Descrição de Padrões de Cenário de Operação de Sistemas Embarcados e Aplicação no Desenvolvimento de Monitores

Alice M. Tokarnia (Orientadora), Emerson Cruz

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, CEP 13083-970 – Campinas, SP, Brasil

{torkarnia, ecruz}@dca.fee.unicamp.br

Abstract – Specification, performance analysis, and testing are key steps in the design of an embedded system. In many designs, these steps are independently executed and rely solely on the skills of the designers. The observation that same scenarios that specify system behavior are used in all these steps has led to software design methodologies that reduce design time. In this paper, the concept of scenario pattern is extended with a formal description that captures both functional and temporal behavior. An example of a scenario for a fire alarm system, which follows a pattern commonly found in several application domains, is presented. The objective of this formal description of scenario pattern is the development a configurable monitor that verifies system behavior using traces generated during tests or in the field.

Keywords – System specification, Scenario patterns, Trace monitor, Embedded systems.

1. Introdução

A especificação de sistemas embarcados geralmente inclui características funcionais e não-funcionais, que precisam ser repetidamente verificadas em várias etapas do desenvolvimento. Para reduzir o tempo de projeto, existem várias propostas de ferramentas que permitem reunir informações da especificação e utilizá-las na geração automática de software de verificação. A proposta deste trabalho é estender os trabalhos apresentados em [1], [2], [3] e [4], introduzindo uma notação para descrever uma variedade maior de padrões de cenários de operação e desenvolver um monitor configurável capaz de verificar os cenários de operação de um sistema.

Cada cenário de operação descreve uma interação entre o sistema e seu ambiente [1]. Como muitas interações podem ser descritas usando estruturas e elementos comuns, é possível associar muitos cenários a um mesmo padrão de cenário [2]. Associando a cada padrão de cenário de operação um padrão de software de teste, é possível obter a uma redução significativa no tempo de desenvolvimento de testes [3]. Conforme descrito em [3], apenas oito padrões de cenário foram suficientes para descrever aproximadamente 95% cenários de operação de um dispositivo médico implantável.

Muitos trabalhos fazem uso de padrões para descrever a especificação de requisitos de sistemas embarcados. Em [2], os autores utilizam uma linguagem natural estruturada para descrever cenários de operação e expressam requisitos de tempo real usando lógica temporal. Em [5], os autores introduzem padrões de operação baseados em linguagem natural. O objetivo é expressar os requisitos de um sistema usando construções padrões precisas e isentas de ambigüidades.

Este trabalho pode ser dividido em duas partes. A primeira parte consiste na elaboração de uma notação que permita descrever uma variedade de cenários de operação, incluindo requisitos funcionais e de tempo real. Na segunda parte, o objetivo é apresentar o núcleo de um monitor configurável a ser desenvolvido para verificar se os cenários especificados para um sistema são atendidos num rastro temporizado.

Este artigo está organizado da forma descrita a seguir. A seção 2 apresenta o modelo de descrição formal para o cenário de operação. A seção 3 fornece um exemplo de cenário de operação empregado em um sistema de detecção de incêndio. A seção 4 apresenta as informações do rastro e o algoritmo do monitor. A seção 5 traz as conclusões e os próximos passos deste trabalho.

2. Cenário de Operação

Para introduzir uma notação que permita a especificação de uma variedade de cenários de operação, foram utilizados conceitos em máquinas de estados programáveis (PSM) [6] e lógica temporal [7]. Os requisitos de tempo,

acrescentados a este modelo, são descritos por uma lógica temporal [7].

2.1 Elementos do cenário de operação

Os seguintes elementos são usados na descrição de um cenário de operação:

1. *Variáveis:* Correspondem às entradas, saídas, e estado. Uma variável V_i é descrita por <*função, tipo, domínio*>, onde a *função* pode ser entrada, saída, entrada/saída, estado; o *tipo* pode ser Booleano, inteiro, caractere, texto ou ponto flutuante e o *domínio* apresenta os valores possíveis na forma de conjunto ou intervalo. Todas as variáveis de estado podem ser lidas e que algumas podem também ser diretamente modificadas. O valor de V_i no tempo T_m é representado por $V_i(T_m)$. O tempo no qual a assume o valor *Val* é *Tempo*[V_i, *Val*]. Exemplos:
EstadoCentral : <estado, inteiro, {0, 1}>
EstadoCentral (20) = 0.

2. *Condição:* Descrita por expressões lógico-aritméticas usando variáveis, uma condição pode ser verdadeira (**V**) ou falsa (**F**).
Uma condição C_i é descrita por <[*elemento* 1] *relação* [*elemento* 2] >, onde [*elemento* i] é uma expressão usando variáveis e *relação* é um operador lógico-aritmético. O tempo no qual C_i se torna verdadeira é indicado por *Tempo*[C_i, **V**]. Exemplo:
$C_{sensor1}$: = < [SsrFumaça==1] || [SsrTérmico==1]>;

3. *Estado:* Um estado St_i é definido por um conjunto de condições sobre as variáveis de estado [St-V_1, St-V_2,⋯ St-V_n]. O estado determina relações entrada-saída e respostas a eventos. O estado do sistema no tempo T_m é dado por ST_SYS(T_m)= [St-V_1(T_m), St-V_2(T_m),⋯, St-V_n(T_m)]. representa o estado do sistema no tempo T_m. Exemplos: stMonitor := <EstadoCentral == 0>;
stAlarme := < EstadoCentral == 1 >.

4. *Evento:* Um evento E_i está associado a uma condição C-E_i sobre as variáveis de entrada e o tempo. Diz-se que o E_i ocorreu no instante *Tempo*[C-E_i, **V**] no qual a condição C-E_i se torna verdadeira. Um evento pode provocar uma transição de estado e/ou uma ação em um tempo posterior a *Tempo*[C-E_i, **V**]. Transição de estado e ação são definidas a seguir.
Exemplo: C-E_{sensor} := $C_{sensor1}$.

5. *Transição:* Mudança de estado ocasionada por um evento. Uma transição Tr_i é descrita por < C-E_i, St_a, St_b>, onde C-E_i representa a condição para que ocorra a transição St_a e St_b, são os estados inicial e final, respectivamente. A transição Tr_i ocorre atomicamente em *Tempo*[Tr_i] que pode ser posterior a *Tempo* [C-E_{if}, **V**]. Exemplo:
$Tr_{central}$:= < C-E_{sensor}, stMonitor, stAlarme >.

6. *Ação:* Atribuição de valores a um conjunto de variáveis de saída ocasionada por um evento. Uma ação A_i associada ao evento E_A é descrita por <C-E_A; [OV_1= Valor$_1$],⋯, [OV_k=Valor$_k$]> onde C-$E_{A\ i}$ é a condição, OV_i e *Valor*$_i$ são variáveis de saída e seus novos valores, respectivamente. A ação A_i ocorre atomicamente no *Tempo*[A_i], posterior a *Tempo*[C-E_A]. Exemplo: $A_{central}$:= < C-E_{sensor}, [Sirene = 1] >.

7. *Atividade:* Descreve a relação entrada-saída correspondente a um estado e a uma condição. A resposta do sistema é caracterizada por atribuições de uma seqüência de valores a variáveis de saída ao longo de um intervalo de tempo. Uma atividade At_i é descrita por <St_i; C_i; [OV= Val_1,Val_2,⋯,Val_n]; T >, onde os valores Val_1,Val_2,⋯,Val_n são atribuidos à variável de saída OV durante um intervalo de tempo T, quando o sistema se encontra no estado St_i e a condição C_i é satisfeita. Exemplo:
$At_{central}$ < stAlarme; $C_{sensor2}$; [Sprinkler=1]; 60s >.

8. *Tempo:* Conforme as definições anteriores o tempo é usado para caracterizar o comportamento do sistema. Na especificação de um cenário, apresentada a seguir, são estabelecidas condições sobre o tempo no qual as variáveis assumem valores. Todas as condições de tempo podem ser escritas na forma de lógica temporal. Exemplos:
Tempo [SsrFumaça, 1] – *Tempo* [SsrTérmico, 1] ≤ 2.

2.2 Cenários de operação

Um cenário de operação *CO* é descrito por dez elementos < *Id, Var, ST, Cond,* Stin, *Tc, Ac, Atvd, Tr, Tce, Te, Tend* >, divididos em quatro seções, conforme descrito a seguir.

Seção identificação:
Id é o número de identificação do cenário
Var é o conjunto de variáveis usadas na descrição do cenário;
ST é conjunto de estados usados na descrição do cenário;

Seção causa:
$\boldsymbol{St_{in}}$ é o estado inicial do cenário;

Cond é um conjunto de condições sobre variáveis de entrada e de estado, que caracteriza o cenário de operação $\{C_1,..., C_n\}$. Pode incluir eventos.
$\boldsymbol{T_C}$ é um conjunto de restrições sobre intervalos de tempo entre os elementos da seção causa;
Seção efeito:
Ac é um conjunto de ações $\{ A_1, \cdots, A_n\}$;
Atvd é um conjunto de atividades $\{ At_1, \cdots, At_n\}$; com seus tempos máximo e mínimo;
Tr é uma transição para o estado final $\boldsymbol{St_f}$;

$\boldsymbol{T_{CE}}$ é um conjunto de restrições sobre intervalos de tempo definidos entre um elemento da seção efeito e outro da seção causa;
$\boldsymbol{T_E}$ é um conjunto de restrições sobre intervalos de tempo entre os elementos da seção efeito;

Seção restrição de tempo de término:
$\boldsymbol{T_{end}}$ é prazo máximo para término do cenário.

3. Exemplo de Cenário

No cenário de operação descrito a seguir, a condição para a transição de estados é a ocorrência de dois ou mais eventos, em qualquer order.

A descrição textual do cenário: "A central de incêndio deverá sair do modo monitoramento para o modo incêndio e acionar a saída de sirene se os sensores termo-velocimétrico e de fumaça, que monitoram o ambiente, enviarem eventos de emergência, em qualquer ordem, num intervalo de até dois segundos."

Descrição formal

Seção identificação:
Id = 1.

Var = {EstadoCentral, SensorFumaça, SensorTérmico, Sirene};
EstadoCentral: < estado, inteiro, {0, 1}>;
SensorFumaça: < entrada, inteiro, {0, 1}>;
SensorTérmico: <entrada, inteiro, {0, 1}>;
Sirene: <saída, inteiro, {0, 1}>;

ST = {stMonitor, stIncedio}
stMonitor = < EstadoCentral == 0>;
stIncedio = < EstadoCentral == 1 >

Seção causa:
***Cond*:**
cSensorTérmico = < SensorFumaça ==1>
cSensorFumaça = < SensorTérmico ==1>

$\boldsymbol{St_{in}}$**:**
stMonitor
$\boldsymbol{T_C}$**:**
***Tc*-1:** | Tempo [*stMonitor* AND *cSensorTérmico*, **V**] - Tempo [*stMonitor* AND *cSensorFumaça*, **V**] | $\leq$ 2s ;

Seção efeito:
***AC*:**
AC-1 = < ***Tc*-1**, [Sirene = 1] >;

***Tr*:**
Tr-1 = < ***Tc*-1**, *stMonitor, stIncedio*>;

Seção restrição de tempo de término:
$\boldsymbol{T_{end}}$ **:**
T_{final} = 10s

Como a descrição textual não especifica até quanto tempo depois da condição ***Tc*-1** ser satisfeita devem ocorrer a transição de estado e ação de tocar a sirena, foi necessário arbitrar um prazo máximo para término do cenário.

4. Rastro e Monitor

O verificador possui dois arquivos de entrada um contendo o rastro de operação (*trace*) e o outro uma lista de cenários. O rastro de operação é o resultado de uma amostragem periódica das variáveis de entrada, saída e estado do sistema durante um intervalo de tempo. A linha *K* do rastro é constituída por $<T_K, V_1(T_K), V_2(T_K), \cdots, V_n(T_K)>$, onde T_K é o tempo correspondente a k amostragem e $V_i(T_K)$ é o valor da variável V_i no tempo T_K e *n* é o número de variáveis monitoradas. Os cenários da lista estão classificados em padrões. Os padrões diferem apenas no nome das variáveis e nos valores numéricos considerados nas condições. Para cada cenário padrão existe uma rotina de verificação específica.

O monitor apresenta como resultado um arquivo que lista para cada cenário as seguintes informações:

i) Identificação do cenário
ii) Descrição do cenário (copiada do arquivo de entrada).
iii) Número N_A de vezes que o cenário foi *acionado*, isto é, a seção causa do cenário foi satisfeita.

iv) Número N_S de vezes que o cenário foi *satisfeito*, isto é, tanto seção causa quanto a seção efeito do cenário foram satisfeitas.
v) Número N_I de vezes que não foi possível determinar se o cenário foi satisfeito ou não.
vi) Fração F_S de vezes que o cenário foi satisfeito, que é calculada como N_S /(N_A - N_I).

A Figura 1 apresenta o pseudocódigo geral para as rotinas de verificação de um cenário de um padrão e a Figura 2 apresenta o pseudocódigo para o monitor. Para cada cenário, o monitor faz uma chamada à rotina correspondente ao padrão deste cenário. A rotina percorre o trace para identificar quando as seções causa e efeito do cenário são satisfeitas. O monitor é configurado de forma a incluir apenas as rotinas correspondentes aos padrões de cenários utilizados na especificação do sistema.

Rotina de verificação dos cenários padrão *L*:

Para cada linha *k* no trace:
Se o estado inicial do cenário for satisfeito:
Para cada condição na seção causa do cenário:
Se a condição for satisfeita, anotar o tempo T_k para esta condição.
Se todas as condições estiverem satisfeitas usar os tempos anotados para verificar equações em T_C
Se todas as equações em T_C forem satisfeitas:
i) Incrementar N_A
ii) Verificar se a seção efeito do cenário é satisfeita até o tempo $T_k + T_{end}$.

Se seção efeito for satisfeita,
Incrementar N_S.
Se seção efeito não for satisfeita,
Se fim do trace antes de $T_k + T_{end}$
Incrementar N_I

Calcular $F_S = N_S /(N_A - N_I)$.

Figura 1. Rotina de verificação

Monitor

Para cada cenário do arquivo de entrada:
i) Ler cenário,
ii) Ler padrão correspondente a este cenário
iii) Determinar variáveis, estados e demais parâmetros.
iv) Chamar rotina de verificação do padrão, passando variáveis e parâmetros correspondentes.

Figura 2. Monitor

5. Conclusão

Este trabalho apresenta uma proposta de descrição formal para especificação de cenários de operação de sistemas embarcados, um exemplo de descrição de um cenário e uma descrição inicial para um monitor que utiliza padrões de cenários para analisar um rastro e determinar se cada cenário é satisfeito.

Os próximos passos neste trabalho são a determinação de um conjunto de padrões de cenários de operação, a codificação de rotinas de verificação e do monitor e a aplicação do monitor a sistemas reais.

Referências

[1] S. Some, R. Dssouli, J. Vaucher, "From Scenarios to Timed Automata: Building Specifications from Users Requirements," *Second Asia-Pacific Software Engineering Conf.*, p. 48, 1995.

[2] S. Konrad, B. Cheng, "Real-time specification patterns," *Proc. of the 27th International Conf. on Software Engineering*, p. 372, 2005.

[3] W. Tsai, L. Yu, F. Zhu, R. Paul, "Rapid Embedded System Testing Using Verification Patterns," *IEEE Software*, v. 22, p. 68, July/Aug. 2005.

[4] W. Tsai, L. Yu, R. Paul, X. Wei, "Rapid Pattern-Oriented Scenario-Based Testing for Embedded Systems," *Software Evolution with UML and XML*, 2004.

[5] C. Denger, D. M. Berry, E. Kamsties, "Higher Quality Requirements Specifications through Natural Language Patterns," *Proc. of the IEEE International Conference on Software-Science, Technology & Engineering*, p. 80, 2003.

[6] F. Vahid, T. Givargis, *Embedded System Design: A Unified Hardware/ Software Introduction*. Wiley, 2002.

[7] X. Chen, H. Hsieh, Y. Watanabe, F. Balarin, "Automatic trace analysis for logic of constraints," *Proc. of the 40th annual Design Automation Conference*, p. 460, 2003.