

Relatório para o Simpósio Doutoral 2006

Óscar Rafael Ribeiro

9 de Fevereiro de 2006

Identificação

Candidato: Óscar Rafael Ribeiro
`oscar.rafael@di.uminho.pt`

Título da Tese: Formalization of Behavioural Software Models

Orientador: João Miguel Fernandes (UM/DI)
`jmf@di.uminho.pt`

Data Início: Junho 2005

Data Término: Junho 2008 (prevista)

Resumo

Pretende-se com este trabalho estudar mecanismos para a obtenção de modelos de redes de Petri, a partir de conjuntos de cenários de utilização de um sistema, representados em diagramas de sequências (presentes na notação UML).

O propósito principal do modelo de redes de Petri é permitir à equipa de desenvolvimento animar e validar junto do cliente, ou utilizador, protótipos de software que retratam o comportamento global que se inferiu a

partir do conjunto dos cenários considerados. Para tal, há que formalizar os meta-modelos envolvidos e definir as regras de transformação para automatizar o processo de obtenção dos modelos de redes de Petri. Desta forma, obtém-se numa fase inicial do processo de desenvolvimento um protótipo de software que será usado para animar, validar e verificar o comportamento e algumas propriedades do sistema considerado. Prevê-se a realização de um caso demonstrativo numa empresa de transportes com vista à validação da metodologia considerada.

1 Enquadramento

Os modelos comportamentais desempenham um papel importante na engenharia de sistemas baseados em software. Esses modelos permitem uma abordagem sistemática para a definição dos requisitos, especificação, concepção da arquitectura, simulação, geração de código, e verificação e validação dos sistemas. Existe uma variedade de notações, técnicas e ferramentas que, suportando a modelação de comportamento, é sugerida para estas tarefas de desenvolvimento. Nestas notações, técnicas e ferramentas, duas abordagens complementares à modelação de comportamento podem ser identificadas [7]:

- baseadas em interacções, ou cenários;
- baseadas em estados.

A modelação baseada em cenários centra-se nas interacções (por exemplo troca de mensagens em sistemas orientados aos objectos) entre os actores e as componentes do sistema. Desta forma, o seu principal construtor de modelação é a comunicação entre essas entidades. A modelação de interacção é habitualmente realizada usando técnicas baseadas em cenários e casos de uso, o que oferece uma visão alargada de um sistema que é particularmente adequada para a comunicação entre os diferentes intervenientes do projecto. A modelação baseada em estados, concentra-se tipicamente no comportamento interno de cada componente. Os formalismos baseados em estados têm diferentes aplicações na engenharia de software, particularmente no desenho de sistemas: orientados ao objecto, distribuídos, tempo real e reactivos.

Em muitos casos, os formalismos baseados em estados suportam uma análise automática e rigorosa, como por exemplo a verificação de modelos.

A notação UML [5] inclui um conjunto de diagramas baseados em interações (nomeadamente diagrama de sequência, de colaboração, de vista global de interações), dos quais o mais comum é o diagrama de sequência [1]. Neste diagrama estão representadas mensagens entre um número de linhas temporais associadas a cada componente. Estes diagramas permitem a interpretação de um conjunto de cenários de utilização do sistema, pelos vários intervenientes no processo de desenvolvimento de software. Isto permite que se possam validar com os utilizadores finais esses cenários de utilização, numa fase inicial do desenvolvimento de software, o que se revela crucial para garantir que o sistema a construir é aquele que o utilizador precisava.

As redes de Petri constituem um modelo adequado para a modelação de sistemas concorrentes, e possuem um extenso conjunto de resultados quer teóricos, quer práticos. As redes de Petri coloridas (CPNs) [2] pertencem à classe das redes de Petri de alto nível, uma vez que introduzem uma série de mecanismos para lidar com a complexidade dos modelos criados, e, devido à sua base formal, têm associados métodos de verificação de correcção [3, 4]. As CPNs estão munidas de uma ferramenta, as chamadas *CPN-tools* [6], que facilita a sua utilização prática, para a sua edição, simulação e verificação. Esta ferramenta permite ainda criar animações baseadas no modelo criado, obtendo-se assim um protótipo do sistema modelado que pode ser usado para validar algumas funcionalidades do sistemas com o utilizador final.

Neste trabalho vai-se estudar formas de transformar diagramas de sequência em CPNs, com vista à obtenção de novos mecanismos que permitam a automatização dessa transformação.

2 Contribuições

2.1 Publicações

Óscar R. Ribeiro, João M. Fernandes and Ricardo J. Machado.

Generation of PROMELA for Verification of Synchronous Petri Nets

Submitted to *Sixth International Conference on Application of Concurrency to System Design (ACSD'06)*.

Abstract: *For developing embedded systems, the design process should use formal methods, before final design and implementation decisions are taken. The Synchronous and Interpreted Petri Net (SIPN) is considered to model embedded systems. This meta-model is based on safe PN with guarded transitions and synchronous transitions firing, and also includes enabling and inhibitor arcs. The Spin tool, whose input language is PROMELA, is a verification system based on the model checking technique. This work presents how to translate an SIPN model into PROMELA code, to verify its properties. It discusses in detail the adequacy of the created PROMELA model for verification through model checking.*

Este artigo aborda a geração de código para efeitos de model-checking a partir de Redes de Petri. Prevê-se que este trabalho possa ser enquadrado no âmbito do projecto, para validar as redes de Petri coloridas obtidas dos cenários.

3 Desenvolvimento

1. Fazer o levantamento do estado da arte associado à passagem de modelos baseados em cenários para modelos baseados em estados.
2. Escrever, até finais de Fevereiro, um relatório de aproximadamente 60 páginas, com a síntese do estado da arte.
3. Definir as regras e os mecanismos de utilização dos cenários para geração do respectivo modelo de rede de Petri coloridas.
4. Formalizar os meta-modelos envolvidos e as regras de transformação, a fim de automatizar o processo de obtenção dos modelos de Redes de Petri coloridas.
5. Aplicar o método de transformação a exemplos concretos. Inicialmente, prevê-se o uso de pequenos exemplos, para mais tarde se recorrer a

sistemas de maior complexidade. Este sistema deverá idealmente ser um exemplo real, para assim validar a aplicabilidade do método em contexto de engenharia. Em princípio, será realizado um caso demonstrativo numa empresa de transportes.

6. Escrita da tese;

Referências

- [1] Martin Fowler. *UML Distilled: A Brief Guide to the Standard Object Modelling Language*. Addison-Wesley, 2004.
- [2] Kurt Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*, volume 1: Basic Concepts of *EATCS*. Springer-Verlag, monographs in theoretical computer science edition, 1997.
- [3] Kurt Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*, volume 2: Analysis Methods of *EATCS*. Springer-Verlag, monographs in theoretical computer science edition, 1997.
- [4] Kurt Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*, volume 3: Practical Use of *EATCS*. Springer-Verlag, monographs in theoretical computer science edition, 1997.
- [5] Object Management Group. *UML 2.0 Superstructure Specification*. Object Management Group, August 2005.
- [6] Anne Vinter Ratzner, Lisa Wells, Henry Michael Lassen, Mads Laursen, Jacob Frank Qvortrup, Martin Stig Stissing, Michael Westergaard, S. Christensen, and Kurt Jensen. CPN Tools for Editing, Simulating, and Analysing Coloured Petri Nets. In W. van der Aalst and E. Best, editors, *Applications and Theory of Petri Nets 2003: 24th International Conference, ICATPN 2003*, volume 2679 of *Lecture Notes in Computer Science*, pages 450 – 462, Eindhoven, The Netherlands, Jan 2003. Springer-Verlag.
- [7] Sebastian Uchitel, Manfred Broy, Ingolf H. Krüger, and Jon Whittle. "guest editorial: Special section on interaction and state-based modeling". *IEEE Transactions on Software Engineering*, 31(12):997–998, 2005.