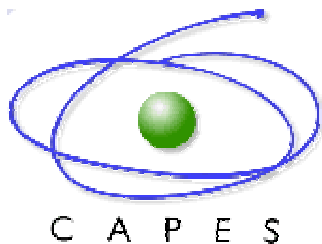


Orientação a Objeto: Definição e Análise de Recursos de Teste e Validação

José Carlos Maldonado (ICMC-USP)

Auri Marcelo Rizzo Vincenzi (ICMC-USP)

Márcio Eduardo Delamaro (FIN-UNIVEM)



Organização

- Introdução
- Objetivo
- Atividades e Resultados Anteriores
- Atividades e Resultados no Período
- Atividades Futuras

Introdução

- Teste de Software
 - Objetivo:
 - Executar um programa com a intenção de identificar a presença de erros
 - Teste de Programas Procedimentais
 - Critérios Funcionais, Estruturais, Baseado em Erros e Baseado em Estados
 - Ferramentas de Apóio
 - Estudos Teóricos e Empíricos
 - Teste de Programas OO
 - Maioria dos Critérios Baseado em Estados
 - Não revelam todos os tipos de erros
 - Poucas Ferramentas de Teste
 - Poucos Estudos Comparativos

Objetivo

- Avaliar a adequação de Critérios Procedimentais no Teste de Programas OO
 - Critérios Baseados em Mutação
 - Análise de Mutantes (Teste de Unidade)
 - Mutação de Interface (Teste de Integração)
 - Critérios Baseados em Análise de Fluxo de Dados
 - Critérios Potenciais-Usos
- Desenvolvimento de um Ambiente Integrado de Teste e Validação
- Realização de Estudos Empíricos

Atividades e Resultados Anteriores

- Definição de Critérios de Teste
 - Mutação de Interface
 - Potenciais-Usos e Potenciais-Usos de Integração (Unicamp)
- Desenvolvimento de Ferramentas
 - Proteum & Proteum/IM
 - PokeTool (Unicamp)
- Realização de Estudos Comparativos
 - Comparação entre Análise de Mutantes e Potenciais-Usos
 - Determinação de Conjuntos Essenciais de Operadores de Mutação
 - Comparação entre Análise de Mutantes e Mutação de Interface

Atividades e Resultados Anteriores

- Trabalhos Desenvolvidos
 - 3 Teses de Doutorado
 - 5 Dissertações de Mestrado
 - Trabalhos de Iniciação Científica
- Publicações (2001/2003)

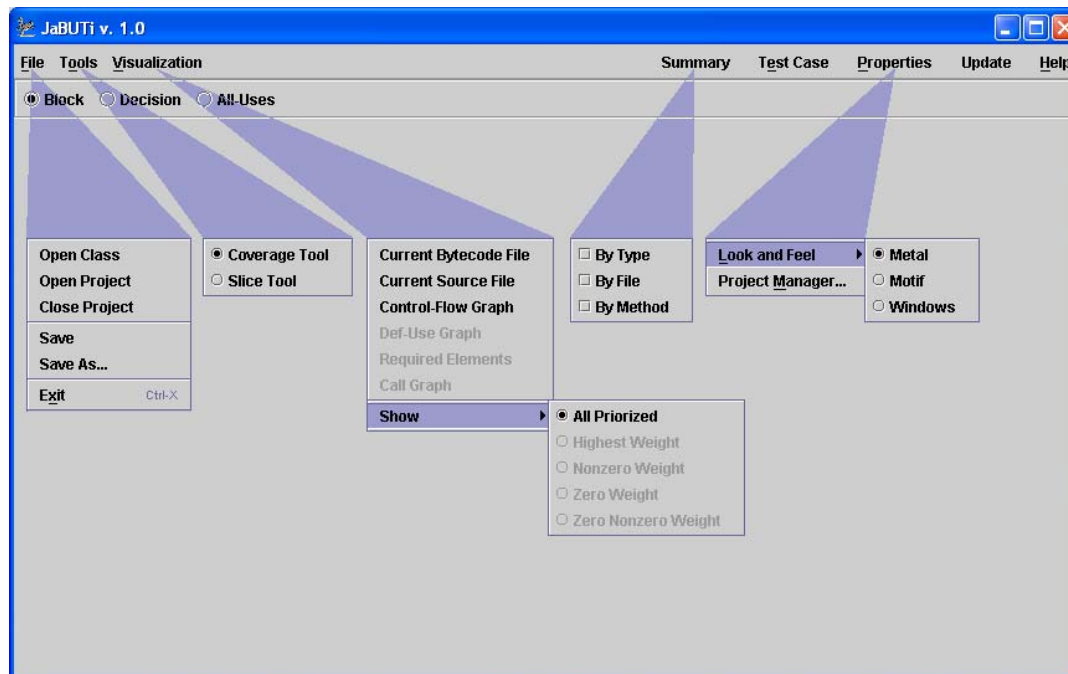
	Nacionais	Internacionais
Congressos	5	6
Revistas	-	3
Relatórios Técnicos	4	

Atividades e Resultados no Período

- Identificação de Técnicas e Critérios de Teste para Programas OO e Componentes
- Avaliação de Ferramentas de Teste para Programas OO
- Escrita de um Capítulo sobre Teste de Programas OO e de Componentes (versões em inglês e português)
- Artigo para o UK-SOFTTEST'2003 “Java Bytecode Static Analysis: Deriving Structural Testing Requirements”
- Artigo para a Sessão de Ferramentas do SBES'2003: “JaBUTi: A Coverage Analysis Tool for Java Programs”

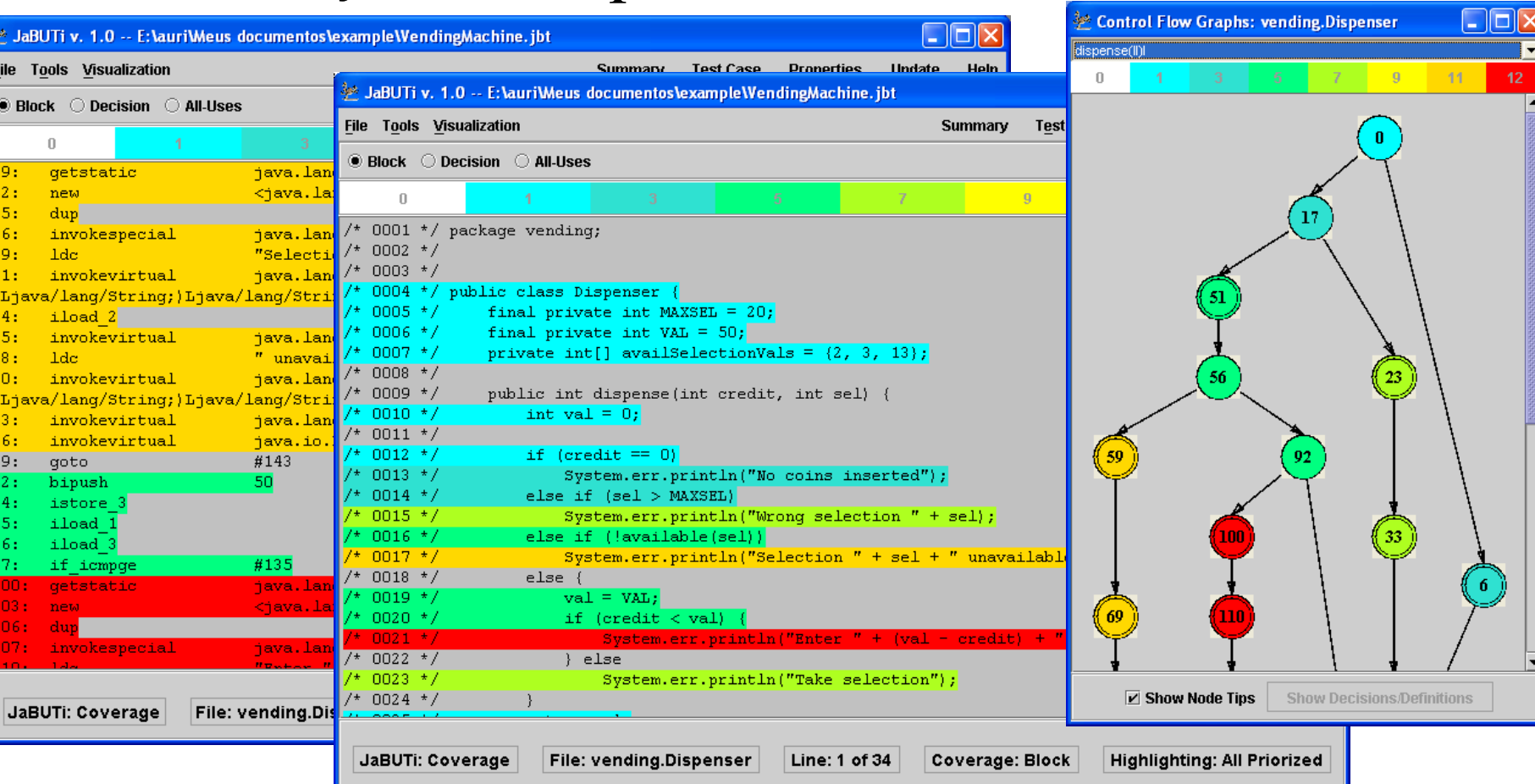
Atividades e Resultados no Período

- Implementação da ferramenta JaBUTi (Java Bytecode Understanding and Testing)
 - Teste estrutural de programas/componentes Java



Atividades e Resultados no Período

- Priorização de requisitos de teste



The screenshot displays the JaBUTi v. 1.0 interface, which is used for testing and coverage analysis. The main window shows the source code of the `vending.Dispenser` class, with lines of code color-coded to represent different coverage levels. The code includes package declarations, class definitions, and a `dispense` method that handles coin insertion and selection.

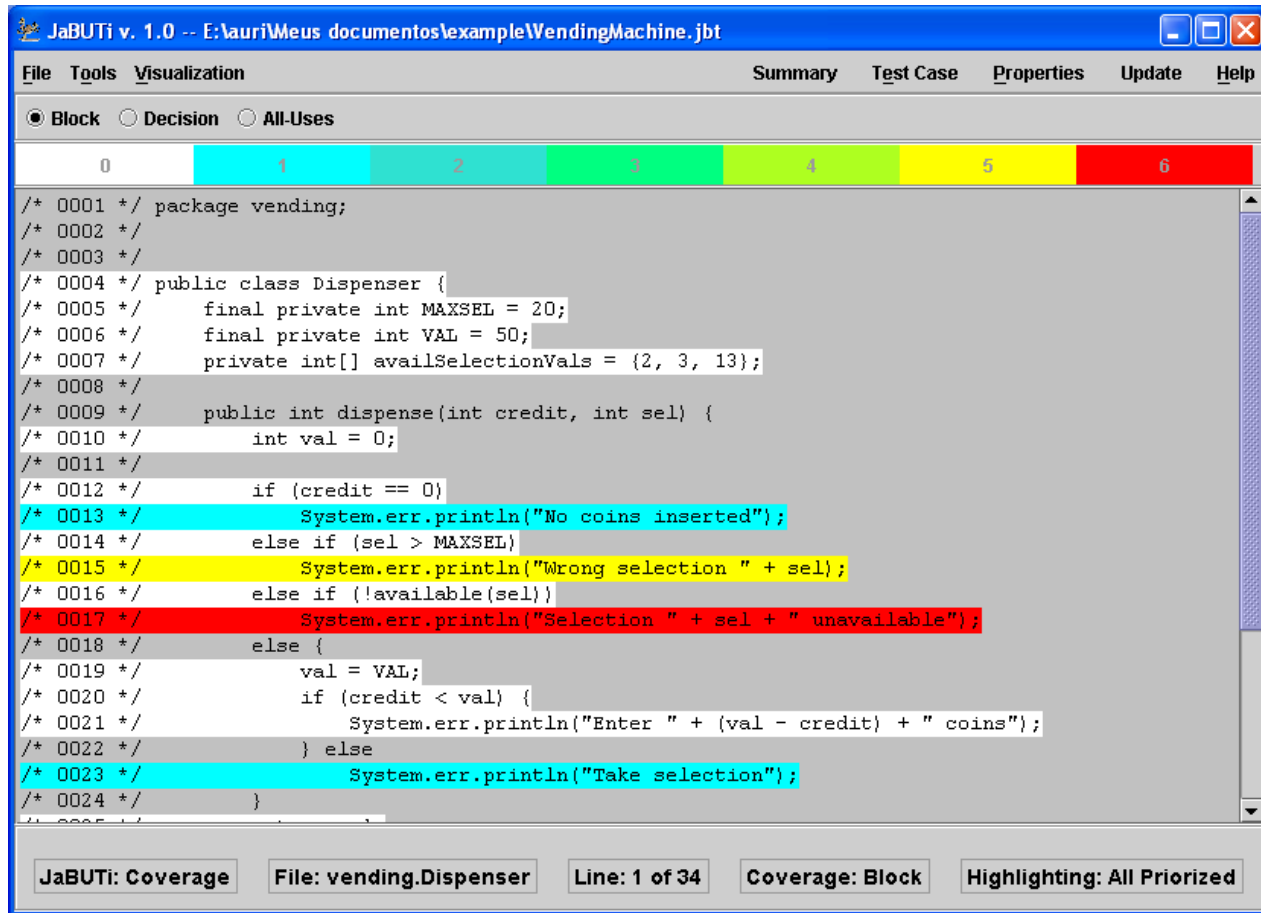
On the right side, a "Control Flow Graphs: vending.Dispenser" window is open, showing a graph of the program's execution flow. The graph consists of nodes (circles) and edges (arrows). Nodes are color-coded: blue for entry/exit points, green for blocks with 100% coverage, yellow for blocks with partial coverage, and red for blocks with no coverage. The graph starts at node 0, branches into 17 and 23, and then further branches into 51, 56, 59, 69, 92, 100, 110, and 6. The graph ends at node 6.

At the bottom of the interface, there is a status bar with the following information:

- JaBUTi: Coverage
- File: vending.Dispenser
- Line: 1 of 34
- Coverage: Block
- Highlighting: All Priorized

Atividades e Resultados no Período

- Priorização de requisitos de teste

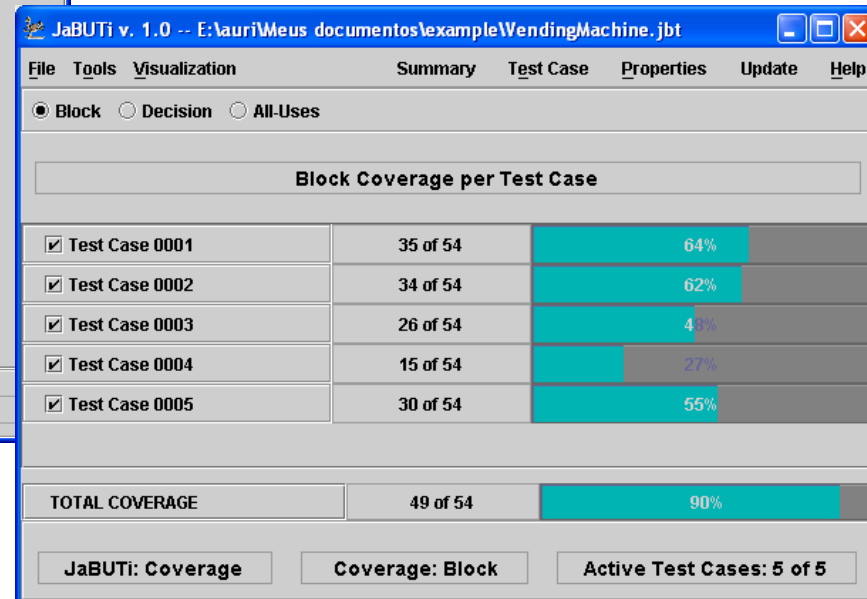
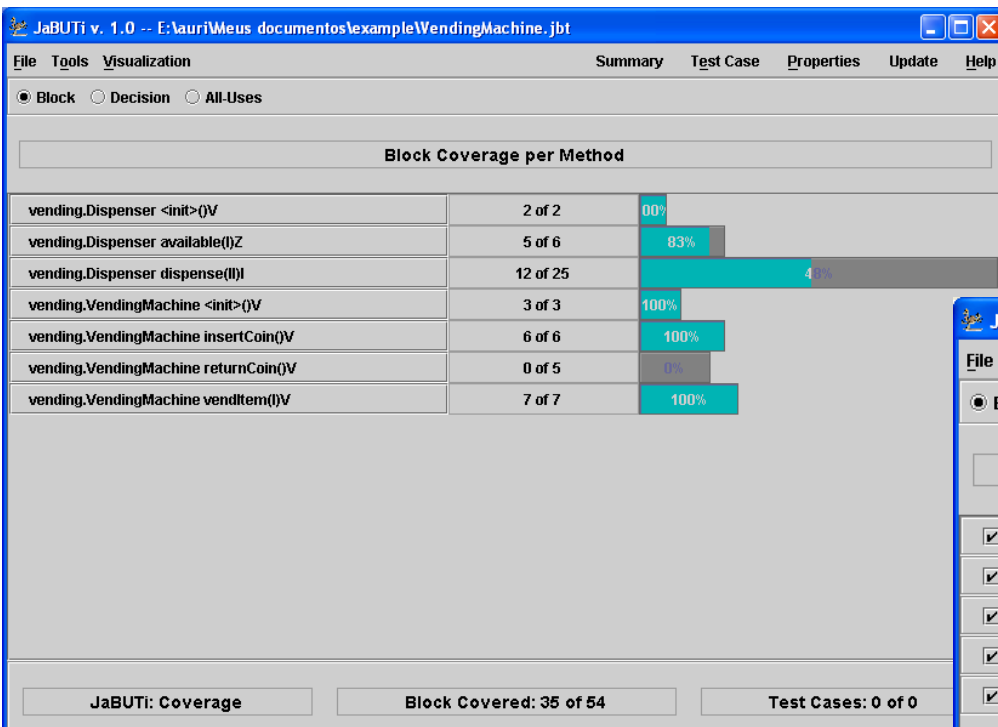


The screenshot shows the JaBUTi v. 1.0 interface. The title bar indicates the file path: E:\auriWeus\documentos\example\VendingMachine.jbt. The menu bar includes File, Tools, Visualization, Summary, Test Case, Properties, Update, and Help. Below the menu bar, there are radio buttons for Block (selected), Decision, and All-Uses. A color-coded bar at the top of the code editor shows seven segments labeled 0 through 6, with colors ranging from white to red. The code editor displays the source code for the VendingMachine class, with lines of code highlighted in various colors corresponding to the coverage bar. The status bar at the bottom shows: JaBUTi: Coverage, File: vending.Dispenser, Line: 1 of 34, Coverage: Block, and Highlighting: All Priorized.

```
/* 0001 */ package vending;
/* 0002 */
/* 0003 */
/* 0004 */ public class Dispenser {
/* 0005 */     final private int MAXSEL = 20;
/* 0006 */     final private int VAL = 50;
/* 0007 */     private int[] availSelectionVals = {2, 3, 13};
/* 0008 */
/* 0009 */     public int dispense(int credit, int sel) {
/* 0010 */         int val = 0;
/* 0011 */
/* 0012 */         if (credit == 0)
/* 0013 */             System.err.println("No coins inserted");
/* 0014 */         else if (sel > MAXSEL)
/* 0015 */             System.err.println("Wrong selection " + sel);
/* 0016 */         else if (!available(sel))
/* 0017 */             System.err.println("Selection " + sel + " unavailable");
/* 0018 */         else {
/* 0019 */             val = VAL;
/* 0020 */             if (credit < val) {
/* 0021 */                 System.err.println("Enter " + (val - credit) + " coins");
/* 0022 */             } else
/* 0023 */                 System.err.println("Take selection");
/* 0024 */         }
/* 0025 */     }
}
```

Atividades e Resultados no Período

- Relatórios variados



Atividades Futuras

- Avaliação/definição de operadores de mutação OO
 - Implementação do ambiente de apoio
- Realização de Estudos Empíricos
 - Avaliação dos Critérios Baseados em Análise de Fluxo de Dados em Programas OO
 - Avaliação dos Critérios Baseados em Mutação em Programas OO
- Equipe
 - 2 Professores, 1 Doutorando

Lista de Publicações

- Teses de Doutorado

[1] Delamaro, M. E., *Mutação de Interface: Um Critério de Adequação Interprocedimental para o Teste de Integração*. Tese de Doutorado, Instituto de Física de São Carlos – Universidade de São Paulo, São Carlos, SP, jun. 1997.

[2] Fabbri, S. C. P. F., *A Análise de Mutantes no Contexto de Sistemas Reativos: Uma Contribuição para o Estabelecimento de Estratégias de Teste e Validação*. Tese de Doutorado, IFSC-USP, São Carlos – SP, out. 1996.

[3] Souza, S. R. S., *Validação de Especificações de Sistemas Reativos: Definição e Análise de Critérios de Teste*. Tese de Doutorado, Instituto de Física de São Carlos/USP, São Carlos, SP, dez. 2000.

Anexo

Lista de Publicações

- Dissertações de Mestrado

- [1] Barbosa, E. F., *Uma Contribuição para a Determinação de um Conjunto Essencial de Operadores de Mutação no Teste de Programas C*. Tese de Mestrado, ICMC-USP, São Carlos – SP, nov. 1998.
- [2] Domingues, A. L. S., *Avaliação de Critérios e Ferramentas de Teste para Programas OO*. Tese de Mestrado, ICMC-USP, São Carlos – SP, jun. 2002.
- [3] Jorge, R. F., *Teste de Mutação: Subsídios para a Redução do Custo de Aplicação*. Tese de Mestrado, ICMC-USP, São Carlos – SP, mar. 2002.
- [4] Souza, S. R. S., *Avaliação do Custo e Eficácia do Critério Análise de Mutantes na Atividade de Teste de Programas*. Tese de Mestrado, ICMC-USP, São Carlos – SP, jun. 1996.
- [5] Vincenzi, A. M. R., *Subsídios para o Estabelecimento de Estratégias de Teste Baseadas na Técnica de Mutação*. Tese de Mestrado, ICMC-USP, São Carlos – SP, nov. 1998.

Lista de Publicações

- Capítulos de Livro

[1] Vincenzi, A. M. R., Maldonado, J. C., Delamaro, M. E., Spoto, E. S., Wong, W. E., *Desenvolvimento Baseado em Componentes*, cap. Teste de Componentes. MANOLE, 2003, (capítulo de livro a ser publicado).

[2] Vincenzi, A. M. R., Maldonado, J. C., Delamaro, M. E., Spoto, E. S., Wong, W. E., *Component-Based Software Quality: Methods and Techniques*, v. 2693 de *Lecture Notes in Computer Science*, cap. Component-Based Software: An Overview of Testing, p. 99–127. New York, NY: Springer-Verlag, jun. 2003.

Lista de Publicações

- Artigos em Revistas Internacionais

[1] Barbosa, E. F., Maldonado, J. C., Vincenzi, A. M. R. Towards the determination of sufficient mutant operators for C. *Software Testing, Verification and Reliability*, v. 11, n. 2, p. 113–136, jun. 2001.

[2] Vincenzi, A. M. R., Maldonado, J. C., Barbosa, E. F., Delamaro, M. E. Unit and integration testing strategies for C programs using mutation-based criteria. *Software Testing, Verification and Reliability*, v. 11, n. 4, p. 249–268, dez. 2001.

[3] Vincenzi, A. M. R., Nakagawa, E. Y., Maldonado, J. C., Delamaro, M. E., Romero, R. A. F. Bayesian-learning based guidelines to determine equivalent mutants. *International Journal of Software Engineering and Knowledge Engineering – IJSEKE*, v. 12, n. 6, p. 675–689, dez. 2002.

Lista de Publicações

- Artigos em Congressos Internacionais

[1] Jorge, R. F., Vincenzi, A. M. R., Delamaro, M. E., Maldonado, J. C., Teste de mutação: Estratégias baseadas em equivalência de mutantes para redução do custo de aplicação. In: *XXVII Latin-American Conference on Informatics – CLEI'2001*, Merida – Venezuela, set. 2001, (disponível em CD-ROM: número do artigo – a202.pdf).

[2] Linkman, S., Vincenzi, A. M. R., Maldonado, J., An evaluation of systematic functional testing using mutation testing. In: *7th International Conference on Empirical Assessment in Software Engineering – EASE*, Keele, UK: The IEE, abr. 2003.

[3] Simão, A. S., Vincenzi, A. M. R., Maldonado, J. C., Santana, A. C. L., IDeL: A language for software instrumentation. In: *XXVIII Latin-American Conference on Informatics - CLEI'2002*, p. 96, Montevideo, Uruguay, nov. 2002, artigo em CD-ROM.

[4] Simão, A. S., Vincenzi, A. M. R., Maldonado, J. C., Santana, A. C. L., IDeL: An abstract language for instrumenting code. In: *24th International Conference on Software Engineering – ICSE'2002*, Buenos Aires, Argentina, maio 2002, (artigo submetido).

Lista de Publicações

- Artigos em Congressos Internacionais (cont.)

[5] Vincenzi, A. M. R., Delamaro, M. E., Maldonado, J. C., Wong, W. E., Java bytecode static analysis: Deriving structural testing requirements. In: *2nd UK Software Testing Workshop – UK-Softest'2003*, Department of Computer Science, University of York, York, England: University of York Press, set. 2003.

[6] Vincenzi, A. M. R., Nakagawa, E. Y., Maldonado, J. C., Delamaro, M. E., Romero, R. A. F., Bayesian-learning based guidelines to determine equivalent mutants. In: *13th International Conference on Software Engineering & Knowledge Engineering – SEKE'2001*, p. 180–187, Buenos Aires, Argentina, jun. 2001, (Aceito para publicação em um edição especial do International Journal of Software Engineering and Knowledge Engineering – IJSEKE).

Lista de Publicações

• Artigos em Congressos Nacionais

- [1] Delamaro, M. E., Pezzè, M., Vincenzi, A. M. R., Maldonado, J. C., Mutant operators for testing concurrent Java programs. In: *SBES'2001 – Simpósio Brasileiro de Engenharia de Software*, p. 272–285, Rio de Janeiro, RJ, out. 2001.
- [2] Domingues, A. L. D., Simão, A. S., Vincenzi, A. M. R., Maldonado, J. C., Evaltool: Um ambiente de apoio à avaliação e seleção de ferramentas de teste para programas orientados a objetos. In: *XVI SBES – Simpósio Brasileiro de Engenharia de Software*, p. 384–389, Gramado, RS, Brasil, out. 2002.
- [3] Simão, A. S., Maldonado, J. C., Vincenzi, A. M. R., mudelgen: A tool for processing mutant operator descriptions. In: *XVI SBES – Simpósio Brasileiro de Engenharia de Software*, p. 426–431, Gramado, RS, Brasil, out. 2002.
- [4] Vincenzi, A. M. R., Delamaro, M. E., Simão, A. S., Wong, W. E., Maldonado, J. C., Jabá – a Java bytecode analyzer. In: *XVI SBES – Simpósio Brasileiro de Engenharia de Software*, p. 414–419, Gramado, RS, Brasil, out. 2002.
- [5] Vincenzi, A. M. R., Wong, W. E., Delamaro, M. E., Maldonado, J. C., JaBUTi: A coverage analysis tool for Java programs. In: *XVII SBES – Simpósio Brasileiro de Engenharia de Software*, Manaus, AM, Brasil, out. 2003, a ser publicado.