ANÁLISE E VERIFICAÇÃO FORMAL DE CENÁRIOS BASEADOS EM MSCs E ÁLGEBRA DE PROCESSOS

Amanda C. Barbosa[1], Miriam B. Alves[2]

[1]Universidade do Vale do Paraíba (UNIVAP)
São José dos Campos - SP – Brasil.

[2]Instituto de Aeronáutica e Espaço (IAE)
Praça Marechal Eduardo Gomes, 50 – São José dos Campos - SP – Brasil.

amandacb3@yahoo.com.br; miriamalves@iae.cta.br

Resumo - Este artigo propõe uma abordagem formal para uma análise e verificação de cenários de um sistema de software concorrente baseado em Diagrama de Seqüência de Mensagens (MSC) e Álgebra de Processos. Os cenários são representados como MSCs e sua semântica é formalmente expressa em Álgebra de Processos, tornando possível identificar e apresentar todos os traçados de execução deste cenário. Após a identificação destes traçados através da Álgebra de Processos, os mesmos são convertidos para MSCs, permitindo não somente a análise e verificação dos requisitos do sistema, mas também seqüências de controle críticas, garantindo importantes propriedades do sistema, tornando-o mais seguro e confiável. Um clássico e simples exemplo de concorrência é apresentado para ilustrar a abordagem formal proposta.

Palavras-chave: Diagrama de Seqüência de Mensagens, Álgebra de Processos, Concorrência, Cenários
Área do Conhecimento: Ciências Exatas e da Terra

Introdução

Diagrama de Seqüência de Mensagens (MSC) têm sido cada vez mais utilizados para analisar e especificar requisitos de um sistema de software [8]. Trata-se de uma linguagem gráfica e textual utilizada para modelar cenários do sistema, descrevendo o comportamento da comunicação de um conjunto de entidades logicamente ou fisicamente distribuídas, mostrando a ordem em que as mensagens são trocadas [5]. Os MSCs são amplamente utilizados para especificação e modelagem dos requisitos nas primeiras fases do desenvolvimento de software.

Além da representação gráfica e textual, o MSC é caracterizado por uma semântica formal, chamada de semântica denotacional [6]. A semântica denotacional de um MSC é expressa pela tradução de uma representação gráfica em uma expressão em Álgebra de Processos, baseada na Álgebra de Processos ACP (Algebra of Communicating Processes) [11]. Esta representação semântica está padronizada na recomendação Z.120 anexo B.

O objetivo deste trabalho é a análise formal de cenários de um sistema concorrente de software, verificando ainda na fase de desenvolvimento, todos os possíveis traçados deste cenário, com o objetivo de se identificar situações críticas que podem ocorrer, tais como deadlock. Considerando que a identificação de cenários possíveis num sistema concorrente é uma tarefa árdua, com uma grande possibilidade de que nem todos os cenários sejam corretamente identificados, este trabalho propõe uma maneira formal de fazer uma identificação completa.

Diagrama de Seqüência de Mensagens

Nesta seção o MSC é apresentado de forma sucinta, considerando somente o essencial para a sua compreensão. Maiores detalhes da documentação do MSC pode ser encontrado em [1] [5] [7].

O diagrama de seqüência de mensagens descreve um cenário onde algumas instâncias se comunicam com outras. Tais cenários incluem a descrição de mensagens enviadas, mensagens recebidas, eventos locais e o ordenamento entre eles. Mensagens em um MSC são consideradas assíncronas, isto é, o ato de enviar uma mensagem é totalmente separado do ato de receber a mesma. Naturalmente, a recepção de uma mensagem tem que ocorrer depois do envio da mensagem.

As instâncias em um MSC são representadas por um eixo vertical indicando a linha de vida e as setas conectadas ao longo dos eixos das instâncias representam a troca de mensagens entre as instâncias (Figura 1). Embora o uso do MSC envolva principalmente a representação gráfica, ele também possui uma representação textual, conforme mostrado na Figura 1.

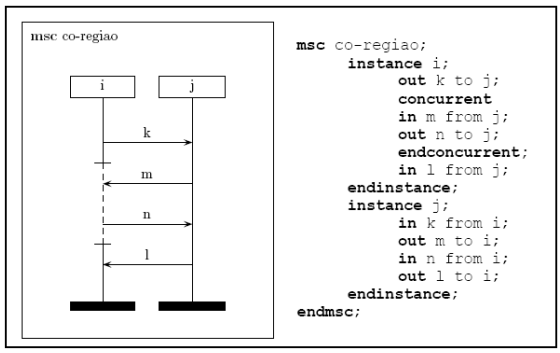

Figura 1 - MSC estruturado com co-região.

A Figura 1 mostra um MSC com uma região concorrente (co-região), representada pela linha pontilhada no eixo da instância **i**. As duas instâncias **i** e **j**, trocam mensagens rotuladas de **k**, **m**, **n** e **l**. As mensagens também podem ser enviadas ou recebidas do ambiente, representado por um retângulo que define os limites do MSC.

A linguagem MSC inclui ainda outros elementos que são necessários para especificar o fluxo de mensagem. Estes elementos são ações, temporizadores, criação e destruição de instâncias e condições. MSCs mais estruturados podem ser construídos incorporando mecanismos que permitem a decomposição da instância e uso, expressões *in-line*. As co-regiões, mencionadas anteriormente, são especialmente importantes para a modelagem de sistemas concorrentes de tempo real, uma vez que elas permitem modelar o não determinismo destes sistemas. Dentro da co-região os eventos não são ordenados no tempo. No exemplo da Figura 1, o recebimento da mensagem **m** não ocorre necessariamente antes do envio da mensagem **n**.

O MSC é uma linguagem que pode ajudar no melhor entendimento de aspectos importantes dos cenários de um sistema, enfatizando a interação entre as instâncias, as regras associadas e as restrições de tempo. Na próxima seção será apresentada brevemente a semântica formal do MSC.

Semântica Formal de um MSC baseado em Álgebra de Processos

No apêndice B da recomendação Z.120 [6] é apresentada uma descrição formal da semântica do MSC usando álgebra de processos. A idéia básica da álgebra de processos é que se defina uma "matemática de software" através da qual programas e especificações (comportamento) possam ser manipulados simbolicamente com a mesma facilidade que expressões algébricas [9].

A semântica de um MSC é uma expressão em álgebra de processos, que é criada a partir dos seguintes operadores: operador de composição seqüencial (**.**) composição alternativa (**+**) e composição paralela (**||**) [3] [10]. O envio de uma mensagem é denotado pela palavra *out*, enquanto o recebimento da mesma é denotado pela palavra *in*, seguidas do nome das instâncias envolvidas e do nome da mensagem.

A idéia básica para a representação semântica do MSC é que as instâncias operam em paralelo, independente uma da outra [4]. Assim, a semântica do MSC representado na Figura 1 é a composição paralela das instâncias **i** e **j**.

Ao longo do eixo de uma instância os eventos são totalmente ordenados do topo para a base. Esta rigorosa ordenação é refletida na semântica pelo uso do operador de composição seqüencial (.). Para que haja um relaxamento desta ordem estrita dos eventos ao longo do eixo de uma instância, é utilizada a co-região. Na Figura 1, a instância **i** contém uma co-região onde o ordenamento dos eventos é totalmente livre. Ao invés de usar o operador de composição seqüencial, para representar a co-região usa-se o operador de composição paralela || (execução intercalada). A semântica da instância **i** é: $out(i,j,k) \,.\, in(j,i,m) \,||\, out(i,j,n) \,.\, in(j,i,l)$. Depois de expandir o operador ||, a expressão resultante para a instância **i** é:

$$out(i,j,k) \,.\, in(j,i,m) \,.\, out(i,j,n) \,.\, in(j,i,l) + out(i,j,k) \,.\, out(i,j,n) \,.\, in(j,i,m) \,.\, in(j,i,l)$$

A semântica da instância **j** é: $in(i,j,k) \,.\, out(j,i,m) \,.\, in(i,j,n) \,.\, out(j,i,l)$.

Considerando que uma mensagem não pode ser recebida antes de ser enviada, é introduzido o operador estado λ para garantir esta condição. A sentença em álgebra de processos para o MSC da Figura 1 com a aplicação do operador λ é a seguinte expressão (1):

$$\lambda(out(i,j,k) \,.\, (\, in(j,i,m) \,.\, out(i,j,n) \;+\; out(i,j,n) \,.\, in(j,i,m)) \,.\, in(j,i,l) \,||\, in(i,j,k) \,.\, out(j,i,m) \,.\, in(i,j,n) \,.\, out(j,i,l)) \quad (1)$$

Com o uso do operador λ, depois de expandir o operador paralelo || da expressão (1), há vários traçados que podem ser eliminados. Após a aplicação λ, a semântica do MSC da Figura 1 é a seguinte:

$$\begin{aligned} &out(i,j,k) \,.\, (\, in(i,j,k) \,.\, (\, out(j,i,m) \,.\, (\, out(i,j,n) \,.\, in(i,j,n) \,.\, in(j,i,m) \;+\; in(j,i,m) \,.\, out(i,j,n) \,.\, in(i,j,n)\,)\,) \\ &\qquad + (\, out(i,j,n) \,.\, (\, out(j,i,m) \,.\, in(i,j,n) \,.\, in(j,i,m) \\ &\qquad\quad + out(j,i,m) \,.\, in(j,i,m) \,.\, in(i,j,n)\,)\,) \\ &\qquad .\, out(j,i,l) \,.\, in(j,i,l)) \end{aligned} \quad (2)$$

A expressão mostra que há quatro possíveis traçados para o MSC em questão.

Usando as técnicas descritas acima, uma vez que o cenário é modelado como um MSC e sua semântica é representada em uma sentença correspondente em álgebra de processos, uma

análise e verificação formal de todos os traçados dos MSC podem ser realizadas.

Geração dos traçados de um MSC baseado em Álgebra de Processos

Logo que um cenário é modelado como um MSC, a expressão em álgebra de processos, que representa formalmente a sua semântica, é construída baseada na representação gráfica do MSC sob análise. A expressão resultante dará todos os possíveis traçados de um cenário. Dependendo do número de instâncias, mensagens e regiões concorrentes de um MSC, o número de traçados de execução possíveis pode ser significativamente alto.

A análise e verificação dos vários traçados possíveis da expressão formal resultante pode ser extremamente trabalhosa e, em alguns casos, até mesmo impossível. Considerando que o analista não tem necessariamente a habilidade para interpretar o resultado da expressão em álgebra de processos, e mais importante, nenhuma ajuda gráfica é fornecida para uma melhor compreensão dos traçados, foi desenvolvido o programa PA2MSC (Process Algebra to MSC) que converte cada um dos traçados de uma expressão em álgebra de processos no MSC gráfico correspondente.

O Programa PA2MSC

O objetivo principal do programa PA2MSC é converter uma expressão formal em álgebra de processos em todos os possíveis traçados de execução representados graficamente como MSCs. O programa PA2MSC foi projetado para ser usado em conjunto com LatexMacro Package para Diagrama de Seqüência de Mensagens [2]. A Figura 2 mostra as interfaces externas do PA2MSC.

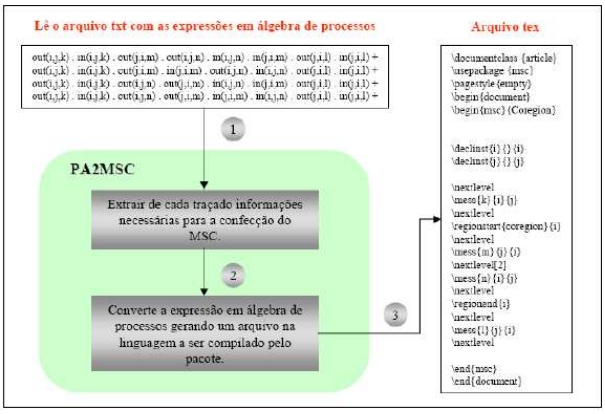

Figura 2 - Diagrama das interfaces externas ao PA2MSC.

O processo para criar a representação gráfica dos traçados (MSCs) foi estabelecido da seguinte maneira: o programa PA2MSC lê o arquivo texto no formato .txt contendo as expressões em álgebra de processos, e então converte as expressões gerando um arquivo no formato .tex que será compilado posteriormente no LatexMacro Package, gerando um arquivo de saída , do tipo Postscript, contendo todos os MSCs referentes a expressão em álgebra.

Estudo de Caso – "Jantar dos Filósofos"

Dentre os vários estudos de caso realizados, foi selecionado o estudo de caso "O Jantar dos Filósofos" [12]. Este problema, clássico da teoria da concorrência, consiste de um conjunto finito de processos (filósofos) que partilham um conjunto finito de recursos (garfos), cada um dos quais pode ser utilizado por apenas um processo por vez, conduzindo assim a potenciais situações de deadlock. A modelagem deste problema foi feita utilizando a linguagem MSC e o conceito de co-região (regiões concorrentes).

A Figura 3 apresenta um exemplo deste problema, considerando três filósofos, representados pelas instâncias PH1, PH2 E PH3, e três recursos, representados pelos grafos pelas instâncias F1, F2 e F3.

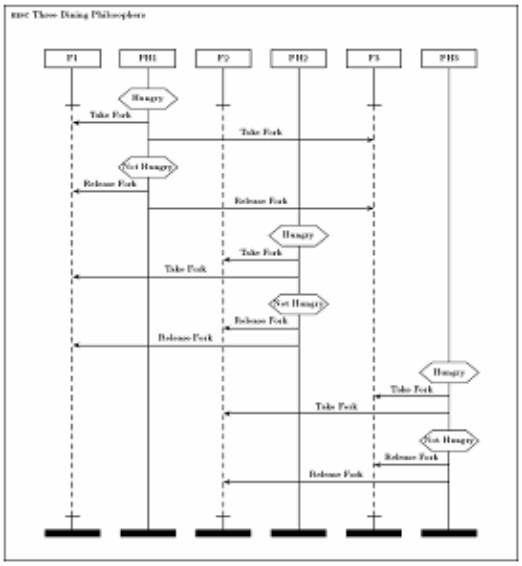

Figura 3 - MSC referente ao cenário do Jantar dos Três Filósofos

O MSC da Figura 3 representa a seguinte situação: quando um filósofo está com fome (condição *Hungry*) ele envia a mensagem *Take Fork* para pegar os garfos, entretanto quando ele não está mais com fome (condição *Not Hungry*) ele envia a mensagem de liberação dos garfos, *Release Fork*.

A semântica em álgebra de processos do MSC da Figura 3 é dada pela expressão (3):

out(PH1,F1,Take Fork).out(PH1,F3,Take Fork).out(PH1,F1,Release Fork).out(PF1,F3,Release Fork)|| in(PH1,F1,Take Fork) || in(PH1,F1,Release Fork).in(PH2,F1,Take Fork) || in(PH2,F1,Release Fork) ||

out(PH2,F2,Take Fork).out(PH2,F1,Take Fork).out(PH2,F2,Release Fork).out(PH2,F1,Release Fork) || *in(PH2,F2,Take Fork)* || *in(PH2,F2,Release Fork)* || *in(PH3,F2,Take Fork)* || *in(PH3,F2,Release Fork)* || *out(PH3,F3,Take Fork).out(PH3,F2,Take Fork).out(PH3,F3,Release Fork).out(PH3,F2,Release Fork)* || *in(PH1,F3,Take Fork)* || *in(PH1,F3,Release Fork)* || *in(PH3,F3,Take Fork)* || *in(PH3,F3,Release Fork))* (3)

A expressão (3) estabelece seis possíveis traçados do cenário da Figura 3, fornecendo ao analista uma descrição mais detalhada do comportamento do sistema. Cada traçado ainda pode ser expandido internamente (operador ||) resultando em todos os possíveis traçados de execução do cenário original da Figura 3. Utilizando o programa PA2MSC e o MSC Macro Package todos os traçados foram graficamente representados como MSCs. Contudo, devido à limitação de espaço, apenas um dos traçados é mostrado na Figura 4.

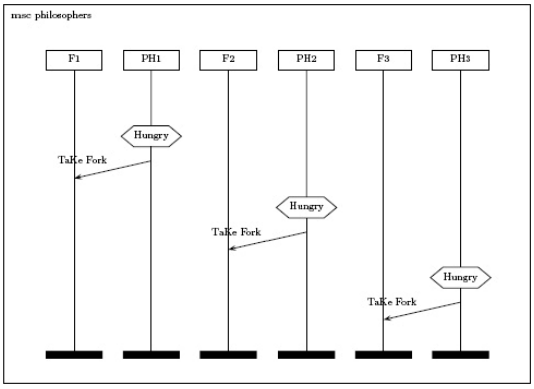

Figura 4 – Traçado com Deadlock.

Quando todos os filósofos pegam seu garfo esquerdo simultaneamente, nenhum será capaz de pegar seu garfo direito e haverá *deadlock*, onde nenhum dos filósofos poderá executar outra ação.

Conclusão

A abordagem proposta neste trabalho ajuda na verificação dos cenários de um sistema, principalmente os concorrentes, melhorando a compreensão dos requisitos, e a identificação formal de erros, ainda na fase de análise. Situações críticas de um sistema concorrente podem ser detectadas, o que seria difícil de realizar usando técnicas tradicionais de teste. Imprecisão e erros dos modelos podem ser encontrados no início, permitindo ajustes com base nos requisitos de sistema.

Normalmente a análise e verificação formal de um sistema não são feitas para o sistema como um todo, mas apenas em partes críticas que merecem mais atenção. Neste trabalho esta abordagem também é válida, considerando que a geração da sentença em álgebra de processos pode se tornar bastante trabalhosa para um número grande de instâncias. Neste sentido, um conjunto de MSCs, representando partes críticas do sistema, pode ser selecionado para ser formalmente analisado e verificado, onde propriedades específicas podem ser investigadas.

Referências

[1] RUDOLPH, E.; GRAUBMANN, P.; and GABOWSKI, J., "Tutorial on Message Sequence Charts", in Computer Networks and ISDN Systems, vol. 28, 1996. http://citeseer.ist.psu.edu/rudolph96tutorial.html - última visita 08/10/07

[2] Dijkstra, E.W. "A Latex Macro Package for Message Sequence Chart – User Manual."

[3] MAUW, S.; RENIERS, M.A. "An Algebric Semantic of Basic Message Sequence Charts". The Computer Journal, vol. 37, 1994.http://citeseer.ist.psu.edu/mauw94algebric.html - última visita 08/10/07

[4] MAUW, S."The Formalization of Message Sequence Charts". Computer Networks and ISDN Systems, vol. 28, 1996. http://citeseer.ist.psu.edu/mauw94algebric.html - última visita 08/10/07

[5] ITU-T. Recommendation Z.120 "Message Sequence Chart " ITU-T, Geneva 04/2004.

[6] ITU-T. Recommendation Z.120 Anexo B "Algebric Semantics of Message Sequence Chart " ITU-T, Geneva 04/1998.

[7] RENIERS, M.A. "Message Sequence Chart: Syntax and Semantics " Eindhoven University of Tecnology,1999.

[8] MAUW, S.; RENIERS, M.A.; WILLEMSE, T.A.C. "Message Sequence Charts in The Software Engineering Process". Handbook of Software Engineering an knowledge Engineering World Scientific, 2000. http://citeseer.ist.psu.edu/mauw01message.html - última visita 08/10/07

[9] BAETEN, J.C.M. "A Brief History of Process Algebra " Theoretical Computer Science, vol. 335, 2005.

[10] CLEAVELAND, W.R; SMOLKA, S.A. "Process Algebra " Ed. Webster, J.G. Encyclopedia of Eletrical Engineering. John Wiley & Sons, 1999 http://ls14-www.cs.uni-dortmund.de/ls14Medien/Dokumente/Lehre/KTSMS/ee99_ps.ps - última visita 10/03/08

[11] BERGSTRA, J. A; KLOP, J. W. "An Introduction to Process Algebra " Ed. Baeten, J.C.M. Aplications of Process Algebra. Cambridge University Press, 2005. http://assets.cambridge.org/97805216/07506/excerpt/9780521607506_excerpt.pdf.

[12] TANENBAUM, S.A; WOODHULL, S. A. "Sistemas Operacionais – Projeto e Implementação." Bookman, 2000.