [CS01] J. Chen and S. Subramaniam. A gui environment for testing gui-based applications in java. *Proceedings of the 34th Hawaii International Conferences on System Sciences*, january 2001.
- [dSGD98] Bruno dAusbourg, Christel Seguin, and Pierre Rochk Guy Durrieu. Helping the automated validation process of user interfaces systems. *IEEE*, 1998.
- [eES80] B. Lientz e E. Swanson. *Software Maintenance Management*. Addison-wesley edition, 1980.
- [ESS03] P. Iglinski E. Stroulia, M. El-ramly and P. Sorenson. User interface reverse engineering in support of interface migration to the web. *Automated Software Engineering*, 2003.
- [LEW⁺02] Marc Loy, Robert Eckstein, Dave Wood, James Elliott, and Brian Cole. *Java Swing, 2nd Edition*. O'Reilly, 2002.
- [MBN03] Atif Memon, Ishan Banerjee, and Adithya Nagarajan. GUI ripping: Reverse engineering of graphical user interfaces for testing. Technical report, Department of Computer Science and Fraunhofer Center for Experimental Software Engineering, Department of Computer Science University of Maryland, USA, 2003.
- [Moo96] M. M. Moore. Rule-based detection for reverse engineering user interfces. *Proceedings of the Third Working Conference on Reverse Engineering*, pages 42-8, Monterey, CA, november 1996.

4 Macro-planeamento das Actividades

Para atingir os objectivos enunciados nesta proposta de doutoramento prevê-se um plano de trabalho que inclui as fases seguintes:

4.1 Fase 1: Estudo das Características dos Sistemas Interactivos e Metodologias de Desenvolvimento Associadas

1. Estado da Arte do Desenvolvimento de Sistemas Interactivos:

Início: Abril de 2005

Duração: 3 meses

Nesta fase, pretende-se efectuar um estudo sobre as características fundamentais dos sistemas interactivos. Assim, prevê-se, numa primeira etapa, efectuar um levantamento das características funcionais e tecnológicas deste tipo de sistemas, assim como das metodologias e ferramentas mais comuns que são utilizadas para o seu desenvolvimento. Prevê-se que esta primeira etapa necessitará de 3 meses para estar completa. Numa segunda etapa desta fase, desenvolver-se-á o estudo das eventuais especificidades dos sistemas interactivos, visto ser neste âmbito que se pretende validar o trabalho desenvolvido nesta tese. Este estudo incluirá as aplicações em Java2/Swing (que serão a base dos casos de estudo) assim como as metodologias, técnicas e tecnologias que suportam actualmente o desenvolvimento dos referidos sistemas. Esta fase permitirá, por um lado, efectuar o levantamento das características do sistema interactivo e das metodologias utilizadas no seu desenvolvimento (incluindo manutenção e evolução), por outro lado, tipificar a manutenção e evolução que normalmente ocorre neste sistema, a forma como é efectuada, os tempos de resposta obtidos e o grau de suporte à mudança dos referidos sistemas. O material resultante desta segunda fase permitirá a elaboração de um capítulo sobre sistemas interactivos que será elaborado tendo como princípio orientador o suporte à usabilidade deste tipo de sistema.

2. **Estado da Arte da Engenharia Reversa de Sistemas Interactivos e respectiva Usabilidade:**

Início: Julho de 2005

Duração: 3 meses

Esta fase consistirá num levantamento dos desenvolvimentos mais recentes e significativos no contexto da engenharia reversa de sistemas interactivos assim como na usabilidade de tais sistemas. Após o levantamento inicial do estado da arte segue-se o estudo detalhado das técnicas e tecnologias mais significativas encontradas. Pretende-se seleccionar as tecnologias que mais se aproximarem conceptualmente do âmbito deste trabalho. O material resultante desta primeira fase permitirá a elaboração de um capítulo sobre o estado da arte da engenharia reversa de sistemas interactivos e respectiva usabilidade.

3. **Estado da Arte dos *FrontEnd* para a análise e transformação de código de Sistemas Interactivos:**

Início: Outubro de 2005

Duração: 3 meses

Nesta fase, pretende-se efectuar um estudo sobre os *Front End* existentes na área da análise e transformação de código em Sistemas Interactivos. Será também estudado a utilização do sistema STRAFUNSKI na análise e transformação de código Java2/Swing.

4.2 Fase 2: Desenvolvimento da Tese

Início: Janeiro de 2006

Duração: 3+6+12 meses

Esta fase consiste no núcleo da elaboração do trabalho da tese. Durante esta fase pretende-se desenvolver o princípio de que a engenharia reversa de sistemas interactivos permite aumentar o suporte à manutenção e evolução

deste tipo de software, avaliando no contexto do projecto IVY a sua usabilidade. Para desenvolver este princípio, esta fase consistirá em três etapas.

1. Utilização de Metodologias para a Manutenção e Evolução de Sistemas Interactivos:

Caracterizar com grande detalhe os problemas existentes actualmente no desenvolvimento, manutenção e evolução de sistemas interactivos. Justificar de que forma a engenharia reversa quando utilizada nos sistemas interactivos pode diminuir significativamente (ou resolver) este tipo de problemas. Resultará desta etapa a elaboração de um capítulo sobre a utilização de engenharia reversa em sistemas interactivos. Estima-se a necessidade de 3 meses para a conclusão desta etapa.

2. Utilização de Técnicas de Engenharia Reversa:

Estudo detalhado de técnicas de engenharia reversa e da sua utilização em sistemas interactivos. O estudo detalhado das características técnicas e funcionais desta metodologia servirá de suporte ao modelo a propor para a engenharia reversa de sistemas interactivos. Resultará desta etapa a elaboração de um capítulo sobre a engenharia reversa de sistemas interactivos e, eventualmente, a elaboração de um artigo científico a publicar em revista ou conferência. Estima-se a necessidade de 6 meses para a conclusão desta etapa.

3. Um modelo e uma Arquitectura Aplicacional:

O objectivo da última etapa desta fase é a elaboração da justificação teórica da tese que se defende, ou seja, um modelo e uma arquitectura aplicacional que suportem a engenharia reversa de sistemas interactivos. O resultado desta etapa poderá servir de base para um artigo científico a publicar em revista ou conferência expondo o modelo defendido na tese. Este modelo deverá ser validado nos casos de estudo previstos para a fase seguinte. A utilização da linguagem Java2/SWING torna possível a validação do modelo proposto com base em aplicações e casos de estudo concretos de sistemas interactivos. Estima-se a necessidade de 12 meses para a conclusão desta etapa.

4.3 Fase 3: Caso de Estudo

Início: Outubro de 2007

Duração: 4 meses

Esta fase consistirá, fundamentalmente, na experimentação do modelo desenvolvido na fase anterior com base na utilização de sistemas interactivos desenvolvidos em Java2/Swing. O objectivo será validar e demonstrar na prática os princípios que a tese defende. O resultado desta fase poderá, eventualmente, ser alvo de um artigo científico a publicar em revista ou conferência expondo as conclusões retiradas destes casos de estudo. O material resultante desta fase permitirá a elaboração de um capítulo da tese dedicado aos casos de estudo.

4.4 Fase 4: Conclusões e Revisão do Texto da Tese

Início: Janeiro de 2008

Duração: 3 meses

Esta fase estará dividida em duas etapas. Numa primeira etapa, com uma duração esperada de dois meses, será efectuada uma reflexão sobre o trabalho desenvolvido. Esta reflexão permitirá efectuar uma análise crítica sobre o trabalho desenvolvido, tirando conclusões e avançando com hipóteses de seguimento do trabalho. A segunda etapa desta fase será o culminar do trabalho da tese, consistindo essencialmente na revisão do texto da tese. Prevê-se uma duração de 1 mês para esta etapa.

5 Contribuições

5.1 Um Modelo e uma Arquitectura Aplicacional

As contribuições técnico-científicas consistem num modelo e numa arquitectura aplicacional que suportem a engenharia reversa de sistemas interactivos desenvolvidos na linguagem Java2/Swing através do ambiente integrado de desenvolvimento NETBEANS.

5.1.1 O Modelo

O protótipo especificado neste trabalho de Doutoramento tem como ponto de partida código fonte JAVA2/SWING de sistemas interactivos desenvolvidos através do ambiente integrado de desenvolvimento NETBEANS. Tal como referido anteriormente, existem três objectivos essenciais que o protótipo deve satisfazer:

1. Engenharia reversa do código para um nível adequado de abstracção;
2. Modelação de interacção;
3. Modelação de interfaces com elevado grau de dinamismo.

O passo inicial consiste na obtenção de um *parser* para a linguagem JAVA2/SWING pelo que foram utilizadas várias funcionalidades disponibilizadas através da UMINHOHASKELLLIBRARY¹, de entre as quais destacam-se as seguintes:

- Gramática no formato SDF da linguagem JAVA2/SWING;
- Geração de um *parser* da linguagem JAVA2/SWING através da ferramenta SGLR aplicando-lhe a gramática anterior;
- A partir da gramática e do parser, geração automática de um conjunto de tipos de dados abstractos em HASKELL (*CompilationUnit*) para a representação em árvore de código JAVA2/SWING (aplicação da biblioteca SDF2HASKELL);
- A partir do PARSE e dos tipos abstractos de dados, conversão de código JAVA2/SWING para uma representação em árvore. Cada definição de classes na linguagem de programação JAVA2/SWING dá assim origem a uma expressão do tipo *CompilationUnit* (aplicação da biblioteca HATERM);
- Travessia das árvores referidas no ponto anterior e extracção de sub-expressões com base em padrões pre-definidos (*Slicing*) (aplicação da biblioteca STRATEGYLIB);

A aplicação de algumas funcionalidades da UMINHOHASKELLLIBRARY permite deste modo obter uma plataforma funcional para a extracção e manipulação de quaisquer expressões contidas em código JAVA2/SWING. Sobre este modelo, foi desenvolvido um protótipo o qual permite reverter código JAVA2/SWING para um nível adequado de abstracção.

¹Conjunto de bibliotecas desenvolvidas na linguagem de programação HASKELL pelo Departamento de Informática da Universidade do Minho

5.1.2 A Arquitectura applicacional

O protótipo foi desenvolvido na linguagem de programação funcional HASKELL e através da plataforma explanada no ponto anterior, permite realizar SLICING da árvore de código JAVA2/SWING, possibilitando assim extrair sub-partes da árvore.

Dado que o protótipo aplica-se a aplicações JAVA2/SWING desenvolvidas com o NETBEANS, então para cada classe o protótipo extrai e manipula em primeiro lugar o método INITCOMPONENTS o qual define todos os componentes gráficos da aplicação.

A seguir protótipo extrai do respectivo método toda a informação relacionada com a parte gráfica da aplicação em estudo. Por exemplo objectos das classes *JButton*, *JLabel*, *JTextField*, *JProgressBar*, *JComboBox*, *JSlider*, etc. Para cada tipo de objecto extrai-se somente a informação relevante para este processo de engenharia reversa.

5.2 Relacionando todos os objectos gráficos

Após extrair os dados dos objectos gráficos assim como todos os dados dos métodos sobre estes executados, é então possível gerar de forma automática um grafo de fluxo de dados entre estes objectos.

Nesse sentido optou-se por utilizar a ferramenta GRAPHVIZ para a geração do grafo de fluxos. Em GRAPHVIZ, os grafos são elaborados com base numa linguagem própria pelo que foi desenvolvido um módulo auxiliar responsável por produzir o código necessário na linguagem GRAPHVIZ.

Toda a informação traduzida para a linguagem GRAPHVIZ é obtida com base nos resultados dos *slices* da secção anterior. Assim, cada objecto gráfico extraído da árvore dá origem a um vértice no grafo e cada invocação de métodos sobre esses objectos gráficos dá origem a uma aresta. Acrescentam-se também no grafo outros dados específicos do código fonte como por exemplo nomes de variáveis, parâmetros, etc.

De forma a facilitar a visualização do grafo, foram atribuídas cores diferentes aos vários tipos de vértices e arestas do grafo.

O modelo e a arquitectura aqui apresentados definem assim uma abordagem para a aplicação de métodos formais no contexto da engenharia reversa de sistemas interactivos desenvolvidos em Java2/Swing via NETBEANS IDE. A seguir pretende-se descrever a aplicação desta metodologia através de um caso de estudo concreto.

5.3 Caso de Estudo: JTurma

A experimentação do modelo e da arquitectura applicacional sobre um pequeno sistema interactivo desenvolvido com o NETBEANS IDE em Java2/Swing permite validar e demonstrar na prática os princípios que a tese defende, isto é, um modelo e uma arquitectura applicacional que suportem a engenharia reversa de sistemas interactivos desenvolvidos através do ambiente integrado de desenvolvimento NETBEANS. A aplicação em estudo designa-se por JTURMA (cf. Figura 1).

Esta aplicação permite armazenar e manipular as classificações teóricas e práticas de uma turma. O sistema disponibiliza uma funcionalidade que permite adicionar no sistema os números, os nomes, as classificações teóricas e as classificações práticas de todos os alunos da turma. Para além do armazenamento dos dados, encontram-se disponíveis funções para consultar ou remover os dados de um aluno bem como uma função para terminar a aplicação. No desenvolvimento desta aplicação foram utilizados vários objectos gráficos JAVA2/SWING, nomeadamente objectos das classes JBUTTON, JTEXTFIELD, JLABEL, JPROGRESSBAR, JCOMBOBOX, JSLIDER, JPANEL e GETCONTENTPANE.



Figura 1: Aplicação *JTurma*

A execução do protótipo sobre o sistema JTURMA permite gerar de forma automática um grafo de fluxo de dados entre todos os seus objectos gráficos (cf. Figura 2).

Através dos resultados obtidos, demonstra-se que a abordagem descrita na secção anterior, ainda que actualmente com muitas limitações, suporta a engenharia reversa de sistemas interactivos desenvolvidos através do NETBEANS.

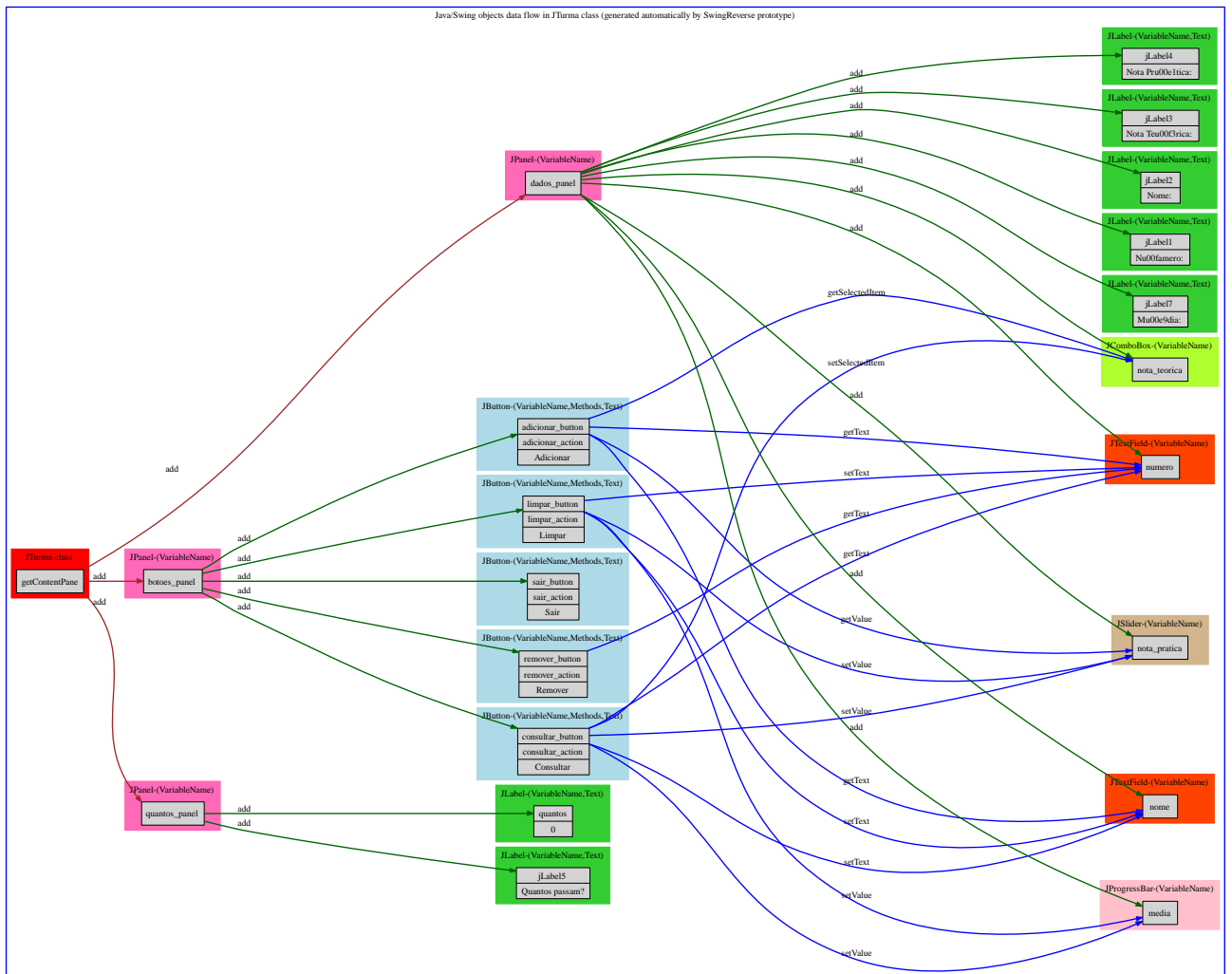


Figura 2: Grafo de fluxos de dados entre os objectos gráficos da aplicação *JTurma*

5.4 Publicação

Publicação a submeter brevemente:

- **Título:** Combining Formal Methods and Functional Strategies Regarding the Reverse Engineering of Interactive Applications.
- **Data:** Fevereiro 2006
- **Resumo:** *Graphical user interfaces (GUIs) make software easy to use by providing the user with visual controls. Therefore, correctness of GUI's code is essential to the correct execution of the overall software. Models can help in the evaluation of interactive applications by allowing designers to concentrate on its more important aspects. This paper describe our approach to reverse engineering a abstract model, represented as an event-flow graph directly, from the GUI's legacy code. We also present results of a case study which are encouraging and give evidence that the goals of reverse engineering user interfaces could be met with more work on this technique.*