Universidade Federal de Pernambuco
Graduação em Ciência da Computação
Centro de Informática

2009.1

Proposta de Trabalho de Graduação
Geração de Modelos de Teste em CSP a partir de Casos de Uso

Aluno: Gleibson Rodrigo Silva de Oliveira (grso@cin.ufpe.br)
Orientador: Augusto Cezar Alves Sampaio (acas@cin.ufpe.br)

Recife, 17 de março de 2009

1. Contexto

O uso de modelos formais, que são maneiras precisas de especificar sistemas computacionais, já é uma realidade na indústria [1]. Um dos benefícios encontrados no uso de modelos formais é um melhor entendimento do sistema antes de sua real implementação. Uma das possíveis aplicações é a geração automática de código a partir modelos CSP [2]. Partindo desses modelos, obtidos da transformação de casos de uso em processos CSP, são gerados automaticamente casos de teste para funcionalidades específicas e interação entre elas, no contexto da pesquisa em cooperação entre Centro de Informática – Cin – UFPE e Motorola Inc. [6] chamada CInBTCRD. A principal dificuldade dessa abordagem é manter a equivalência entre o modelo CSP e os casos de uso após a transformação, que geralmente é uma atividade manual.

A aplicação manual de transformações de casos de uso para modelos CSP [12] pode incorrer em uma série de problemas relacionados à equivalência entre o modelo e o caso de uso. A formalização dos casos de uso é feita através da definição de uma linguagem de especificação. Essa linguagem é chamada CNL (Controlled Natural Language) [4]. Por razões óbvias, não é possível permitir toda a linguagem como fonte. Foi definido, um subconjunto de Inglês com uma gramática fixa para permitir a automatização da tradução para CSP. Devido ao contexto da pesquisa, CInBTCRD, a CNL proposta reflete o domínio. O modelo formal obtido na transformação é usado internamente para geração de casos de teste [1].

Particularmente, CSP é o formalismo padrão adotado no projeto de pesquisa Brazil Test Center (BTC), uma cooperação entre a Universidade Federal de Pernambuco e Motorola Inc., no contexto de testes de softwares embarcados que roda sobre celulares. O rico repertório de operadores CSP é usado para modelar funcionalidades individuais como também para padrões de interações entre funcionalidades. Os modelos CSP são automaticamente construídos [13] a partir de casos de uso descritos em uma linguagem específica de domínio para aplicações móveis (CNL) [6]. Os modelos CSP são entradas para o processo de geração de casos de teste. Esses testes obtidos representam apenas o caminho a ser seguido, não informando dados de entrada nos estados.

A figura abaixo representa o fluxo de geração de casos de teste do projeto de pesquisa BTC. No início do mesmo (à esquerda) encontram-se os documentos de casos de uso, escritos em linguagem natural (CNL) e os modelos CSP. A transformação dos casos de uso para os modelos em CSP, na atual abordagem [4], produz casos de teste (à direita) apenas com caminhos a serem percorridos, não representando explicitamente dados de entrada, saída nem manipulação de dados. A evolução proposta é levar dados do caso de uso com estados até o caso de teste, provendo um caso de teste mais completo.

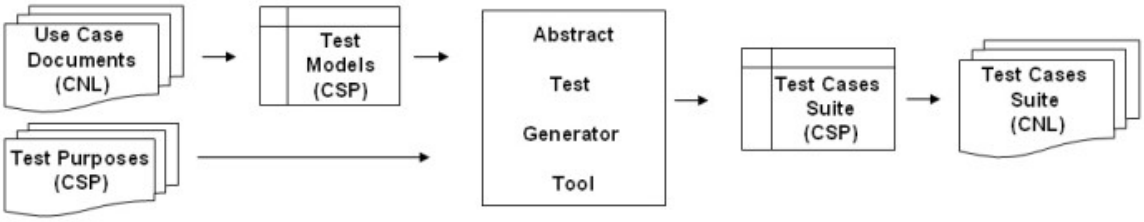

Figura 1 Ciclo Completo de geração de casos de teste

2. Objetivos

Uma solução eficaz que automatiza a geração de modelos CSP a partir de casos de uso com estados, levando em consideração a utilização de dados, além do próprio caminho a ser percorrido durante o teste. Esses dados de entrada nos casos de teste serão gerados considerando os estados dos casos de uso. Existem trabalhos iniciados na literatura que não consideram esses dados (*inputs*) para o modelo CSP e conseqüentemente para o caso de teste.

Como ponto de partida, utilizaremos neste trabalho um conjunto de pesquisas relacionadas que produziram modelos, linguagens (CNL) e ferramentas para transformação de casos de uso em modelos CSP.

Ao final, será produzida uma ferramenta que realize a transformação de maneira automatizada de casos de uso com estados para modelos CSP, com a inclusão de dados para *inputs* de casos de teste.

3. Cronograma

	Mês															
Atividade	**Março**				**Abril**				**Maio**				**Junho**			
Estudo do estado da arte																
Revisão da ferramenta existente																
Implementação das melhorias																
Testes e Validação Final																
Escrita do relatório																
Preparação da apresentação																

O cronograma de trabalho foi definido de forma a concentrar as atividades iniciais no estudo da literatura e na implementação da ferramenta. Em seguida será dado enfoque na fase de testes. Numa última fase serão documentados todos os processos/técnicas de implementação além dos resultados obtidos.

4. Referências

1. Cabral, G. Sampaio, A. Automated Formal Especification Generation and Refinements from Requirements Documents.
2. Alexander Holt. Formal verification with natural language specifications: guidelines, experiments and lessons so far. South African Computer Journal, 24:253–257, 1999.
3. S. Dalal, A. Jain, N. Karunanithi, J. Leaton, C. Lott, G. Patton, and B. Horowitz. Model-based testing in practice. In ICSE'99: Proceedings of the 21st international conference on Software engineering, pages 285–294. IEEE Computer Society Press, 1999.
4. Gustavo Cabral and Augusto Sampaio. Formal specification generation from requirement documents. In Brazilian Symposium on Formal Methods (SBMF), pages 217–232, 2006.
5. Simon St.Laurent, Evan Lenz, and Mary McRae. Office 2003 XML: Integrating Office with the rest of the world. O'Reilly & Associates, Inc., 2004
6. Nogueira, S. Sampaio, A. Mota, A. Guided Test Generation from CSP Models. Theoretical Aspects of Computing - ICTAC 2008 (2008), pp. 258-273.
7. Leitâo, D., Torres, D., Barros, F.A.: Nlforspec: Translating natural language descriptions into formal test case specifications. In: SEKE, Knowledge Systems Institute Graduate School, pp. 129–134 (2007)
8. Maritta Heisel and Jeanine Souqui` eres. A method for requirements elicitation and formal specification. In ER'99: Proceedings of the 18th International Conference on Conceptual Modeling, pages 309–324. Springer, 1999.
9. Alexander Holt. Formal verification with natural language specifications: guidelines, experiments and lessons so far South African Computer Journal, 24:253–257, 1999.
10. Anneke Kleppe, Jos Warmer, and Wim Bast. MDA Explained: The Model Driven Architecture: Practice and Promise. Addison-Wesley, 2003.
11. Daniel Leitão. NLForSpec: Translating natural language descriptions into formal test case specifications. Master's thesis, Universidade Federal de Pernambuco (UFPE), 2006.
12. Hoare, C. A. R. "Communication Sequential Processes". Prentice-Hall, 1985
13. Cabral, G. Sampaio, A.: Formal Specification Generation From Requirements Documents. Electron. Notes Theor. Comput. Sci. 195, 171-188 (2008); Best Paper Award.

5. Assinaturas

_____

Augusto Cezar Alves Sampaio
(Orientador)

_____

Gleibson Rodrigo Silva de Oliveira
(Aluno/Proponente)

Recife, 17 de março de 2009