Verificação e Refinamento de Requisitos em Árvores de Características usando Linhas de Produtos de Requisitos e Redes de Petri

Carla Ferreira Fernandes[1], Liliane do Nascimento Vale[1]

Departamento de Ciência da Computação – Campus Catalão
Universidade Federal de Goiás (UFG)
Catalao – GO – Brasil

carlanandes@gmail.com, lili_malman@yahoo.com.br

Abstract: *To minimize the problem o phase requirements elicitation, this work proposes the use of the features' tree as part of the requirements elicitation process and a refinement is applied through the Product Line of requirements in order to eliminate ambiguous and inconsistent requirements. In the end, the requirements will be translated into the Petri Nets (Colorful and Ordinary) to allow the execution of differents scenarios and simulation software's interface.*

Resumo: *A fim de minimizar os problemas da fase de elicitação de requisitos,, este trabalho propõe o uso de árvore de características como parte do processo de elicitação dos requisitos e refinamento desta através das Linhas de Produtos de Requisitos. Ao final, os requisitos serão traduzidos para as Redes de Petri (Coloridas e Ordinárias) para permitir a execução e simulação de diferentes cenários de interface de software.*

1. Introdução

Muitas vezes as pessoas detentoras do conhecimento específico do software a ser construído não compreendem bem os requisitos essenciais que o sistema deveria conter e repassam informações inconsistentes ao especialista, o que torna a construção mais difícil provocando o re-trabalho.

Muitas técnicas de elicitação de requisitos se baseiam em informações textuais e por meio de linguagem natural, promovendo problemas de interpretação ou ainda empregando casos de uso, que é pobre em detalhes.

Baseado nesta realidade, este trabalho propõe o uso da metodologia de árvores de características como parte do processo de elicitação dos requisitos. Em seguida será realizado um refinamento na árvore de características para eliminar problemas de ambiguidade e omissão de informações, que são comuns em requisitos textuais. Para o gerenciamento da variabilidade na árvore de características, que possui o objetivo de evitar requisitos semelhantes ou duplicados, será utilizado as linhas de produto de requisitos, que resultará na árvore características final.

Com a versão final da árvore de características, essa será traduzida para as Redes de Petri (Coloridas e Ordinárias) para permitir a execução e simulação de diferentes cenários de interface. E, por fim, serão adicionadas as Redes de Petri Coloridas com expressões da Lógica Modal para avaliação de restrições.

2. Fundamentação Teórica

Uma das etapas fundamentais na construção de software é a fase de Levantamento de Requisitos. Segundo [Sommerville 2009], os requisitos de sistema definem detalhadamente, as funções, os serviços e as restrições operacionais do sistema.

Dentre as técnicas para elicitação de requisitos do usuário e/ou cliente, utilizaremos basicamente o modelo de árvore de características para representar os requisitos fornecidos. A árvore de características é usada para descobrir as características que o sistema deverá possuir.

Após realizar o processo de elicitação de requisitos a partir da árvore de características, esta deverá passar por um processo de refinamento usando Linhas de Produtos de Software (LPS), a fim de extrair requisitos inconsistentes e ou incompletos, pois a ocorrência de ambigüidade é provável [Pressman 2005].

Como forma de buscar maneiras eficazes de promover o reúso em requisitos de software, surge o conceito de Linhas de Produtos de Software (LPS), no final da década de 90. De acordo com [Gomma 2005], LPS é um conjunto de sistemas intensivos de software e possuem características em comum compartilhadas com outros softwares para satisfazer necessidades específicas de um ramo do mercado particular, minimizando riscos, custos, encurtando o tempo de entrega do sistema e proporcionando o aumento da qualidade do mesmo.

Neste trabalho, será criada a expressão Linhas de Produtos de Requisitos, onde esta possui, basicamente, a mesma idéia das LPS's, mas de forma que se adapte ao foco deste trabalho. Ao aplicar as Linhas de Produtos de Requisitos na árvore de características inicial, será feita uma análise dos requisitos a fim de gerenciar requisitos comuns e variáveis e manter requisitos obrigatórios, obtendo a árvore de características final, com requisitos funcionais reduzidos e completos.

Após a obtenção da árvore final, traduziremos a mesma para as Redes de Petri Coloridas (alto nível) e Ordinárias (baixo nível) para permitir a execução e simulação de diferentes cenários de interface. A escolha das Redes de Petri para formalização dos requisitos se deve ao fato desta ser uma poderosa técnica de modelagem na representação dos processos, permitindo a exibição de concorrência, paralelismo, sincronização, não-determinismo e exclusão mútua. Além disso, redes de Petri possibilitam uma representação matemática formal e disponibilizam mecanismos de análise que tornam possíveis verificar a correção do modelo e checar suas propriedades, além de permitir a simulação e execução do modelo, o que nenhum outro modelo permite [Cardoso e Valette 1997].

São formadas por quatro tipos de elementos: a transição, componente ativo, que corresponde a alguma ação realizada dentro do sistema; o lugar, passivo e relacionado a alguma variável de estado do sistema; a ficha, que define o estado de um sistema e o arco que interliga lugar e transição.

Graficamente, os lugares são representados por círculos e as transições por traços ou barras. Formalmente uma RdP é representadas por uma quádrupla, onde: RP = (P, T, F, M_0), tal que P é o conjunto finito de lugares; T é o conjunto finito de transições; F é o conjunto finito de arcos e M_0 é a marcação inicial.

Após construídas as RdP's a partir da árvore final de características, será aplicado conceitos de Lógica Modal utilizando expressões desta lógica na RdP Colorida. As Redes de Petri Modal (RdPM) serão aplicadas, pois apresentam uma representação gráfica de fácil entendimento além de estabelecer restrições para as transições das Redes de Petri. A escolha por usar Lógica Modal nas Redes de Petri, deve-se a vantagem desta oferecer meios de se obter capacidade de prova e dessa forma garantir a corretude do sistema ou das funções do sistema. Além do fato de que se pode

representar questões que envolvem modalidades como possibilidade e necessidade. O que não existe em outra forma de representação da mesma.

3. Trabalhos Relacionados

A modelagem de sistemas tem sido cada vez mais discutida no contexto de formalização de software das mais diversas naturezas. Em [Oliveira 2009], a autora propõe o uso do modelo de árvore de características como forma de elicitação de requisitos e em seguida é apresentada a tradução desta árvore para uma Rede de Petri Colorida com expressões lógicas aplicada ao conceito de Redes de Petri e Lógica Modal.

No contexto de técnicas de modelagem e Lógica Modal, em [Carvalho 2009] é proposto a modelagem de Interfaces de software utilizando árvore de características e Lógica Modal para definir uma nova forma de modelar projetos de interface de software. A autora também propõe uma comparação do método de modelagem com outros métodos usuais de Engenharia de Software, como por exemplo, Redes de Petri.

Em consideração as Redes de Petri Coloridas, no trabalho de [Prata 2006] propõe elaborar um modelo de avaliação de desempenho operacional em terminais portuários, baseado nas Redes de Petri Coloridas, tendo como principal variável de decisão o tempo total de deslocamento das cargas utilizadas em um porto. Utilizou-se o programa CPNtools, editor e simulador de Redes de Petri Coloridas.

No texto de [Silva et al 2004] apresenta uma solução de desenvolvimento sistemático de modelos de sistemas de manufatura aliado ao reúso de projetos bem sucedidos e às Redes de Petri coloridas, reduzindo o tempo de desenvolvimento do mesmo, de forma rápida, sistemática e segura.

Dessa forma, verifica-se que a análise de requisitos dentro do contexto da atividade de eliminar ambigüidades e inconsistências ainda é feita sob meios informais consistindo, na maioria das vezes, de somente uma revisão textual. Entretanto, quando aplicado a árvore de característica para identificar os requisitos e refinamentos sucessivos por meio de LPS, esta contribui para o controle de requisitos inconsistentes, duplicados e ambíguos.

4. Redes de Petri Modais (RdPM)

Para representar de forma simplificada as Redes de Petri Coloridas com Expressões Lógicas, empregando a Lógica Modal com o sistema k, é criado um novo modelo denominado Redes de Petri Modal, onde, cada lugar no modelo é uma atividade, definindo assim o conjunto de lugares: $L = \{L_1, L_2 ... L_n\}$; as transições definem o conjunto: $T = \{T_1, T_2 ... \quad T_n\}$; os arcos ligam lugares em transições e transições em lugares; lugares, arcos e transições são representados conforme na Figura 1.

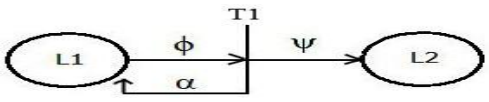

Figura 1 – Modelo de Rede de Petri Modal

Os lugares são representados pelas circunferências, simulando a atividade do sistema. As transições são representadas pela barra vertical (*T1*)e os arcos são ligados as transições. As transições seguem um esquema do tipo: $< L_1, \Phi > \rightarrow \{< \psi, L_2>, < \alpha, L_1\}$, no qual temos que ψ, ϕ e α são cláusulas (conjunções de literais) do tipo: ψ: $p \wedge \neg q \wedge r$; ϕ: $\neg p$; α: $q \wedge \neg r$.

A adoção da Lógica Modal neste trabalho se deve ao fato desta impor restrições às transições, acrescentando assim, maior correção comportamental [Fairlough e Mendler 1997]. Desta forma, exemplifica-se, de forma geral, a relação existente entre Redes de Petri, Lógica Modal e Atividade (ou função) de um sistema. A associação de uma linguagem lógica às Redes de Petri se deve a presença de uma linguagem criada por expressões bem formadas, sobre as quais podem atribuir significados. Isso inclui símbolos da linguagem, juntamente com os conjuntos de regras formais para composição de expressões bem formadas, denominado por sintaxe da linguagem. Já a semântica define como interpretar expressões bem formadas sobre um universo, que pode ser composto de gráfico, números naturais, entre outros.

A Lógica Modal surgiu de uma tentativa de demonstrar a noção de correção para restrições comportamentais [Fairlough e Mendler 1997]. Em outras palavras é uma lógica voltada para a habilidade de lidar com modalidades, como por exemplo, modos de tempo, possibilidade e probabilidade. Sua semântica desdobra a semântica da Lógica proposicional clássica e considera dois operadores modais: diamante, e caixa, representados pelos símbolos $\Diamond$, $\Box$ respectivamente. O operador caixa, nos diz algo que vai necessariamente acontecer, enquanto o operador diamante nos diz se algo vai possivelmente ocorrer.

O sistema modal k (que advém da semântica de kripke) é o menor dos sistemas modais normais e contém apenas as tautologias proposicionais. Trata de um conjunto de axiomas e regras de inferência que representam formalmente o raciocínio válido e é fechado para tese necessitação, isto é, se A é uma tese $\Box$A.

5. Estudo de caso

Por ser um objeto de uso comum à maioria e por possuir facilidades em seu uso, o que facilita a didática, o forno elétrico será adotado como objeto do estudo de caso. Uma observação a ser feita sobre a árvore de características é a sua simplicidade. As características são apresentadas de forma hierárquica e estruturada, como observado na Figura 2:

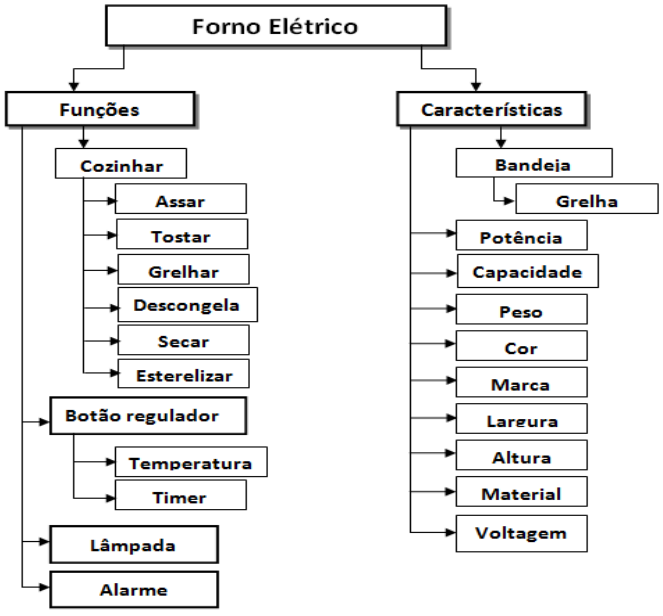

Figura 2: Árvore de características de um Forno Elétrico

Após construída a árvore de característica, esta passará por um processo de gerenciamento das similaridades e variabilidades utilizando as Linha de Produto de Requisitos (LPR), dentro do conceito de Linha de Produto de Software (LPS).

Em [Kang et al. 1990] define que as características são de três tipos: mandatórias, alternativas e opcionais. As mandatórias são aquelas que são disponíveis em todas as aplicações de uma família de sistema, no caso do forno elétrico ela pode ser representada pela característica "*cozinhar*". As opcionais representam as variabilidades existentes na família de sistema, as quais podem ser incluídas ou não, como por exemplo a característica "*lâmpada*". As alternativas permitem uma seleção onde apenas uma característica pode ser selecionada em um conjunto de características, como por exemplo "*cor*".

Para obter um claro entendimento sobre os requisitos funcionais, será empregado um diagrama de Casos de Uso da UML, fazendo a completa descrição do que o sistema de Forno Elétrico faz em um determinado domínio, como mostrado na Figura 3.

Figura 3: Diagrama de Caso de Uso

Para auxiliar a elaboração das descrições textuais dos casos de uso, além da representação gráfica Gomma, em [Gomma 2005] oferece um template para o gerenciamento das variabilidades e similaridades que será abordado neste trabalho. O Algoritmo 1 mostra um trecho da descrição textual do caso de uso "Cozinhar" da LPR de um forno elétrico. Nesta descrição, o ponto de variação *Alarme* pode ocorrer na linha 9, adicionando opcionalmente mais funcionalidade ao caso de uso.

Algoritmo 1: Trecho da Descrição textual de Caso de Uso da LPS de Forno Elétrico

Nome do caso de Uso: Cozinhar
Categoria de reúso: Kernel
Resumo: Usuário insere a comida no forno e o forno cozinha o alimento inserido.
Pré-condição: Forno pré-aquecido
Descrição:

1. Usuário abre a tampa (t1), põe a comida no forno (t2), fecha a tampa (t3).
2. Alimento pronto pra ser cozido (t4)
3. Usuário regula o botão de temperatura. (t5)
4. Sistema acende a luz que indica seu funcionamento (t6)
5. Sistema verifica o tempo em que atinge a temperatura desejada (t7)
6. Sistema apaga a luz (t8)
7. Usuário define o tempo de cozimento (t9)
8. Sistema rola o botão de timer até chegar ao ponto inicial (t10)
9. Sistema inicia o cozimento do alimento (t11)
10. Sistema desliga automaticamente ao acabar o tempo solicitado (t12)
11. Usuário abre a tampa(t13), remove a comida do forno (t14) e fecha a

tampa (t15)
Nome: Alarme **Tipo de funcionalidade:** Opcional **Número(s) de linha(s):** 9 **Descrição da funcionalidade:** Se o sistema possuir a função alarme, ele apita ao iniciar o cozimento e também ao término do cozimento, indicando que o alimento está pronto.

Ao construir a tabela da descrição textual da LPS "cozinhar", esta facilitará a construção da Rede de Petri, uma vez que as transições, representadas por *t*, são bem definidas. Nessa representação, cada transição é associada a uma implicação lógica do tipo (A → B). A Rede de Petri com expressões da Lógica Modal apresenta uma representação gráfica de fácil entendimento além de estabelecer restrições para as transições das redes de Petri.

A Rede de Petri Ordinária e Colorida com expressões da Lógica Modal correspondente a Tabela 1, podem ser observadas, respectivamente, pela Figuras 4 e Figura 5.

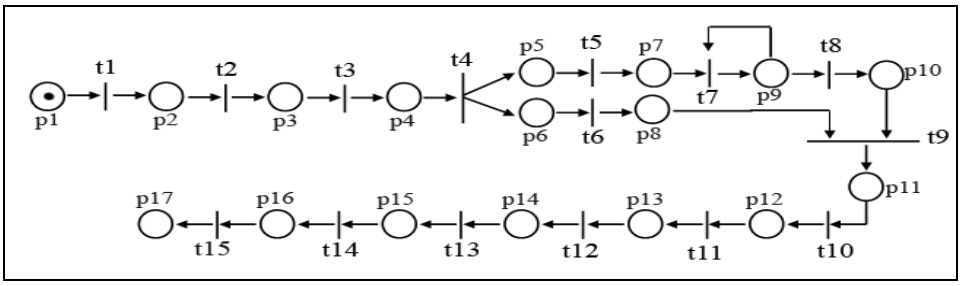

Figura 4: Rede de Petri Ordinária

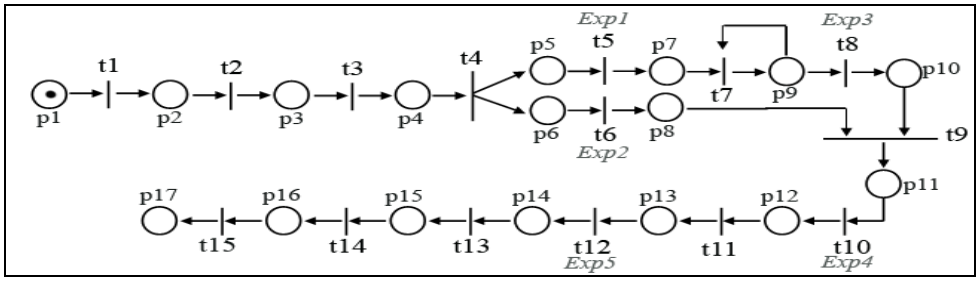

Figura 5: Rede de Petri Colorida com Expressões da Lógica Modal

Para a execução da Rede de Petri Ordinária (Figura 4), será dada a sequência dos disparos apresentados a seguir. A marcação inicial (ficha que está associada ao lugar p1) sensibiliza a transição t1, que não possui pré-condição. O disparo de t1 indica que a tampa foi aberta pelo usuário. Após o disparo de t1, um novo estado é produzido no lugar, estando a ficha agora em p2.

Com uma ficha no lugar p2, a transição t2 é sensibilizada. O disparo de t2 indica que a comida foi colocada no forno. Após o disparo de t2, um novo estado é produzido, permanecendo a ficha agora em p3.

Com uma ficha no lugar p3, a transição t3 é sensibilizada. O disparo de t3 indica que a tampa do forno é fechada. Após o disparo de t3, um novo estado é produzido, permanecendo a ficha agora em p4.

Com uma ficha no lugar p4, a transição t4 é sensibilizada. O disparo de t4 indica que o alimento está pronto para ser cozido. Após o disparo de t4, dois novos estados são produzidos, ocorrendo o paralelismo. Agora há 2 fichas, uma em p5 e outra em p6, indicando que as transições t5 e t6, (usuário regula botão de temperatura e sistema acende a luz que indica seu funcionamento, respectivamente) acontecem concorrentemente. Até que o sistema inicie o cozimento do alimento, haverá a concorrência das transições e estados. Após o disparo de t5, um novo estado é produzido, permanecendo a ficha agora em p7. Da mesma forma, após o disparo de t6 um novo estado é produzido, permanecendo a ficha agora em p8.

Com uma ficha no lugar de p7, a transição t7 é sensibilizada. O disparo de t7 indica que o sistema está verificando o tempo em que atinge a temperatura desejada. Após o disparo de t7, um novo estado é produzido, permanecendo a ficha agora em p9.

Com uma ficha no lugar p9, as transições t7 e t8 são sensibilizadas, mas somente a transição t8, que indica se a temperatura desejada foi alcançada, é satisfeita. Após o disparo de t8 um novo estado é produzido, permanecendo a ficha agora em p10.

Com uma ficha no lugar p10 e no lugar p8, a transição t9 é sensibilizada. O disparo de t9 indica que o usuário definirá o tempo de cozimento do alimento. Após o disparo de t9, um novo estado é produzido, permanecendo a ficha agora em p11.

Com uma ficha no lugar p11, a transição t10 é sensibilizada. O disparo de t10 indica que o botão de timer do sistema está rolando até chegar ao ponto inicial. Após o disparo de t10, um novo estado é produzido, permanecendo a ficha agora em p12.

Com uma ficha no lugar p12, a transição t11 é sensibilizada. O disparo de t11 indica que o alimento está sendo cozido. Após o disparo de t11, um novo estado é produzido, permanecendo a ficha agora em p13.

Com uma ficha no lugar p13, a transição t12 é sensibilizada. O disparo de t12 indica que o tempo solicitado foi alcançado. Após o disparo de t12, um novo estado é produzido, permanecendo a ficha agora em p14.

Com uma ficha no lugar p14, a transição t13 é sensibilizada. O disparo de t13 indica que o usuário abriu a tampa. Após o disparo de t13, um novo estado é produzido, permanecendo a ficha agora em p15.

Com uma ficha no lugar p15, a transição t14 é sensibilizada. O disparo de t14 indica que o usuário remove a comida do forno. Após o disparo de t14, um novo estado é produzido, permanecendo a ficha em p16.

Com uma ficha no lugar p16, a transição t15 é sensibilizada. O disparo de t15 indica que o usuário fecha a tampa do forno. Após o disparo de t15, um novo estado é produzido, permanecendo a agora no último estado, p17. Assim encerra sua execução.

Na figura 5, a sequência da execução da Rede de Petri é a mesma da Ordinária, porém há expressões da Lógica Modal estabelecendo restrições comportamentais à Rede de Petri Colorida, representadas por *Exp*. Tais transições são do tipo:

- *Exp1* – Necessariamente a tampa será fechada, então necessariamente usuário regula botão de temperatura. Sendo k = tampa fechada e L = botão de temperatura regulado, temos: $\Box k \rightarrow \Box L$
- *Exp2* - Necessariamente o botão de temperatura será regulado, então necessariamente a luz vai apagar. Sendo: m = botão de temperatura regulado e n = luz apagada, temos: $\Box m \rightarrow \Box n$
- *Exp3*- Necessariamente a temperatura desejada vai ser alcançada, então necessariamente a luz vai apagar. Sendo: p = Temperatura desejada e q = luz apagada, temos: $\Box p \rightarrow \Box q$

- *Exp4* - Necessariamente o timer será definido, então necessariamente o botão será rolado até o ponto inicial. Sendo: r = timer será definido e s = botão rolado, temos: □ r → □ s
- *Exp5* – Necessariamente o tempo solicitado será atingido, então necessariamente o sistema desliga.Sendo: t = tempo solicitado e u = sistema desligado, temos: □ t → □ u

6 . Conclusão e Trabalhos futuros

Neste trabalho foi apresentado como a extração de requisitos pode acontecer de modo eficiente. Inicialmente utilizamos como forma de elicitação para o estudo do problema do forno elétrico, a árvore de características, que é um modelo simples e de fácil entendimento. As Linhas de Produtos de Software com foco nos requisitos, foi empregada a fim de diferenciar requisitos mandatórios, alternativos e opcionais, o que contribuiu para eliminação de requisitos ambíguos, inconsistentes e duplicados.

A tabela de descrição textual do caso de uso da LPS forno elétrico, permitiu de modo claro e objetivo, a visualização de como ocorre a descrição de cada variável do sistema. Foi construído a Rede de Petri Ordinária e Colorida com expressões da Lógica Modal, originando as Redes de Petri Modais, que é um método formal de modelagem, verificação e simulação de software. As expressões da Lógica Modal na Rede de Petri Colorida foram utilizadas para estabelecer restrições a determinados disparos da transição da rede.

Como trabalho futuro, será construído um protótipo para validar o modelo de Redes de Petri Modal, permitindo assim, a simulação de diferentes cenários de interface.

7. Referências

Cardoso, J. & Valette, (1997) R. Redes de Petri. Editora da UFSC, Santa Catarina, Brasil, 1997.

Fairlough, M., Mendler, M., (1997) "Propositional Lax Logic". Information and Computation, 137(1):1–33

Gomma, H., (2005) Designing Product Lines with UML: From Use Casesto Pattern-Based Architectures, Addison-Wesley.

Kang, K. *et al*. Feature-oriented domain analysis (FODA) feasibility study. Pittsburgh:Software Engineering Institute, Carnegie Mellon University, 1990. (CMU/SEI-90-TR-021,ADA235785).

Oliveira, C.C. (2009) Uma análise de metodologias formais, baseadas em Redes de Petri para modelagem de software. Dissertação de Mestrado.

Prata, B. (2006) Avaliação de desempenho operacional de terminais portuários de carga utilizada: Uma aplicação de redes de Petri Coloridas. Universidade Federal do Ceará.

Pressman, R. S. (2005) Engenharia de Software. São Paulo: Pearson Makron Books.

Sommerville, L. (2009) Engenharia de Software. 8ª Ed. São Paulo: Pearson Addison Wesley.