

LABORATÓRIO DE LINGUAGENS DE PROGRAMAÇÃO

Roberto da Silva Bigonha

Mariza A. S. Bigonha

UFMG

26 de fevereiro de 2008

**Todos os direitos reservados
Proibida cópia sem autorização do autor**

- **Predição de Esforço de Manutenção Baseada em Conectividade**
KECIA ALINE MARQUES FERREIRA (doutoranda)
Profa. Mariza Andrade da Silva Bigonha (orientadora)
Prof. Roberto da Silva Bigonha (Co-Orientador)
 - Métricas de Software
 - Custo de Manutenção
 - Conectividade, Coesão e Acoplamento
 - Modelo de Estabilidade
 - Reestruturação de Programas

- **Semântica Denotacional Multidimensional**

FÁBIO TIRELO (doutorando)

Prof. Roberto da Silva Bigonha (orientador)

Prof. João Saraiva (Universidade do Minho, Portugal)

Tays Soares (Mestrado concluído)

Mirlaine Crepalde (IC)

Felipe Loredo (IC)

- Escalabilidade de Definições Semânticas
- Produção Industrial
- Definição incremental
- Conceito de Vagueza
- Linguagem Notus (projeto, implementação)
- Validação da Metodologia

- **Compreensão de Programas Baseada na Identificação de Interesses no Código Fonte**

DENIS PINTO PINHEIRO (mestrando)

Prof. Roberto da Silva Bigonha (Orientador)

Prof. Marcelo de Almeida Maia (UFU)

- Interesses Transversais
- Ontologia (OWL)
- Mineração de Aspectos
- Apresentação dos interesses

- **Recuperação de Arquitetura de Software Para Manutenção de Sistemas**

THIAGO HENRIQUE BRAGA (mestrando)

Prof. Roberto da Silva Bigonha (Orientador).

Prof. Marcelo de Almeida Maia (UFU)

- **Grafo de Execução**

- **Recuperação da Arquitetura:
Dutos, Filtros e Componentes**

- **Anotação no Código Fonte:
aspectos com restrições arquiteturais**

- **Verificação de Conformidade
aspecto de verificação geram novo grafo de execução**

- **Programação Concorrente Baseada em Acordes**
SÉRGIO VALE E PACE (mestrando)
Prof. Roberto da Silva Bigonha (Orientador)
 - join-calculus, π -calculus
 - Sincronização de processos por troca de mensagens
 - Abstrações para concorrência
 - Biblioteca para introduzir acordes em Java
 - Novas Abstrações em Java

NOÇÃO DE ACORDES

```
public class Buffer {
    public String get() & public async put(String s) {
        return s;
    }
}

public class User {
    public static void main(String [ ] a ) {
        Buffer b = new Buffer();
        b.put("abc"); b.put("cde");
        System.out.println(b.get()); // imprime abc
        System.out.println(b.get()); // imprime cde
        System.out.println(b.get()); // bloqueia
    }
}
```

- **EQUIPE:**

- Roberto da Silva Bigonha
- Mariza A. S. Bigonha
- Marco Túlio O. Valente (PUC-MG)
- Vladimir Oliveira Di Iorio (UFV)
- Marcelo Maia (UFU)
- Elaine Pimentel (DM)
- alunos de doutorado e mestrado.

- 1. Computação Paralela.**
- 2. Alocação de Registradores.**
- 3. Sistemas MULTI-CORE.**
- 4. Programação Orientada a Linguagens.**
- 5. Análise Estática de Código**

Computação Paralela

- **EQUIPE:** Mariza A. S. Bigonha
Roberto S. Bigonha
Marco Túlio O. Valente (PUC-MG)
alunos de mestrado e doutorado.
- **Sugestão de Tema para Tese de Doutorado:**
 - Usar shape analysis para descobrir árvores ou listas e aplicar inversão para produzir algoritmos paralelos.
- **Sugestão de Tema para Dissertação de Mestrado:**
 - Desenvolver bibliotecas para linguagens paralelas. Um bom exemplo é Fortress (<http://research.sun.com/projects/plrg/>), que está em estágio bastante inicial.

... Computação Paralela

Nesta linha os principais artigos são:

1. K. Morita, Akimasa Morihata, Kiminori Matsuzaki, Zhenjiang Hu, M. Takeichi: Automatic inversion generates divide-and-conquer parallel programs. PLDI 2007: 146-155.
2. Milind Kulkarni, Keshav Pingali, B. Walter, Ganesh Ramanarayanan, Kavita Bala, L. Paul Chew: Optimistic parallelism requires abstractions. PLDI 2007: 211-22
3. Chen Ding, Xipeng Shen, Kirk Kelsey, Chris Tice, Ruke Huang, Chengliang Zhang: Software behavior oriented parallelization. PLDI 2007: 223-234.
4. Armando Solar-Lezama, Gilad Arnold, Liviu Tancau, Rastislav Bodik, Vijay A. Saraswat, Sanjit A. Seshia: Sketching stencils. PLDI 2007: 167-178
5. Sriram Krishnamoorthy, Muthu Baskaran, Uday Bondhugula, J. Ramanujam, Atanas Rountev, P. Sadayappan: Effective automatic parallelization of stencil computations. PLDI 2007: 235-244
6. Lakshminarayanan Renganarayanan, DaeGon Kim, Sanjay V. Rajopadhye, Michelle Mills Strout: Parameterized tiled loops for free. PLDI 2007: 405-414

Sistemas Multi-core

Nesta linha, Um artigo bem interessante é:

- 1. Perry H. Wang, Jamison D. Collins, Gautham N. Chinya, Hong Jiang, Xinmin Tian, Milind Girkar, Nick Y. Yang, Guei-Yuan Lueh, Hong Wang: EXOCHI: architecture and programming environment for a heterogeneous multi-core multithreaded system. PLDI 2007: 156-166**

Alocação de Registradores

- **EQUIPE:** Mariza A. S. Bigonha
Roberto da Silva Bigonha
Vladimir Oliveira Di Iorio (UFV)
alunos de mestrado.
- **Sugestão de Tema para Dissertação de Mestrado:**
 - Desenvolver algoritmos para computadores com múltiplos bancos de registradores, por exemplo, SPARC V9
(www.sparc.org/standards/SPARCV9.pdf).

... Alocação de Registradores

- **Os principais artigos são:**

1. Maximiliano Poletto & Vivek Sarkar, **Linear Scan Register Allocation**, Transactions on Programming Languages and Systems (TOPLAS), volume 21, no. 5, ACM Press, pages 895-913.
2. Alkis Evlogimenos, **Improvements to Linear Scan Register Allocation**, April/2004.
3. Vivek Sarkar et al., **Extended Linear Scan: an Alternate Foundation for Global Register Allocation - ETAPS 2007**
4. Sebastian Hack et al., **Register Allocation for Programs in SSA-Form**, ETAPS2006.
5. Sebastian Hack et al., **Towards Register Allocation for Programs in SSA-form**, Universitat Karlsruhe, 2005, ISSN 1432-7864.
6. Fernando Magno Q. Pereira & Jens Palsberg, **Register Allocation via Coloring of Chordal Graphs**, APLAS, Springer, pages = 315-329, 2005.
7. Daniel Grundi et al., **A Fast Cutting-Plane Algorithm for Optimal Coalescing**, 2007.

Programação Orientada a Linguagens

- **EQUIPE:** Vladimir Oliveira Di Iorio (UFV)
Roberto da Silva Bigonha
Mariza A. S. Bigonha
alunos de mestrado.
- **Sugestões de Temas para Dissertação de Mestrado:**
 1. Infra-Estrutura Para Implementação de Linguagens de Domínio Específico.

Com tal padrão, diferentes ferramentas podem compartilhar suas definições de DSLs, evitando que os usuários fiquem presos a formatos proprietários de ferramentas específicas. Principais ferramentas atuais usam formatos próprios bem diferentes.

... Programação Orientada a Linguagens

2. Ferramenta para Construção de Editores de Linguagens de Domínio Específico.

Proposta de um conjunto de recursos apropriados para construção desses editores especializados, focando em alguns tipos de domínios escolhidos.

3. Ambiente de Implementação de Linguagens de Domínio Específico.

Desenvolvimento de mecanismos para especificação da tradução de representações abstratas para a linguagem hospedeira. Esses mecanismos devem ser definidos na forma de gabaritos para geração de código, e permitir a identificação e fácil correção de erros no formato dos programas.

... Programação Orientada a Linguagens

- Os principais artigos são:

1. M. Fowler. LanguageWorkbenches and Model Driven Architecture, 2005.
<http://www.martinfowler.com/articles/mdaLanguageWorkbench.html>, Last update in June 2005.
2. S. Dmitriev. Language Oriented Programming: The Next Programming Paradigm. onBoard Online Magazine, 1, november 2004.
3. M. Fowler. Language Workbenches:The Killer-App for Domain Specific Languages?, 2005.
<http://martinfowler.com/articles/languageWorkbench.html>, Last update in June 2005.
4. M.P.Ward.Language-oriented programming. Software,Concepts and Tools, 15(4):147-161,1994.
5. D.P.Coura. Produzindo animação através da programação por demonstração. Master's thesis, UFV, 2006.
6. D. P. Coura, V. O. D. Iorio, A. G. Lima, A. de Paiva Oliveira, and M. V. A. Andrade. Animações através da programação por demonstração.In ACM International Conference Proceeding Series - Proceedings of VII Brazilian symposium on Human factors in computing systems, pages 81-90,NY,USA,2006.

... Programação Orientada a Linguagens

Metodologia introduzida por Ward na década de 90 como uma nova forma de se organizar o desenvolvimento de grandes sistemas de software.

- **Etapas:**

- Isola um problema importante dentre as tarefas a serem realizadas pelo sistema e define, formalmente, uma nova linguagem de programação com recursos para agilizar a especificação as soluções especificamente orientada ao domínio do problema escolhido.
- Usando a linguagem construída, implementa-se as partes do sistema que estejam relacionadas ao problema escolhido.
- Implementa um compilador ou interpretador para a nova linguagem.

Estes passos devem ser executados para quantos problemas se considerar adequados. O código final do sistema utilizará linguagens de programação convencionais e uma ou mais linguagens de domínio específico (DSL, do inglês domain-specific languages).

... Programação Orientada a Linguagens

- **Dificuldades:**

- Idealmente, as DSLs definidas devem ser fortemente associadas ao domínio do problema alvo, permitindo que até mesmo usuários não experientes em linguagens de programação convencionais possam usá-las.
- Em geral, elas se tornam muito distintas da linguagem hospedeira, a linguagem convencional base em que o sistema é escrito. Símbolos definidos na linguagem hospedeira são freqüentemente referenciados no código escrito nas DSLs, mas os ambientes de desenvolvimento integrado atuais (IDE, do inglês Integrated Development Environment), não estão preparados para estender às novas DSLs recursos importantes como refactoring e navegação no código.

- **Solução:**

- Desenvolvimento de ferramentas denominadas **language workbenchs**.