

10 Conclusões e Trabalhos Futuros

A linha central da pesquisa considerada na tese está relacionada ao conceito de agentes de software autoadaptativos capazes de coordenar a execução de testes responsáveis por garantir adaptações mais confiáveis e seguras. Para ajudar na criação de aplicações que tenham agentes desse tipo, o framework JAAF+T (Java self-Adaptive Agent Framework for self-Test) foi proposto.

Também foi formulado um modelo conceitual (ver capítulo 5) composto por informações de teste usadas pelo JAAF+T e que geralmente não são manipuladas por conhecidas abordagens de modelagem de teste. Usando como base esse modelo conceitual, a UML Testing Profile for Coordination (UTP-C) foi proposta para permitir a modelagem de informações úteis de teste em diagramas UML. Visando exemplificar o uso da UTP-C e do JAAF+T, um estudo de caso relacionado ao domínio mercado virtual foi apresentado.

Foram também desenvolvidas ferramentas que permitem gerar artefatos de teste a partir de diagramas UTP-C. Inicialmente foram apresentados componentes desenvolvidos na linguagem Lua, que permitem a geração de arquivos XML utilizados por instâncias do framework JAAF+T. Além desses componentes, também foi desenvolvido um novo plug-in para a ferramenta Rational Software Architecture (RSA), que além de gerar esses arquivos XML também permite a geração de comentários javadoc para scripts de teste desenvolvidos em Java e a geração de relatórios.

O resultado da pesquisa levou à criação de soluções para teste de sistemas multiagentes autoadaptativos que ajudam a tratar problemas em projetos reais da indústria, assim como os utilizados no texto da tese.

10.1. Principais Vantagens das Abordagens Propostas

A partir da tese, as seguintes vantagens foram alcançadas:

- JAAF+T permite maior facilidade em criar aplicações com agentes de software capazes de realizar autotestes (ver capítulo 4). Tal

framework permite que diferentes processos de autoadaptação sejam criados, independente do domínio relacionado. Geralmente essa é uma limitação imposta por diferentes abordagens oferecidas na literatura, assim como mencionado na Seção 9.1.

- O modelo conceitual (ver capítulo 5) considera informações de teste identificadas e utilizadas pelo JAAF+T, podendo ser aplicadas em diferentes linguagens de modelagem.
- UTP-C (ver capítulo 6) contempla informações geralmente negligenciadas por conhecidas linguagens de modelagem de teste.
- A partir do uso do plug-in criado para o RSA (ver Seção 8.3) um conjunto de artefatos de teste pode ser gerado de forma automática a partir de diagramas UTP-C. Essa automatização procura ajudar na coordenação dos testes de algum SUT.

10.2.

Principais Limitações das Abordagens Propostas

A seguir são apresentadas as principais desvantagens das abordagens apresentadas na tese.

- Como o custo de manutenção dos arquivos XML do JAAF+T pode ser alto dependendo da quantidade de informação presente, pretendemos oferecer uma ferramenta gráfica que permita ao desenvolvedor editar esses arquivos de forma mais eficiente e fácil. Assim, o usuário poderá escolher se deseja utilizar essa ferramenta ou o plug-in apresentado na Seção 8.3.
- Investigar soluções que ajudem a criar e manter modelos UTP-C, principalmente quando houver um grande volume de classes e diagramas modelados.
- Por não haver modelagens propostas na literatura que contemplem as informações de teste consideradas no modelo conceitual proposto na tese, não foi possível aplicar avaliações que comparasse alguma abordagem do tipo com a UTP-C. Tal restrição impede que uma análise mais completa possa ser realizada.
- O plug-in desenvolvido para o RSA permite a geração dos arquivos TF.xml e CFF.xml a partir de modelos UTP-C. No entanto, como o JAAF+T também utiliza os arquivos CEF.xml e DF.xml, pretende-

se analisar como a ferramenta pode ajudar a gerar esses arquivos ajudando nas atividades de criação e manutenção.

- A avaliação do plug-in para o RSA, apresentada na Seção 8.3.4, não apresenta em detalhes o impacto que a ferramenta pode ter em algum projeto real. Devido essa limitação, a ferramenta está sendo implantada em um dos projetos do LES para que assim uma melhor análise possa ser realizada.

10.3. Trabalhos Futuros

Alguns trabalhos futuros foram identificados a partir das abordagens oferecidas no documento. Veja a seguir esses trabalhos.

- Integrar o JAAF+T com outras ferramentas de teste, como, por exemplo, ferramentas voltadas para testes funcionais e não funcionais.
- Identificar outras informações relacionadas à área de testes de software que ajudem agentes a coordenar a execução dos testes necessários para validar autoadaptações realizadas.
- Aplicar uma avaliação voltada à manutenção de modelos que envolva mais participantes do que a avaliação aplicada na Seção 6.4.1. Isso porque não foi possível concluir se a UTP-C reduz o esforço em manter modelos de teste comparado a modelos baseados na UTP original com o uso livre de técnicas UML.
- Medir a eficácia dos modelos UTP-C, isto é, mesmo que os modelos estejam corretos segundo verificadores, devemos identificar quais situações eles não funcionam na prática, e por que não o fazem. Além disso, pretendemos identificar o custo de seguir essa abordagem, quando comparado com formas convencionais.
- Estender o plug-in desenvolvido para o RSA para permitir a geração de outros artefatos úteis para equipes de teste, ajudando assim no trabalho dos testadores. Um dos principais pontos que pretendemos explorar é a geração de outros relatórios.
- Aplicar esse plug-in em projetos industriais, para melhor identificarmos as suas principais vantagens e desvantagens.

- Criar um manual de uso detalhado desse plug-in para leitores leigos no RSA. Atualmente há um manual disponível em (EQS, 2012). Tal manual considera que o leitor conhece as principais funcionalidades do RSA.