SPLOT-TM: Apoio ao Gerenciamento de Testes Funcionais em Linha de Produto de Software

Regis H. Hattori, Marcelo Fantinato, Marcelo J. G. Faria, Marcos L. Chaim

Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)

Rua Arlindo Béttio, 1000 – 03828-000 – São Paulo – SP – Brasil

{regis.hattori, m.fantinato, marcelo.jose.faria, chaim}@usp.br

Abstract. *Product Line (PL) is a Software Engineering's approach that seeks to increase quality and to reduce cost and time-to-market of information system development. However, due to its particularities, this approach requires specific techniques and tools for software testing. This paper proposes a tool to support the management of functional tests for PL, based on Feature Modeling, developed as an extension of the SPLOT tool.*

Resumo. *Linha de Produto (LP) é uma abordagem da Engenharia de Software que busca aumento da qualidade e redução no custo e no tempo de desenvolvimento de sistemas de informação. No entanto, tal abordagem possui particularidades que requerem técnicas e ferramentas específicas de teste de software. Este artigo propõe uma ferramenta de apoio ao gerenciamento de testes funcionais para LP, baseados em Modelos de Características, desenvolvida como uma extensão da ferramenta SPLOT.*

1. Introdução

A indústria de software tem buscado a adoção de técnicas sistemáticas para o desenvolvimento de sistemas de informação com maior qualidade e produtividade. Nesse contexto, abordagens como Linha de Produto (LP) de Software têm sido aplicadas, pois podem contribuir significativamente para a melhoria dos sistemas de informação desenvolvidos, mesmo considerando contextos de alta complexidade de produto, de processo e de equipes [Pohl *et al.* 2005].

LP é uma abordagem da Engenharia de Software que trata o desenvolvimento de software dividido em dois processos – Engenharia de Domínio e Engenharia de Aplicação – de forma a proporcionar maior reúso de artefatos e, consequentemente, aumento da qualidade e redução no custo e no tempo de desenvolvimento. A Engenharia de Domínio envolve a definição e o desenvolvimento de artefatos reusáveis, estruturados por meio de pontos comuns e variáveis, além de uma plataforma comum, que oferece uma estrutura para os componentes. A Engenharia de Aplicação envolve o desenvolvimento de produtos específicos por meio da exploração das variabilidades dos artefatos reusáveis desenvolvidos na Engenharia de Domínio [Pohl *et al.* 2005].

No contexto de processo de desenvolvimento de sistemas de informação, o teste de software é uma das atividades mais importantes para o desenvolvimento de produtos com maior qualidade e confiabilidade, entretanto, Pressman (2006) e Delamaro, Maldonado e Jino (2007) relatam que essa atividade frequentemente responde por altos custos dentro do orçamento de desenvolvimento. Pressman (2006) destaca ainda que, se

realizada sem estratégias sistemáticas e bem definidas, a etapa de teste pode desperdiçar parte do orçamento sem que se alcancem os benefícios desejados.

Apesar das vantagens de LP, seus benefícios ainda não foram alcançados especificamente no teste de software de forma plena [Silveira *et al.* 2011; Pohl e Metzger 2006]. LP requer estratégias de testes adaptadas a seu contexto, o que torna o teste um gargalo para LP [Käköla e Dueñas 2006]. Paralelamente, do ponto de vista de apoio computacional, Neto, Almeida e Meira (2012) relatam a ausência de ferramentas que forneçam apoio a todo o ciclo de vida da LP chegando, portanto, até as diferentes tarefas da atividade de teste de software. Ferramentas específicas para a atividade de teste dentro de LP também são ainda bastante raras, como está apresentado neste artigo em seção apropriada de trabalhos relacionados. Isso ocorre provavelmente porque a própria teoria dessa área não está ainda bem solidificada, como pode ser constatada pela revisão sistemática de Narciso, Nunes e Delamaro (2011).

Dentro desse contexto, este artigo apresenta uma proposta de ferramenta de gerenciamento de testes de software funcionais para LP, chamada SPLOT-TM (SPLOT – Test Management), por meio da extensão da ferramenta SPLOT (*Software Product Line Online Tools*) desenvolvida por Mendonça e Branco (2009). A SPLOT-TM possui como diferencial o uso de Modelos de Características (MC) – uma das principais técnicas empregadas no gerenciamento de variabilidades em LP – na atividade de teste de software em LP, possibilitando a realização de testes nos níveis de Engenharia de Domínio e Engenharia de Aplicação de forma flexível e alinhada aos conceitos da área.

Para isso, este artigo apresenta as seguintes seções: visão geral da técnica de gerenciamento de variabilidade e da ferramenta SPLOT, usadas como base para o desenvolvimento da SPLOT-TM; apresentação das funcionalidades da ferramenta desenvolvida, realizada por meio de exemplos que foram feitos com o objetivo de verificar seu uso; análise de trabalhos relacionados; breve discussão do trabalho apresentado; e, conclusão do artigo.

2. Modelos de Características e Ferramenta SPLOT

Uma das técnicas mais relevantes para representar pontos comuns e variáveis em LP são os Modelos de Características (MC) [Czarnecki, Helsen e Eisenecker 2005]. MCs descrevem, por meio de diagramas, propriedades de sistema que sejam relevantes aos interessados em diferentes níveis de abstração, tratando principalmente da representação dos pontos comuns e dos pontos variáveis. Em MCs, as funcionalidades da LP são expressas via Características inter-relacionadas, representadas em uma estrutura de árvore, em que a raiz representa um conceito principal e os nós descendentes representam conceitos derivados [Sousa e Fantinato 2010]. Em LP, o MC é elaborado durante a Engenharia de Domínio, e uma nova configuração desse MC é gerada para cada nova instância do produto a ser criada durante a Engenharia de Aplicação.

Existem várias notações para representar variabilidades de Características em MCs. Na Figura 1 é apresentado um exemplo de MC seguindo a notação gráfica adotada pela ferramenta SPLOT, usada como base para este trabalho, que representam:

-  : uma Característica raiz;
- : uma Característica obrigatória a todas as configurações do MC;
- : uma Característica opcional, que para cada Configuração do MC pode ser selecionada ou não;

– [1..1]: um grupo de Características alternativas do tipo "ou exclusivo" que, para cada Configuração do MC, exatamente uma delas deve ser selecionada;
– [1..*]: um grupo de Características alternativas do tipo "ou inclusivo" que, para cada Configuração do MC, pelo menos uma delas deve ser selecionada;
– : uma Característica alternativa, filha de um grupo de Características.

SPLOT é uma ferramenta web de código aberto que oferece dois principais serviços: Criação e Configuração de MCs, e Raciocínio Automatizado [Mendonça e Branco 2009]. Apenas a parte de criação e configuração de MCs foi usada na SPLOT-TM. Ela foi desenvolvida em Java, usa uma interface HTML interativa, e conta com um repositório de MCs. Uma limitação da versão atualmente disponibilizada da ferramenta é a ausência de um repositório de configurações de MC, os quais podem apenas ser exportados, mas não gravados para recuperação posterior.

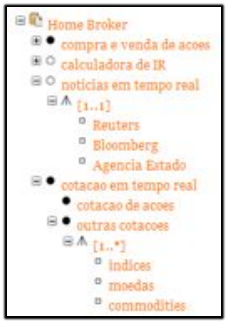

Figura 1. Exemplo de MC pela notação da ferramenta SPLOT

3. Apresentação da Ferramenta SPLOT-TM

A SPLOT-TM foi projetada e desenvolvida para apoiar o gerenciamento de testes funcionais no contexto de LP com base em MCs, tendo sido implementada como uma extensão da ferramenta SPLOT apresentada na Seção 2. O objetivo da SPLOT-TM é apoiar sistematicamente as diferentes tarefas de teste de software, incluindo o projeto, a execução e a análise de teste, todas com base em MCs no contexto de LP, ou seja, tanto para a Engenharia de Domínio quanto para a Engenharia de Aplicação.

A ferramenta SPLOT foi escolhida como base para esse desenvolvimento por permitir que Projetos de Teste (PjT) funcionais pudessem ser elaborados: (i) com base em MCs que representam as funcionalidades do sistema sendo desenvolvido; e, (ii) com estrutura similar a MCs, já que eles também são inerentemente representados por estruturas hierárquicas do tipo árvore, por meios de cenários, casos e passos de teste.

A Figura 2 apresenta, por meio da notação BPMN, as tarefas projetadas para a SPLOT-TM. O subprocesso Elaborar PjT, presentes nas duas raias, é detalhado na parte inferior da figura. Considerando a interatividade dessa atividade, a partir de cada tarefa, é possível retornar a qualquer outra tarefa já executada; porém, em prol da simplicidade, apenas o fluxo básico do processo é apresentado.

O fluxo se inicia na Engenharia de Domínio, com a tarefa Criar MC. A partir de um MC criado, é possível realizar as tarefas de teste da Engenharia de Domínio, ou realizar a tarefa Configurar MC, para que seja então possível realizar as tarefas de teste da Engenharia de Aplicação. As tarefas associadas a MCs são realizadas pela ferramenta SPLOT, com adaptações.

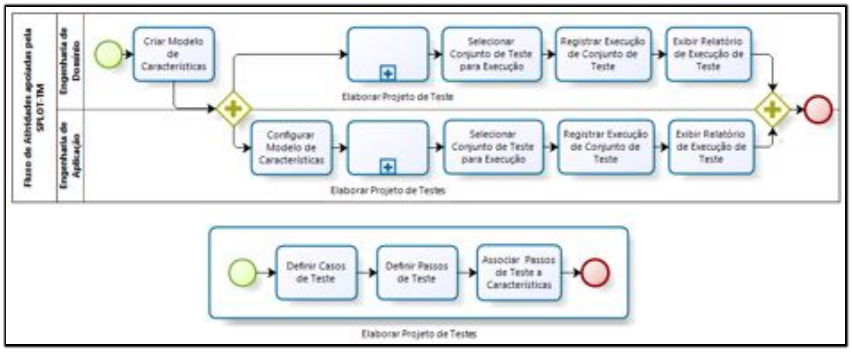

Figura 2. Fluxo básico do processo apoiado pela SPLOT-TM

As tarefas de teste dos dois processos da LP são estruturalmente as mesmas. Porém, enquanto o PjT na Engenharia de Domínio é elaborado com base no MC, os PjT's na Engenharia de Aplicação são elaborados com base nos MCs configurados, representando instâncias do PjT da Engenharia de Domínio. Essas relações são apresentadas pelo Diagrama de Classes da Figura 3.

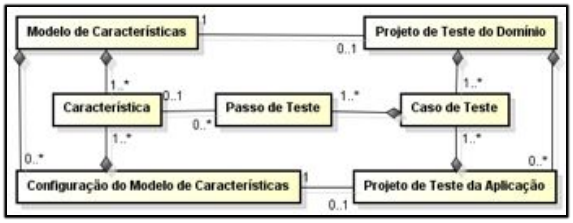

Figura 3. Relacionamento entre PjT do Domínio e PjT da Aplicação

Tanto o PjT do Domínio quanto o PjT da Aplicação é formado por um conjunto de Casos de Teste (CsT), que por sua vez são formados por um conjunto de Passos de Teste (PsT). Considerando que, do ponto de vista de estratégia de teste para LP, há deferentes formas para se realizar o teste em cada um dos dois processos da LP [Pohl e Metzger 2006; Silveira *et al.* 2011], a SPLOT-TM foi projetada visando flexibilidade nesse contexto com o objetivo de apoiar o gerenciamento de teste independentemente da estratégia escolhida pelos analistas de teste. Assim, ambos os PjT's são opcionais, tanto no nível de domínio quanto no nível de aplicação; além disso, o conjunto de CsT's no nível de aplicação pode ser ou não um subconjunto do nível de domínio.

A SPLOT-TM visa também apoiar a elaboração de PjT's com base em Características: a associação de CsT a Característica, seja no nível de domínio seja no nível de aplicação, é sempre realizada via um PsT. Para o PjT do Domínio, seus CsT's podem ser associados a Características pertencentes ao MC da LP; enquanto que, para o PjT da Aplicação, seus CsT's podem ser associados a Características pertencentes à Configuração do MC, que é um subconjunto do MC da LP.

A Figura 4 apresenta, por meio de outro Diagrama de Classes, o detalhamento das entidades usadas pela SPLOT-TM em seu funcionamento completo, de acordo com o fluxo apresentado na Figura 2, partindo das entidades parcialmente apresentadas na Figura 3. O detalhamento é apresentado apenas para a Engenharia de Domínio, visto sua equivalência para a Engenharia de Aplicação.

Segundo a Figura 4, diferentes Conjuntos de Testes (CjT) podem ser associados a cada PjT. Cada CjT contém um subconjunto dos CsT's que fazem parte do PjT em

questão. Esses diferentes CjT's podem ser executados pelos testadores para um mesmo PjT em diferentes momentos de acordo com a estratégia de teste usada. Para a seleção dos CsT's do PjT que devem fazer parte de um determinado CjT, um Critério de Seleção deve ser usado, a ser escolhido entre: os critérios de teste funcional baseados em MCs para LP propostos por Sousa e Fantinato (2010), considerando apenas os "por tipo de Característica"; e o critério "livre", em que nenhum controle de cobertura é realizado. Para cada sessão de execução de testes realizada para um CjT, um Registro de Execução do CjT é criado. Para cada CsT do CjT em execução, deve haver um Resultado de Execução do Passo do CsT para cada Registro de Execução do CjT criado.

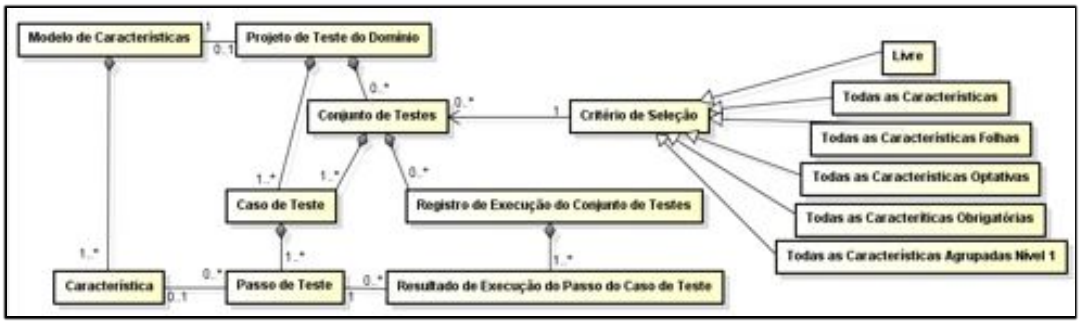

Figura 4. Entidades usadas pela SPLOT-TM para a Engenharia de Domínio

Nas subseções a seguir, são apresentadas informações mais completas sobre a SPLOT-TM, seguindo as tarefas previstas na Figura 2, com exceção da última tarefa de cada processo da LP, relativa aos relatórios de execução, por ainda não terem sido implementadas na versão atual da ferramenta SPLOT-TM sendo apresentada.

3.1. Criar Modelo de Características e Configurar Modelo de Características

As duas primeiras tarefas de teste, uma para cada um dos processos da LP, previstas na Figura 2, são realizadas com o apoio da ferramenta SPLOT original: a criação de MCs e a Configuração de MCs. Especificamente no caso da Configuração de MCs, a ferramenta SPLOT precisou ser estendida, como parte dos esforços para o desenvolvimento da SPLOT-TM, para permitir que tais configurações pudessem ser armazenadas no repositório e posteriormente recuperadas. Sem essa extensão, não seria possível realizar as tarefas de teste projetadas para a Engenharia de Aplicação.

A Figura 1 apresentou um exemplo da tarefa Criar MC, enquanto a Figura 5(a) apresenta um exemplo da tarefa Configurar MC, associado ao MC da Figura 1. Os ícones existentes na Configuração do MC apresentado na Figura 5(a), conforme a SPLOT original, são usados para: representar uma Característica selecionada pelo usuário; representar uma Característica selecionada automaticamente pela ferramenta, como o caso de Características obrigatórias; representar a ação disponível para seleção de uma Característica por parte do usuário; e, representar a ação disponível de desseleção de uma Característica por parte do usuário.

A Figura 5(b) apresenta a tabela de recuperação das configurações dos MCs, desenvolvida no escopo deste trabalho. Para o exemplo apresentado: o MC Splot não possui ainda nenhuma configuração criada (já que não há um botão de expansão – ⊞ – associado a ele); O MC Home Broker possui duas configurações, destacadas em uma subtabela logo abaixo dele na figura; e, o MC Home Banking possui pelo menos uma configuração, que está oculta (já que há um botão de expansão – ⊞ – associado a ele).

Figura 5. (a) Exemplo de configuração de MC; (b) Tabelas para recuperação

3.2. Elaborar Projeto de Teste

O subprocesso Elaborar PjT é o principal grupo de tarefas apoiado pela SPLOT-TM, usado de forma bastante similar tanto na Engenharia de Domínio quanto na Engenharia de Aplicação. Conforme exemplo da Figura 6(a), um PjT na SPLOT-TM é elaborado de forma bastante similar a um MC na SPLOT; por isso, a estrutura interna de elaboração de MCs da ferramenta SPLOT foi aproveitada em sua extensão para a SPLOT-TM.

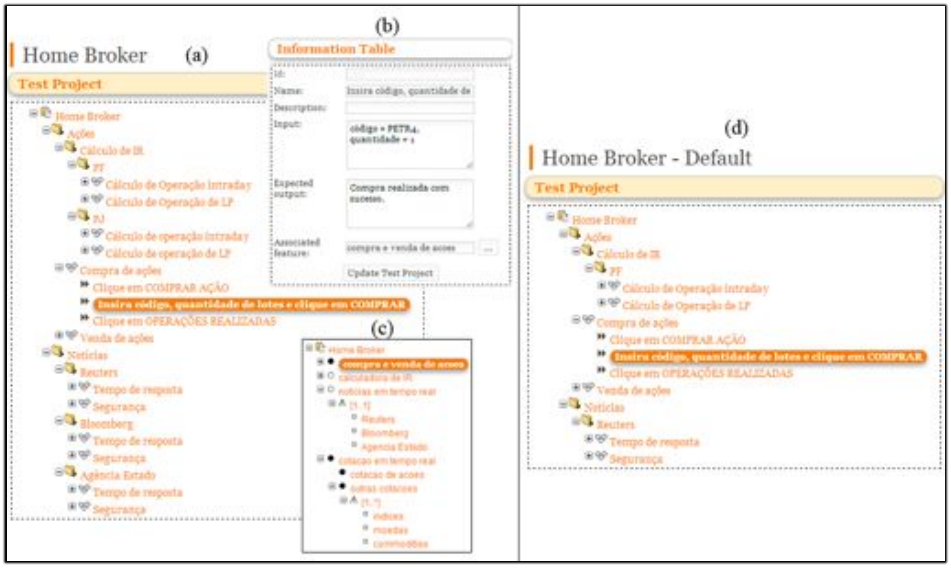

Figura 6. Exemplos de PjT's: (a) PjT do Domínio; (b) Tabela de informações do PjT do Domínio; (c) Associação de PsT com Característica; (d) PjT da Aplicação.

O exemplo da Figura 6(a) apresenta um PjT do Domínio associado ao MC Home Broker da Figura 1. Os ícones apresentados no PjT são usados para representar: – o PjT; – o Cenário de Teste (CnT); – o CT; e, – o PsT. O conceito de CnT se refere apenas a um agrupamento de CsT's ou de outros CnT's e, por isso, não é representado formalmente no metamodelo da Figura 4.

A Figura 6(b) apresenta a Tabela de Informações, que contém os detalhes do item selecionado na Figura 6(a). Sempre que o item selecionado for um PsT, como nesse exemplo (Insira código, quantidade de lotes e clique em COMPRAR), os campos Input e Expected output são apresentados para edição e consulta; o conjunto de "entradas" e "resultados esperados" de todos os PsT's de um CsT resultam nas

"entradas" e "resultados esperados" desse CsT. Além disso, o campo Associated Feature apresenta a Característica com a qual o PsT está associado, de acordo com o relacionado apresentado nas Figuras 3 e 4. Caso ainda nenhuma Característica tenha sido associada ao PsT em questão ou deseje-se trocar a Característica associada por outra, o botão ..., ao lado do campo Associated Feature é usado para isso. Por meio desse botão é possível acessar o MC, ou a Configuração do MC – a depender do caso, que está associado ao PjT em questão, permitindo a escolha de uma Característica. A Figura 6(c) apresenta como exemplo o MC, da Figura 1, apresentado nesta parte da ferramenta para associar a Característica compra e venda de ações ao CsT selecionado na Figura 6(a).

Para a elaboração de um PjT da Aplicação, associado a uma Configuração de MC, dois mecanismos podem ser usados: o PjT da Aplicação pode ser elaborado manualmente, de forma independente do PjT do Domínio, ou automaticamente, com base no PjT do Domínio. Para o segundo caso, o PjT da Aplicação é gerado incluindo os mesmos CsT's do PjT do Domínio exceto aqueles que estão associados a Características que não foram selecionadas na Configuração do MC associada ao PjT da Aplicação em questão. Como exemplo da forma automática, um PjT da Aplicação, chamado Home Broker – Default e apresentado na Figura 6(d), foi criado baseado no PjT do Domínio Home Broker da Figura 6(a). O PjT da Aplicação da Figura 6(d) está associado a uma Configuração de MC que não possui algumas das características associadas ao CsT's existentes no PjT do Domínio apresentado na Figura 6(a), como, por exemplo, os CsT's agrupados no CnT PJ (para Pessoa Jurídica), além de outros não visíveis na figura; assim, o PjT da Aplicação não possui tais CsT's, como pode ser observado na Figura 6(d), incluindo o CnT PJ que não está presente.

3.3. Selecionar Conjunto de Teste para Execução

Um CjT é uma seleção de parte dos CsT's de um PjT criada para organizar a execução dos testes de acordo com alguma estratégia de teste definida pelos analistas de teste. Para um PjT, vários CjT's podem ser definidos, a serem executados em diferentes momentos, com diferentes propósitos, de acordo com a estratégia usada.

Figuras 7(a) e 8(a) apresentam exemplos de CjT's selecionados a partir de seus respectivos PjT's, apresentados respectivamente nas Figuras 6(a) e 6(c). O primeiro exemplo se refere a um PjT do Domínio e o segundo a um PjT da Aplicação, embora isso não faça diferença para essa tarefa da SPLOT-TM. Durante a seleção de um CjT, a SPLOT-TM permite que o analista de teste decida por quais CsT's do PjT ele deseja manter ou não em tal conjunto. A seleção ou desseleção de um CsT ou CnT é realizada na SPLOT-TM usando os ícones ✔ e ✖, conforme apresentado nas Figuras 7(a) e 8(a). Os demais ícones apresentados são usados da mesma forma como na SPLOT original, conforme já apresentados na Seção 3.1.

As regras de seleção manual ou automática dos itens em um PjT (CnT's e CsT's) podem implicar ou não em propagações automáticas, cujo mecanismo foi herdado da ferramenta SPLOT original e adaptado. As regras para a seleção do conjunto de teste para execução são as seguintes: (i) a seleção manual de um item provoca a seleção automática de todos os itens acima dele em linha até a raiz da árvore – por exemplo, selecionar um CsT significa que o CnT do qual ele faz parte é também necessariamente selecionado e assim sucessivamente; e, (ii) a desseleção de um item provoca a desseleção de todos os itens abaixo dele – por exemplo, desselecionar um CnT significa que todos os CnT's e CsT's abaixo dele são também necessariamente desselecionados.

Além disso, CsT que não estejam associados a nenhuma Característica do MC ou da Configuração do MC são automaticamente, e permanentemente, desselecionados no CjT para execução, já que os mesmos são considerados ainda incompletos e não devem, portanto, ser disponibilizados para teste; por exemplo, o CsT Segurança da Figura 7(a) está desmarcado e sem opção de resselecionar, pois ele não está associado a nenhuma característica no PjT associado, o que significa que atualmente ele não tem objetivo de testar nada do sistema sendo desenvolvido.

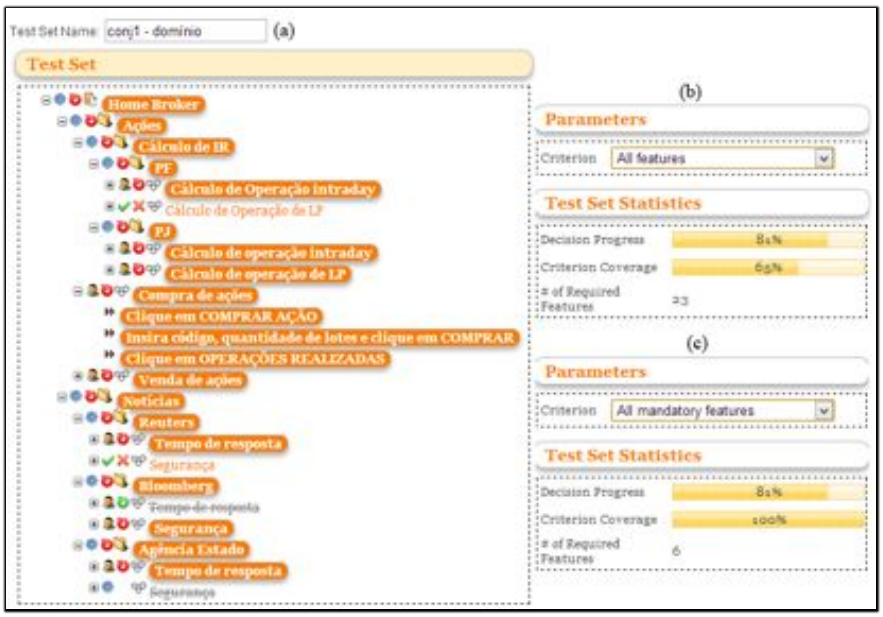

Figura 7. Exemplo de CjT para Domínio: (a) Seleção dos CsT's; (b) Critério "Todas as Características"; (c) Critério "Todas as Características Obrigatórias"

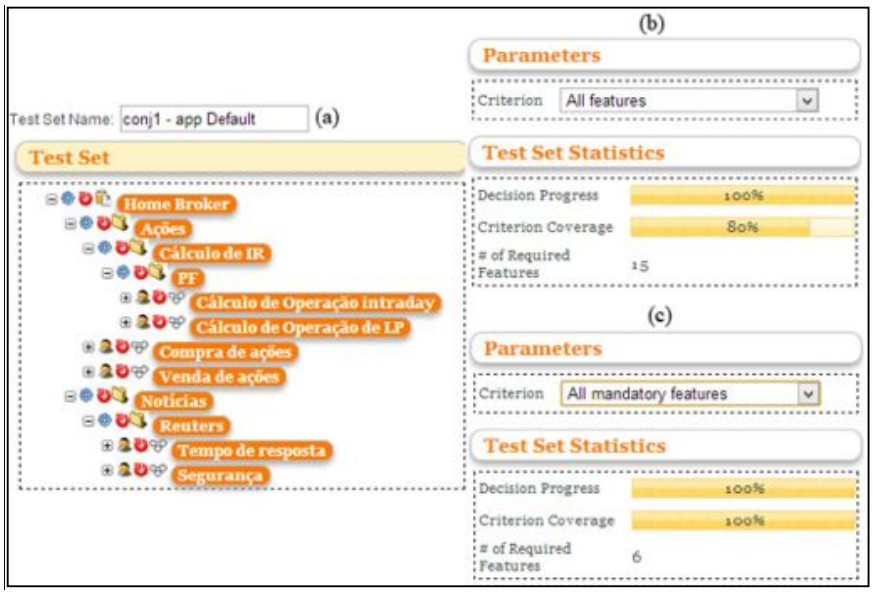

Figura 8. Exemplo de CjT para Aplicação: (a) Seleção dos CsT's; (b) Critério "Todas as Características"; (c) Critério "Todas as Características Obrigatórias"

As Figuras 7(b), 7(c), 8(b) e 8(c) apresentam o painel de Parâmetros e de Estatísticas dos CjT's, usados para auxiliar o analista de teste na seleção. A SPLOT-TM

permite usar diferentes critérios de cobertura na seleção dos CsT's. Essa cobertura é calculada em relação às Características existentes no MC ou na Configuração do MC associado ao PjT em questão. Para isso, a SPLOT-TM implementa os critérios de seleção de teste conforme apresentados na Figura 4. O resultado do cálculo de cobertura das Características pelos CsT's selecionados é apresentado dinamicamente na parte Estatísticas. Como exemplo, nas Figuras 7(b) e 8(b), são apresentados os percentuais de cobertura de Características, considerando o critério Todas as Características, que representam 65% e 80%, do total de todas as Características do MC e da Configuração do MC associados respectivamente ao PjT do Domínio e ao PjT da Aplicação; enquanto que, nas Figuras 7(c) e 8(c), os percentuais de cobertura, considerando o critério Todas as Características Obrigatórias, para as mesmas seleções de CsT's, representam 100% para ambas. Além do critério de cobertura, a SPLOT-TM permite também acompanhar o progresso da seleção do CjT dinamicamente; por exemplo, nas Figura 7(b) e 7(c) é apresentado que 81% dos CsT's disponíveis foram selecionados.

3.4. Registrar Execução de Conjunto de Teste

Para cada CjT selecionado, é possível realizar várias execuções de teste, por exemplo, para diferentes versões da aplicação. A Figura 9(a) apresenta um exemplo de Registro de Execução de Teste disponibilizado pela SPLOT-TM para o CjT da Figura 7(a). Nesse registro, cada item (PsT, CsT ou CnT) pode ter quatro estados, representados pelos seguintes ícones: ● – não testado; ● – aprovado; ● – falhado; e, ● – bloqueado.

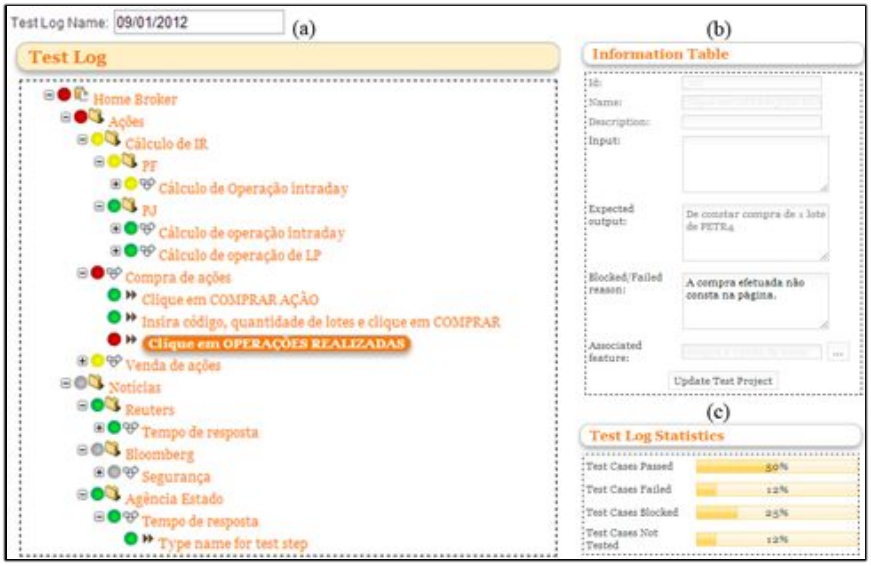

Figura 9. Exemplo de registro de execução: (a) Registro de execução; (b) Tabela de informações; (c) Estatísticas do registro de execução.

Os estados dos PsT's são alterados manualmente via menu de contexto, enquanto que dos demais itens (PjT, CnT's e CsT's) são alterados de forma automática, via propagação, a partir dos PsT's filhos em linha até a raiz da árvore, seguindo as seguintes regra precedência de estado: 1) falhado; 2) bloqueado; 3) não testado; e, 4) aprovado. Ou seja, basta um PsT falhado em um CnT para que o CnT, e todos os itens acima dele até a raiz, seja considerado também falhado, pois ele é o estado de maior precedência em relação aos demais, como na Figura 9(a).

É possível consultar as informações de cada CsT e de seus respectivos PsT's, no painel Tabela de Informações, apresentado na Figura 9(b), que contém um campo para informar o motivo do bloqueio ou da falha (Blocked/Failed reason), preenchido pelo testador quando apropriado. Além disso, por meio desse painel, é possível consultar as Características que cada CsT sendo executado está associado, o que pode ser útil na execução do mesmo. Por fim, o painel Estatísticas, apresentado na Figura 8(c), exibe o percentual de CsT's, e não de PsT's, em cada um dos quadro estados.

4. Trabalhos Relacionados

A ferramenta SPLMT-TE (*Software Product Lines Management Tool – TEst module*) [Neto, Almeida e Meira 2012] é a única até então encontrada semelhante à SPLOT-TM. Assim como a SPLOT-TM, a SPLMT-TE é uma ferramenta web, que busca reduzir o esforço de teste em projetos que usam LP, realizando tarefas de forma sistemática e gerenciável, focada na elaboração de CsT's, além de incentivar o reúso de artefatos; porém, sem ser baseada em MCs como a SPLOT-TM. Informações adicionais da SPLMT-TE são apresentadas na Seção 5, em que uma discussão da ferramenta SPLOT-TM é apresentada. Outros trabalhos que são relacionados à SPLOT-TM, de uma forma não tão próxima quanto a SPLMT-TE, são citados a seguir.

Algumas ferramentas usam a técnica de Teste de Interação Combinatorial para selecionar um pequeno subconjunto de configurações cujas falhas de interação são mais prováveis. Johansen, Haugen e Fleurey (2011) apresentam uma ferramenta que, dado um parâmetro *t*, gera um conjunto de configurações que possui todas as combinações *t* a *t* de Características, enquanto que, no trabalho de Hervieu, Baudry e Gotlieb (2011), são dadas apenas as combinações 2 a 2, também chamado *2-wise*, porém, com o objetivo de gerar um conjunto cujo número de configurações seja o mínimo possível.

Outra ferramenta para LP usa um MC estendido para capturar preferências dos envolvidos com base em objetivos e requisitos de negócio em forma de variáveis linguísticas *fuzzy* [Bagheri, Asadi e Soltani 2010]. Ela permite ordenar e selecionar as Características mais relevantes de um MC, de modo a satisfazer os envolvidos. Ela não é uma ferramenta diretamente relacionada à atividade de teste de software, porém pode ser usada como apoio a essa atividade considerando a ideia de Pohl e Metzger (2011) de que é recomendado testar o quanto antes todas as variantes que provavelmente serão usadas em muitas aplicações. Assim, a ordenação realizada por essa ferramenta pode ser usada na priorização de CsT's, uma vez que Características de maior relevância tendem a ser selecionadas com maior frequência [Ensan *et al.* 2011].

5. Avaliação Empírica e Comparativa da SPLOT-TM

No contexto de teste de software para LP, é mais usual a preocupação com problemas específicos desse contexto, tais como a complexidade do teste devido ao caráter exponencial das possíveis configurações para formação de diferentes produtos [Hervieu, Baudry e Gotlieb 2011; Johansen, Haugen e Fleurey 2011; Kato e Yamaguchi 2011] e a rastreabilidade dos artefatos [Ghanam e Maurer 2010]. Ferramentas de gerenciamento da atividade de teste para LP, como a SPLOT-TM, são menos exploradas, sendo encontrada apenas a SPLMT-TE [Neto, Almeida e Meira 2012] como base de comparação. A SPLMT-TE encontra-se em fase inicial de desenvolvimento, assim seus requisitos funcionais são usados para entendimento de seus objetivos.

Um dos benefícios apresentados pela SPLOT-TM, em relação à SPLMT-TE, é ser baseada em MCs: tanto em termos de associação de CsT's a Características quanto em termos de estruturação do PjT e de seus itens (CnT's, CsT's e PsT's). Considerando que MCs são uma das principais técnicas usadas para gerenciar as variabilidades em LP, é importante que a atividade de teste de software em LP também seja baseada nessa mesma técnica. Dessa forma, toda a equipe de desenvolvimento de sistemas trabalha baseada no mesmo conceito básico – que são as Características. Pela SPLOT-TM isso é feito não apenas na forma de se projetar e selecionar CsT's, o que permite realizar análises de cobertura, mas também na forma de se estruturar visualmente um PjT.

Um benefício da SPLMT-TE, em relação à SPLOT-TM, é o foco no reúso de CsT's a partir de um repositório, os quais podem ser ativados, desativados, removidos ou agrupados em diferentes CjT's. O reúso de CsT's na SPLOT-TM está atualmente limitado a instanciação de um PjT da Aplicação com base em um PjT do Domínio, dirigido pela Configuração do MC, o que por sua vez não é apresentado na SPLMT-TE. Ainda como benefício da SPLMT-TE, em relação à SPLOT-TM, é citada a geração automática de CsT's, embora poucas informações sejam apresentadas sobre isso.

A SPLOT-TM se apresenta como uma ferramenta bastante alinhada aos conceitos da abordagem de LP, possibilitando, por exemplo, realizar testes tanto no nível de Engenharia do Domínio quanto no nível de Engenharia da Aplicação. Considerando que há diferentes possibilidades de trabalhar com estratégias de teste para o contexto de LP, esta ferramenta busca oferecer flexibilidade para os analistas de teste.

6. Conclusão

Embora a SPLOT-TM não esteja completamente desenvolvida, ela se apresenta como uma ferramenta abrangente de apoio ao teste funcional para LP no desenvolvimento de sistemas de informação. Ela é considerada uma ferramenta de teste funcional por ser baseada em Modelos de Características, uma técnica usada em LP para representar as funcionalidades obrigatórias e opcionais ou alternativas do sistema que podem estar presentes em cada produto específico a ser desenvolvido pela LP a ser testada.

Uma possibilidade de extensão, além da implementação da tarefa "Exibir Relatório de Execução de Teste" já projetada, é o oferecimento de reúso dos CsT's entre PjT's do Domínio e não apenas entre PjT's da Aplicação para um mesmo domínio.

Agradecimentos

À Fapesp (Fundo de Amparo à Pesquisa do Estado de São Paulo) pelo apoio financeiro.

Referências

Bagheri, E.; Asadi, M.; Gasevic, D.; Soltani, S. (2010) "Stratified analytic hierarchy process: Prioritization and selection of software features". In: Proc. of the 14th Int. Conf. on Software Product Lines: Going Beyond, South Korea, Springer, p. 300-315.

Cohen, D.; Dalal, S. (1997) The AETG System: An approach to Testing Based on Combinatorial Design, In *IEEE Trans. on Software Engineering* 23(7), p. 437-444.

Czarnecki, K.; Helsen, S.; Eisenecker, U. (2005) Staged Configuration Through Specialization and Multi-Level Configuration of Feature Models. In *Software Process: Improvement and Practice* 10(2), p. 143-169.

Delamaro, E. D.; Maldonado, J. C.; Jino, M., (2007) "Conceitos Básicos", In: Introdução ao Teste de Software, Elsevier, Brasil.

Ensan, A.; Bagheri, E.; Asadi, M.; Gasevic, D.; Biletskiy, Y. (2011) "Goal-Oriented Test Case Selection and Prioritization for Product Line Feature Models". In Proc. of the 8th Int. Conf. on Information Technology: New Generations, USA, p. 291-298.

Ghanam, Y.; Maurer, F. (2010) "Linking Feature Models to Code ArtifaCTs Using Executable Acceptance Tests". In Proc. of the Software Product Lines: Going Beyond, South Korea, p. 211-225.

Hervieu, A.; Baudry, B.; Gotlieb, A. (2011) "PACOGEN: Automatic Generation of Pairwise Test Configurations from Feature Models". In Proc. of the IEEE 22nd Int. Symp. on Software Reliability Engineering, p. 120-129.

Johansen, M.; Haugen, Ø.; Fleurey, F. (2011) "Properties of Realistic Feature Models Make Combinatorial Testing of Product Lines Feasible". In: Proc. of the Model Driven Engineering Languages, New Zeeland, Springer, p. 638-652.

Käkölä, T. and Dueñas, J.C. (2006) Research Issues in Software Product Lines-Engineering and Management. Springer, Germany.

Kato, S.; Yamaguchi, N. (2011) "Variation Management for Software Product Lines With Cumulative Coverage of Feature Interactions". In Proc. of the 15th Int. Software Product Line Conf., Japan, p.140-149.

Kuhn, R.; Walace, D. R.; Gallo Jr., A. M.. (2004) Software Fault Interactions and Implications for Software Testing. In *IEEE Trans. on Soft. Eng.* 30(6), p. 418-421.

Mendonca, M.; Branco, M.; Cowan, D. (2009) SPLOT: Software Product Lines Online Tools**.** In *Systems Engineering* 44(4), p. 761-762, ACM Press.

Narciso, E. N; Nunes, F. L. S.; Delamaro, M. E. (2011) "Seleção de Casos de Teste Utilizando Conceitos de Variabilidade: Uma Revisão Sistemática". Em Anais do VIII Simp. Bras. de Sistemas de Informação, São Paulo – SP, Brasil, p. 115-125.

Neto, C. R. L.; Almeida, E. S. de; Meira, S. R. de. A. L. (2012) "Software Product Lines System Test Case Tool and its Initial Evaluation". In Proc. of the 2012 IEEE 13th Int. Conf. on Information Reuse & Integration (IRI), USA, p. 25-32.

Pohl, K.; Böckle, G.; Linden, F. Van Der. (2005) "Introduction to Software Product Line Engeneering" in Software Product Line Engineering: Foundations, Principles, and Techniques. Springer, Germany.

Pohl, K.; Metzger, A. (2006) Software product line testing. In *Communications of the ACM - Software product line* 49(12), p. 78-81.

Pressman, R. S., (2006) "Estratégias de Teste de Software", In: Engenharia de Software, McGraw-Hill, Brasil.

Silveira N., P. A. M. *et al.* (2011) Testing Software Product Lines. In *IEEE Software* 28(5), p. 16-20.

Sousa, D. M.; Fantinato, M. (2010) "Functional Testing Criteria Based on Feature Modeling for Software Product Line". In: Proc. of the IADIS Int. Conf. Applied Computing, Romania, p. 3-10.