



# Recursos Humanos

## RELATÓRIO DE ACTIVIDADES DE BOLSA DE INVESTIGAÇÃO

( A preencher pelo Bolseiro)

RELATÓRIO INTERCALAR

RELATÓRIO FINAL

### Identificação do bolseiro

Nome completo: Maria Leonor Ferreira da Costa Cunha

### Identificação da bolsa

Tipo de bolsa: BI-L Referência: PTDC/EEA-ELC/122756/2010  
Período: De: 2012 - 03 - 19 a: 2013 - 03 - 18  
Nome do projecto: CerVANTES: Co-Validation Tool for Embedded Systems  
Área de trabalho: \_\_\_\_\_  
Orientador científico: Professor José Carlos Campos Costa

### Actividades desenvolvidas

Inicialmente dedicámos a nossa atenção ao estudo de ferramentas de validação para software e hardware existentes. Estudámos a sintaxe de SystemC e a maneira como esta linguagem estava implementada a partir de C++. O objetivo desta atividade era tentar descobrir ferramentas e tecnologias compatíveis com SystemC.

Como resultado do nosso estudo, descobrimos que existiam várias ferramentas de validação para C e C++ que davam a entender inicialmente que a sua migração para SystemC seria uma tarefa factível neste projeto. Esta premissa devia-se ao facto de o SystemC não ser uma linguagem, mas sim uma biblioteca de C++ criada a partir de estruturas e classes de C++ que permitem a simulação de hardware.

A tarefa que se seguiu consistia em descobrir um *parser* de SystemC que permitisse colocar o programa numa estrutura de dados facilmente manipulável. Explorámos várias opções, mas algumas estavam sem suporte à alguns anos e/ou não executavam tarefas que fossem compatíveis com o nosso objectivo. Escolhemos PinaVM devido a vários fatores, em especial devido ao facto de utilizar a framework do LLVM.

A instalação do PinaVM foi bastante demorada uma vez que o projeto estava abandonado à algum tempo e apenas era compatível com versões antigas de SystemC, GCC e LLVM. Além disso levou algum tempo a detectar a origem de alguns erros relacionados com a sua compatibilidade, uma vez que algumas bibliotecas já não existiam nos sistemas operativos recentes.

Seguidamente utilizámos o output do PinaVM e tentámos manipulá-lo. Devido ao facto de o PinaVM efetuar diversas transformações e optimizações ao LLVM IR (Representação Intermédia), o código por ele gerado não era compatível com os objetivos do nosso projeto.

Depois desta tentativa falhada voltámos a olhar para as ferramentas que inicialmente estudámos.

De entre as diversas ferramentas analisadas, destacamos o XOGastan, um projeto já não disponível que permitia converter LLVM IR em XML. Infelizmente depois de algum tempo para a conseguir instalar, o output dado pela ferramenta produzia um XML contendo apenas nós que representavam tipos de dados indeterminados. Concluímos assim que os símbolos de SystemC não eram reconhecidos pela ferramenta e para o serem, seria necessário refazer a ferramenta por completo, solução que colocámos logo de parte.

Tentámos extrair a AST (Abstract Syntax Tree) usando o GCC, o GAWK e o Graphviz. O resultado desta combinação foi uma AST do programa na forma gráfica. Esta solução falhou, uma vez, que a estrutura resultante não se adaptava as necessidades do projeto e os símbolos de SystemC não eram interpretados corretamente.

Nesta fase do projeto, repensámos a nossa abordagem ao problema e optámos por desistir das ferramentas existentes e de usar o output do PinaVM. Decidimos então construir a nossa própria ferramenta baseada na framework do LLVM e de onde se extraia um grafo com a informação necessária para a nossa análise. Como input utilizámos um LLVM IR extraído de uma das fases intermédias de análise do PinaVM.

Apesar desta solução ser mais demorada, revelou-se a solução ideal uma vez que toda a informação necessária ao projeto era armazenada de uma forma mais facilmente manipulável. Esta abordagem foi refeita algumas vezes até conseguirmos fazer o *parser* com sucesso.

O protótipo do *parser* ainda não consegue efetuar a análise de todas as estruturas de dados do C++ e SystemC, mas veio permitir que se avançasse para outra atividade do projeto: a extração de um CFG do programa. Est CFG permite a extração do caminho mais longo para depois verificar a cobertura atingida por este caminho.

Para poder avançar mais rapidamente no projeto decidimos construir a nossa ferramenta baseada em módulos independentes, de forma a poder ir completando o *parser* do programa de SystemC e a restante ferramenta, em simultâneo.

(continuar em folhas adicionais, se necessário)

## **Desvios em relação ao planeado e respectiva justificação**

Este projeto desviou-se um pouco do plano de estudos aprovado, devido ao tempo que se levou a descobrir a abordagem correta para o tratamento do código em SystemC.

Inicialmente as abordagens onde tentámos utilizar o output do PinaVM e de outras ferramentas já existentes compuseram uma série de soluções que, depois de testadas, não deram resultados.

Como este projeto aborda uma linguagem (SystemC) e um tipo de análise quase pioneiro, não tivemos como nos guiar por outros projetos da área.

Outro ponto que constitui um ponto negativo ao projeto foi o facto de se ter de utilizar versões antigas de bibliotecas do Fedora 16 e da framework do LLVM.

No final deste ano o PinaVM voltou a estar ativo e está a ser migrado para a versão do LLVM-3.2 e do SystemC-2.2.0, versões mais atuais, que além de permitirem a sua utilização sem restrições em qualquer versão atual do Linux, permitiram também a utilização da nova informação de debug que o LLVM-3.2 disponibiliza, algo que virá melhorar o nosso parser.

## **Publicações e trabalhos elaborados no âmbito da bolsa**

Este projeto está a ser utilizado como minha tese de mestrado.

*(continuar em folhas adicionais, se necessário)*

### **Bolseiro**

Assinatura: \_\_\_\_\_ Data: \_\_\_\_ - \_\_\_\_ - \_\_\_\_

### **Orientador Científico**

Assinatura: \_\_\_\_\_ Data: \_\_\_\_ - \_\_\_\_ - \_\_\_\_

**Actividades desenvolvidas** *(continuação)*

