

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

DISSERTAÇÃO DE MESTRADO

GERAÇÃO DE CASOS DE TESTE DE INTERAÇÃO PARA
APLICAÇÕES DE CELULARES

WILKERSON DE LUCENA ANDRADE

CAMPINA GRANDE – PB

MARÇO DE 2007

Geração de Casos de Teste de Interação para Aplicações de Celulares

Wilkerson de Lucena Andrade

Dissertação submetida à Coordenação do Curso de Pós-Graduação em
Ciência da Computação da Universidade Federal de Campina Grande
como parte dos requisitos necessários para obtenção do grau de Mestre
em Ciência da Computação.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Engenharia de Software

Patrícia Duarte de Lima Machado

(Orientadora)

Campina Grande, Paraíba, Brasil

©Wilkerson de Lucena Andrade, Março de 2007

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

A553g Andrade, Wilkerson de Lucena
2007 Geração de casos de teste de interação para aplicações de celulares / Wilkerson de Lucena Andrade. — Campina Grande, 2007.
109f.: il.

Referências.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.

Orientadora: Patrícia Duarte de Lima Machado.

1— Engenharia de Software 2— Teste de Interação 3— LTS I—Título

CDU 004.41

Resumo

O mercado de telefonia celular tem se mostrado cada vez mais competitivo, demandando um padrão de qualidade cada vez maior. Neste contexto, aplicações são desenvolvidas como conjuntos de funcionalidades, chamados de *features*. Tais *features* são combinadas em cenários de uso das aplicações podendo, inclusive, haver interações entre elas. Interações entre *features* podem ocorrer em dois cenários diferentes: no primeiro caso uma *feature* pode solicitar um serviço ou dados de uma outra e no segundo caso, uma *feature* pode interromper a execução de outra *feature*. Devido ao fato de as *features* serem usualmente desenvolvidas e testadas de forma isolada, os testes de suas interações nos cenários de uso das aplicações ficam comprometidos. Dado que essas interações estejam especificadas, modelos contendo esses comportamentos podem ser obtidos, e assim, usando técnicas de teste baseado em modelo, casos de teste podem ser gerados para verificar tais interações. Neste trabalho, propomos uma estratégia para dar suporte ao teste de interação entre *features* em aplicações para celulares. Para isso, estendemos um formato usado para a especificação de requisitos de interação de *features*, desenvolvemos um procedimento para a geração do modelo comportamental a partir dessa especificação e uma estratégia de geração de casos de teste a partir desse modelo comportamental baseando-se em propósitos de teste, cujo objetivo é extrair cenários de interação entre *features* de forma que tais interações sejam testadas. A ferramenta LTS-BT foi estendida para dar suporte a geração automática de casos de teste de interação e um estudo de caso foi realizado para demonstrar a aplicação da estratégia proposta.

Abstract

The mobile phone market has become even more competitive, demanding high quality standards. In this context, applications are developed as sets of functionalities, called features. Such features are combined in use case scenarios of the application. They can also have interactions between them. Interactions between features can occur in two different scenarios: in the first case each feature can ask for a service or data of another feature and, in the second case, a feature can interrupt the execution of another feature. Due to the fact that the features are usually developed in an isolated way, the tests of their interactions in such scenarios are compromised. Given that the interactions are specified, models containing such behaviors can be obtained, and thus, using model-based techniques, test cases can be generated to verify these interactions. In this work, we propose a strategy to support the interaction test between features in mobile phone applications. For this, we extended a form used to specify the feature interaction requirements, we develop a procedure to generate the behavioral model from such specification and a test case generation strategy from this model based on test purposes, that aims to extract interaction between features such that the interactions are tested. The LTS-BT tool was extended to support the automatic generation of interaction test cases and a case study was conducted to demonstrate the application of the proposed strategy.

Agradecimentos

Primeiramente a Deus por permitir-me concluir mais uma importante etapa da minha vida. Aos meus pais, Sebastião Solano de Andrade e Jucenilda de Lucena Andrade, que sempre me apoiaram de maneira incondicional e sempre estiveram presentes em todos os momentos em que precisei.

Um agradecimento muito especial a minha amada, Janaína, e a minha princesa, Gabriella, pela paciência e pelos bons momentos juntos, incentivando-me muito durante o trabalho. As minhas irmãs, Wilkersya e Wilkerly, pela amizade e companheirismo.

Outro agradecimento especial a minha orientadora, Patrícia Duarte de Lima Machado, pela dedicada orientação, apoio e incentivo. A Laísa, Emanuela e Neto, pelo imenso apoio.

Aos professores Jorge Figueiredo e Guido Lemos, pelos preciosos comentários visando melhorar a qualidade do trabalho. Por fim, a todas as pessoas que, direta ou indiretamente, contribuíram para o sucesso deste trabalho.

Conteúdo

1	Introdução	1
1.1	Objetivos	4
1.2	Resultados	5
1.3	Contexto de Desenvolvimento da Pesquisa	5
1.4	Estrutura da Dissertação	6
2	Fundamentação Teórica	9
2.1	Teste de Software	9
2.1.1	Casos de Teste	10
2.1.2	Tipos de Teste	11
2.1.3	Teste Baseado em Modelo	12
2.1.4	Teste de Interação	14
2.2	Aplicações para Celulares	15
2.3	Interação entre Features	16
2.4	LTS	18
2.4.1	LTS Anotado	19
2.4.2	Geração e Seleção de Casos de Teste com TGV	20
2.4.3	Geração e Seleção de Casos de Teste com LTS-BT	25
2.5	Considerações Finais	27
3	Especificação em Casos de Uso	28
3.1	Especificando Features Simples	28
3.2	Especificando Interações Estáticas	31
3.3	Especificando Interações Dinâmicas	33

3.4	Evolução dos <i>Templates</i>	37
3.5	Considerações Finais	39
4	Transformação de Casos de Uso em Modelos LTS	40
4.1	Modelos LTS de <i>Features</i> Simples	40
4.2	Modelos LTS para Interações Estáticas	41
4.3	Modelos LTS para Interações Dinâmicas	47
4.4	Considerações Finais	54
5	Geração e Seleção de Casos de Teste de Interação	56
5.1	Casos de Teste de Interação	56
5.2	Geração de Casos de Teste de Interação	57
5.3	Seleção de Casos de Teste de Interação	60
5.3.1	Seleção de Casos de Teste de Interação Estática	62
5.3.2	Seleção de Casos de Teste de Interação Dinâmica	64
5.4	Considerações Finais	70
6	Estudo de Caso	72
6.1	Especificação das <i>Features</i> em Casos de Uso	72
6.2	Geração de Casos de Teste de Interação Estática	75
6.3	Geração de Casos de Teste de Interação Dinâmica	79
6.4	Considerações Finais	83
7	Conclusão	84
7.1	Contribuições	85
7.2	Trabalhos Futuros	85
A	Casos de Teste	92
A.1	Casos de Teste de Interação Dinâmica Gerados na Seção 5.2	92
A.2	Casos de Teste Gerados no Estudo de Caso	97

Lista de Figuras

1.1	Visão Geral do Projeto CInBTCRD	6
1.2	Visão Geral do Processo de Teste de <i>Feature</i> e do Teste de Interação	7
2.1	Atividades do Teste Baseado em Modelo	13
2.2	Conjunto de <i>Features</i>	15
2.3	Classificação das interações entre <i>features</i>	17
2.4	LTS que representa um caso de uso da <i>feature</i> Mensagens	19
2.5	Exemplo de um LTS Anotado	21
2.6	Visão Funcional da Ferramenta TGV	22
2.7	Exemplo de IOLTS Utilizado pelo TGV	22
2.8	Exemplo de Propósito de Teste Utilizado pelo TGV	23
2.9	Arquitetura do TGV	24
2.10	Visão Funcional da Ferramenta LTS-BT	26
3.1	Especificação do Cenário Principal	29
3.2	Especificação dos Cenários Alternativos	29
3.3	Exemplo de Caso de Uso - Cenário Principal	30
3.4	Exemplo de Caso de Uso - Cenário Alternativo	31
3.5	<i>Feature</i> Números Embutidos	32
3.6	<i>Feature</i> Contatos - Cenário Principal	32
3.7	<i>Feature</i> Contatos - Cenários Alternativos	33
3.8	Caso de Uso para a Interrupção Chegada de Mensagem	34
3.9	Redefinição de Comportamentos para Interrupções	35
3.10	Especificação de Comportamentos para Interrupções	36
3.11	Exemplo da Primeira Alteração	37

3.12	Exemplo da Terceira Alteração	38
4.1	LTS que Representa um Caso de Uso da <i>Feature</i> Mensagens Prediletas	42
4.2	LTS da <i>Feature</i> Números Embutidos	43
4.3	LTS que Representa a <i>Feature</i> Contatos	44
4.4	LTS da <i>Feature</i> Contatos com seus Vértices Renomeados	45
4.5	LTS Resultante	46
4.6	Especificação da Interrupção Causada pela Chegada de um Aviso	47
4.7	Modelo LTS da Interrupção Causada pela Chegada de um Aviso	48
4.8	Modelo LTS com Interrupções	49
4.9	Seção <i>Interrupções</i> para o Primeiro Cenário	50
4.10	Modelo LTS do Primeiro Cenário	51
4.11	Seção <i>Interrupções</i> para o Segundo Cenário	52
4.12	Modelo LTS do Segundo Cenário	53
4.13	Seção <i>Interrupções</i> para o Terceiro Cenário	54
4.14	Modelo LTS do Terceiro Cenário	55
5.1	Exemplo de um Caso de Teste Simples	57
5.2	Exemplo de Caso de Teste de Interação Estática	57
5.3	Exemplo de Caso de Teste de Interação Dinâmica	58
5.4	Algoritmo de Geração de Casos de Teste	59
5.5	Fluxo da Seleção de Casos de Teste Baseada em um Propósito	61
5.6	Modelo LTS Obtido com o Propósito: “*;O aviso "Número telefônico Armazenado"é mostrado;*;Aceitar”	63
5.7	Caso de Teste 01 - Obtido a partir do Modelo da Figura 5.6	64
5.8	Caso de Teste 02 - Obtido a partir do Modelo da Figura 5.6	64
5.9	Modelo LTS Obtido com o Propósito: “*;A mensagem contém um número telefônico;*;O aviso "Número telefônico Armazenado"é mostrado;*;Aceitar”	65
5.10	Modelo LTS Obtido com o Propósito: “*;A pasta "Mensagens Prediletas"é exibida;Chegada de um aviso;*;Aceitar”	67
5.11	Caso de Teste 01 - Obtido a partir do Modelo da Figura 5.10	68
5.12	Caso de Teste 02 - Obtido a partir do Modelo da Figura 5.10	68

5.13	Modelo LTS Obtido com o Propósito: “*;Ir até uma mensagem;Chegada de um Aviso;*;Aceitar”	69
5.14	Caso de Teste 01 - Obtido a partir do Modelo da Figura 5.13	70
5.15	Caso de Teste 02 - Obtido a partir do Modelo da Figura 5.13	70
6.1	<i>Feature</i> Compositor de Mensagens - Cenário Principal	73
6.2	<i>Feature</i> Compositor de Mensagens - Cenários Alternativos	74
6.3	<i>Feature</i> Contatos	75
6.4	Métricas de Especificação da <i>Feature</i> Compositor de Mensagens	76
6.5	Métricas de Especificação da <i>Feature</i> Contatos	76
6.6	Métricas de Especificação da <i>Feature</i> Chegada de uma Mensagem	76
6.7	Modelo LTS Gerado	77
6.8	Modelo LTS da Interrupção Causada pela Chegada de uma Mensagem	80
6.9	Redefinição do Comportamento da <i>Feature</i> Chegada de uma Mensagem	81
6.10	Modelo LTS do Segundo Comportamento da <i>Feature</i> Chegada de uma Mensagem	81
A.1	Caso de Teste 01 - Obtido a partir do Modelo da Figura 4.10	92
A.2	Caso de Teste 02 - Obtido a partir do Modelo da Figura 4.10	93
A.3	Caso de Teste 03 - Obtido a partir do Modelo da Figura 4.10	93
A.4	Caso de Teste 04 - Obtido a partir do Modelo da Figura 4.10	94
A.5	Caso de Teste 05 - Obtido a partir do Modelo da Figura 4.10	94
A.6	Caso de Teste 06 - Obtido a partir do Modelo da Figura 4.10	95
A.7	Caso de Teste 07 - Obtido a partir do Modelo da Figura 4.10	95
A.8	Caso de Teste 08 - Obtido a partir do Modelo da Figura 4.10	96
A.9	Caso de Teste 09 - Obtido a partir do Modelo da Figura 4.10	96
A.10	Caso de Teste 01 - Obtido a partir do Modelo da Figura 6.7	97
A.11	Caso de Teste 02 - Obtido a partir do Modelo da Figura 6.7	97
A.12	Caso de Teste 03 - Obtido a partir do Modelo da Figura 6.7	98
A.13	Caso de Teste 04 - Obtido a partir do Modelo da Figura 6.7	98
A.14	Caso de Teste 05 - Obtido a partir do Modelo da Figura 6.7	99
A.15	Caso de Teste 06 - Obtido a partir do Modelo da Figura 6.7	99

A.16 Caso de Teste de Interação Dinâmica 01 - Primeiro Cenário	99
A.17 Caso de Teste de Interação Dinâmica 02 - Primeiro Cenário	100
A.18 Caso de Teste de Interação Dinâmica 03 - Primeiro Cenário	100
A.19 Caso de Teste de Interação Dinâmica 04 - Primeiro Cenário	101
A.20 Caso de Teste de Interação Dinâmica 05 - Primeiro Cenário	101
A.21 Caso de Teste de Interação Dinâmica 06 - Primeiro Cenário	102
A.22 Caso de Teste de Interação Dinâmica 01 - Segundo Cenário	102
A.23 Caso de Teste de Interação Dinâmica 02 - Segundo Cenário	102
A.24 Caso de Teste de Interação Dinâmica 03 - Segundo Cenário	103
A.25 Caso de Teste de Interação Dinâmica 04 - Segundo Cenário	103
A.26 Caso de Teste de Interação Dinâmica 05 - Segundo Cenário	104
A.27 Caso de Teste de Interação Dinâmica 01 - Terceiro Cenário	104
A.28 Caso de Teste de Interação Dinâmica 02 - Terceiro Cenário	105
A.29 Caso de Teste de Interação Dinâmica 03 - Terceiro Cenário	105
A.30 Caso de Teste de Interação Dinâmica 04 - Terceiro Cenário	106
A.31 Caso de Teste de Interação Dinâmica 01 - Quarto Cenário	106
A.32 Caso de Teste de Interação Dinâmica 02 - Quarto Cenário	107
A.33 Caso de Teste de Interação Dinâmica 03 - Quarto Cenário	107
A.34 Caso de Teste de Interação Dinâmica 04 - Quarto Cenário	108
A.35 Caso de Teste de Interação Dinâmica 05 - Quarto Cenário	108
A.36 Caso de Teste de Interação Dinâmica 06 - Quarto Cenário	109

Capítulo 1

Introdução

O mercado de telecomunicações tem crescido cada vez mais no decorrer dos anos, principalmente o mercado de celulares. O aumento do número de pessoas usando esse tipo de aparelho de comunicação móvel implica no aumento da complexidade do software embutido nesses aparelhos. Esse aumento da complexidade se dá por dois motivos: a concorrência gerada pelo crescimento do mercado e a necessidade de acompanhar a evolução das infraestruturas de rede e de hardware. Nesse contexto, a necessidade de produtos de qualidade é eminente, visto que, o mercado está cada vez mais competitivo. Por outro lado, a complexidade dos sistemas nos remete a um cenário onde atingir tal qualidade requer ferramentas e técnicas também evoluídas.

Na prática, o teste de software é a técnica mais utilizada na tentativa de se aferir tal qualidade. A abordagem de teste mais adequada a esse contexto é o teste funcional por ser uma abordagem capaz de reduzir os custos inerentes ao processo de teste, uma vez que, a partir da especificação do software, casos de teste podem ser obtidos em paralelo ao seu desenvolvimento. O teste funcional é um tipo de teste baseado na visão de que o software pode ser considerado como uma função que mapeia valores do domínio no seu contradomínio [Jor01], ou seja, é um tipo de teste realizado com o objetivo de observar se, para uma dada entrada, o software produz a saída correta. O teste funcional (também conhecido como caixa-preta) é realizado baseando-se apenas na especificação e interface do software. Em função da redução de custos, muitas pesquisas vêm sendo desenvolvidas com o objetivo de produzir técnicas efetivas para a derivação de casos de teste a partir da especificação dos sistemas. Entretanto, isso só é possível se essa especificação for definida de maneira precisa,

caracterizando com exatidão o comportamento desejado para o sistema [Bei95].

Uma importante característica das aplicações para celulares é o de seu desenvolvimento se dar de maneira evolutiva, onde novos conjuntos de funcionalidades (conhecidas por *features*) são adicionados a versões previamente desenvolvidas. Uma *feature* é um conjunto de requisitos individuais que descreve uma unidade de funcionalidade coesa e identificável [TWFL98]. Como um exemplo de *feature* em aplicações para celulares pode-se citar a *feature* Mensagem que possui, dentre outros requisitos, os requisitos de enviar e receber mensagens. Em geral, as novas *features* são desenvolvidas e testadas de maneira isolada, ou pelo menos com algumas outras *features* das quais ela depende [CKMRM03].

No entanto, esta forma isolada de tratar as *features* pode acarretar alguns problemas, como os causados pelas interações entre as *features*. No contexto deste trabalho, são chamados de interações de *features* os cenários onde as funcionalidades de uma *feature* dependem das funcionalidades de uma outra *feature* (interações estáticas) ou cenários onde há combinações de comportamentos independentes, isto é, cenários onde ocorrem interrupções (interações dinâmicas) [LTX01].

Aplicações para celulares são basicamente aplicações reativas, ou seja, aplicações que interagem com o ambiente no qual estão inseridas, onde a reatividade é caracterizada pela interação da aplicação com o ambiente através das suas entradas e saídas produzidas [dBORZ00]. Nesse contexto, uma das formas mais conhecidas para especificar requisitos de aplicações reativas se dá através de casos de uso. Caso de uso é um conceito amplamente difundido e muito utilizado para a documentação e o desenvolvimento de requisitos [BS02; Coc00; LW03; SW01]. Um caso de uso é uma descrição do comportamento do sistema em termos de seqüências de ações visíveis ao usuário [LW03]. A descrição de casos de uso através de *templates* tem sido muito utilizada na indústria [CS06; dFAM06; PIO05], visto que, a padronização dos documentos traz inúmeras vantagens como a possibilidade de processamento automático, adição de mais detalhes sobre o caso de uso (não suportado em diagramas UML), melhor comunicação entre os membros da equipe, dentre outras. Com relação a especificação de aplicações para celulares, Cabral et al [CS06] apresenta *templates* definidos para especificar casos de uso de *features* e Figueiredo et al [dFAM06] apresenta uma extensão dos mesmos *templates* para a especificação de interações entre *features*.

Testes de *feature* e de interação entre *features* podem ser obtidos através dos *templates*. Para isso, modelos comportamentais que representam o sistema devem ser extraídos dos mesmos. Sistemas de Transições Rotuladas (*Labelled Transition Systems* - LTS) é um formalismo que tem sido muito utilizado na definição de semânticas de especificações comportamentais [JJ05; UKM03; dVT00]. Existem várias ferramentas de animação para esse tipo de modelo no mercado, o que viabiliza a validação dos requisitos de uma aplicação. Em geral, LTS provê uma descrição precisa do comportamento de um sistema, e mais, caminhos em modelos LTS podem ser vistos como seqüências de testes. Logo, modelos LTS são muito úteis na geração de casos de teste.

No contexto das aplicações para celulares, nem sempre o teste de interação permite a execução de todos os possíveis casos de teste gerados. Isso ocorre devido a existência de fortes restrições com relação ao tempo e a recursos necessários a execução. Por isso, a comunidade científica e a indústria têm investido muito em estratégias de seleção de casos de teste [TB02].

A comunidade científica é unânime quando o assunto é a necessidade de automação das técnicas de geração e seleção de casos de teste, de forma que, tanto a geração quanto a seleção tenham a menor intervenção humana possível, evitando assim, a introdução de erros no processo. Uma vez obtida a especificação do comportamento da aplicação, pode-se realizar a geração e seleção automática de casos de teste.

Embora existam várias técnicas de geração e de seleção de casos de teste já disponíveis, a maioria delas utiliza como entrada uma especificação formal, o que pode inviabilizar sua aplicação na prática. Desse modo, partindo da observação de que as empresas utilizam *templates* para especificar aplicações, decidiu-se pela utilização de *templates* de casos de uso neste trabalho, de modo que casos de teste de interação possam ser gerados automaticamente, sem trazer junto com a tarefa de teste, a exigência da produção de artefatos específicos.

Existem várias ferramentas responsáveis pela geração e seleção de casos de teste através de modelos LTS. Dentre as ferramentas relacionadas a este trabalho, pode-se mencionar TGV [JJ05] e LTS-BT [CAN⁺07b]. TGV é uma ferramenta de suporte a testes de conformidade que provê geração e seleção de casos de teste para sistemas reativos e não-determinísticos de forma automática, onde a seleção de casos de teste é realizada através de propósitos de teste. Já LTS-BT, resultado da automação do procedimento apresentado em [Car06], é uma

ferramenta para geração e seleção de casos de teste funcionais voltados para o contexto das aplicações para celulares. LTS-BT dá suporte ao teste de *features* isoladas. O formato da especificação dos propósitos de teste em TGV é muito geral, o que dificulta a exploração de técnicas de seleção específicas, enquanto que, LTS-BT utiliza um formato bem mais simples. Além disso, a utilização de um modelo LTS anotado facilita o processo de adaptação de LTS-BT para o contexto do teste de interação entre *features*.

Teste de interação ainda é uma área muito pouco explorada dentro do contexto deste trabalho. Mesmo se considerarmos um contexto mais amplo de aplicações de tempo real dirigidas a interrupção, não conseguimos encontrar abordagens correlatas. Particularmente, em [LTX01], Redes de Petri Coloridas são usadas como ferramenta de modelagem das interações com o objetivo de fazer avaliações relacionadas a interface do sistema.

Em geral, pode-se observar que existe a necessidade de abordagens voltadas para o teste de interação, baseando-se em notações largamente utilizadas na indústria, como *templates*, e passíveis de automação. O uso de LTS como modelo de geração de casos de teste já é abordado em diferentes trabalhos. No entanto, sua derivação a partir de *templates* de casos de uso e aplicação de técnicas de seleção efetivas são tópicos de interesse ainda não abordados no teste de interação entre *features* no contexto de aplicações para celulares.

1.1 Objetivos

O objetivo deste trabalho é propor uma estratégia para dar suporte ao teste de interação entre *features* em aplicações para celulares. *Features* e suas interações são especificadas usando um *template* proposto para a especificação de casos de uso. Os casos de uso são transformados em um modelo comportamental LTS e a partir desse modelo, casos de teste são gerados automaticamente. O processo final de construção da suíte de testes é apoiado por técnicas de seleção que visam a redução do número de casos de teste com o objetivo de focar o teste em interações específicas.

Para atingir esse objetivo, quatro etapas foram definidas:

1. Refinar o *template* para a especificação de casos de uso;
2. Estender o modelo LTS utilizado pela ferramenta LTS-BT a fim de que o mesmo con-

- temple o comportamento das interações (estáticas e dinâmicas) entre as *features*;
3. Estender o procedimento sistemático de geração e seleção de casos de teste para *features* isoladas, apresentado em [Car06], para dar suporte ao teste de interação;
 4. Automatizar a solução completa.

1.2 Resultados

A principal contribuição deste trabalho é o desenvolvimento de uma estratégia de suporte a geração e seleção de casos de teste de interação entre features de forma automática para aplicações de celulares.

Um ponto relevante deste trabalho é a não inserção de custos extras no processo de teste, uma vez que somente artefatos existentes são utilizados. Além disso, a automação da estratégia pode reduzir custos relacionados ao processo de teste, visto que, erros poderão ser detectados previamente, evitando assim, a propagação dos mesmos para as etapas seguintes do desenvolvimento, o que tornaria mais onerosa a correção.

Por fim, este trabalho possibilitará maior qualidade dos produtos vendidos pelas empresas de telecomunicações, uma vez que, novos cenários (os cenários que causam interações entre features), antes exercitados raramente, serão incluídos no conjunto de casos teste, melhorando a qualidade dos testes realizados antes da entrega do produto ao usuário final.

1.3 Contexto de Desenvolvimento da Pesquisa

Este trabalho faz parte de um projeto de pesquisa, chamado *CIn Brazil Test Center Research and Development (CInBTCRD)*, fruto da cooperação entre a Motorola, o CIn/UFPE (Centro de Informática da Universidade Federal de Pernambuco) e o DSC/UFCG (Departamento de Sistemas e Computação da Universidade Federal de Campina Grande) através do Motorola Brazil Test Center (BTC), em Recife, cujo principal objetivo é definir um processo integrado para a geração, seleção e avaliação de casos de teste para aplicações de celulares. A Figura 1.1 apresenta uma visão geral do projeto CInBTCRD, que aborda a criação de modelos formais não ambíguos a partir de requisitos expressos em casos de uso (usando *templates*)

ou diagramas UML. A partir desses modelos, diversas informações pertencentes ao processo de teste podem ser obtidas de maneira sistemática e automática. Há, também, linhas de pesquisa direcionadas à estimativa do tempo de execução de casos de teste e análise de cobertura de código. Considerando o contexto do CInBTCRD, o escopo deste trabalho é o teste de interação partindo dos *templates* de casos de uso, cujo processo está em destaque na Figura 1.1.

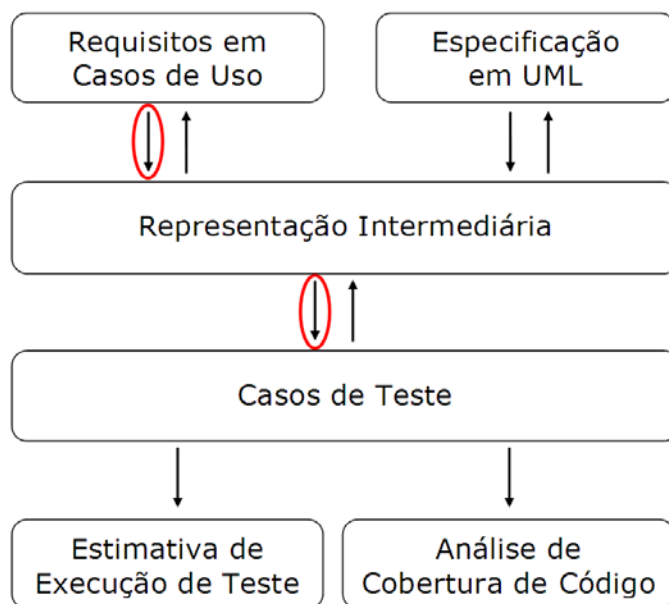


Figura 1.1: Visão Geral do Projeto CInBTCRD

Em geral, o processo de teste no contexto do projeto CInBTCRD se dá com a especificação dos requisitos das *features* e suas interações em casos de uso. Em seguida, o modelo comportamental LTS é gerado a partir dos casos de uso. Por fim, através do modelo LTS gerado, casos de teste de *feature* podem ser obtidos e, combinando o modelo LTS com propósitos de teste, casos de teste de interação podem ser obtidos. A geração de casos de teste de interação utiliza propósitos de teste com o intuito de manter o teste focado em interações específicas, considerando que a verificação de *features* isoladas é coberta pelo teste de *feature*. Uma visão geral desse processo de teste é apresentada na Figura 1.2.

1.4 Estrutura da Dissertação

As demais partes deste documento estão estruturadas da seguinte forma:

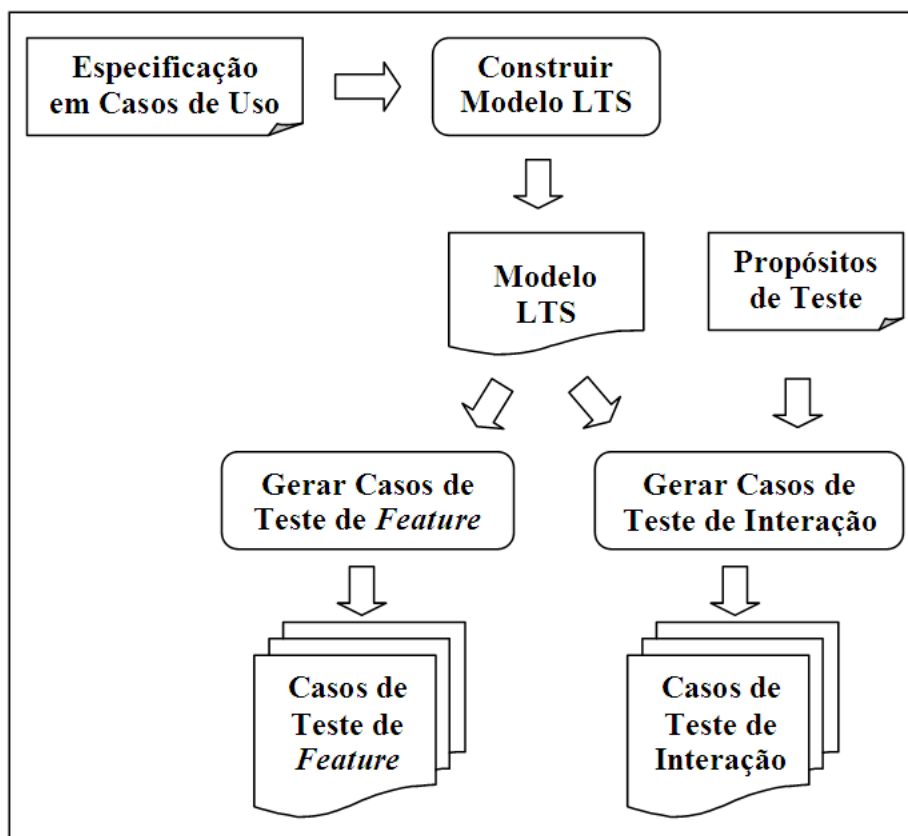


Figura 1.2: Visão Geral do Processo de Teste de *Feature* e do Teste de Interação

Capítulo 2: Fundamentação Teórica Este capítulo fornece o embasamento teórico necessário ao entendimento deste trabalho. São apresentados os principais conceitos relacionados a teste, destacando o teste baseado em modelo e também, conceitos inerentes à aplicações para celulares e LTS.

Capítulo 3: Especificação em Casos de Uso Neste capítulo, os *templates* usados na especificação dos casos de uso de *features* são apresentados, procurando focar no procedimento de como se deve especificar interações estáticas e dinâmicas.

Capítulo 4: Transformação de Casos de Uso em Modelos LTS Este capítulo apresenta o procedimento sistemático para extensão do modelo LTS, usado pela ferramenta LTS-BT, para que tanto as interações estáticas quanto as interações dinâmicas estejam presentes.

Capítulo 5: Geração e Seleção de Casos de Teste de Interação Neste capítulo, é apresentada a extensão do procedimento sistemático apresentado em [Car06], visando a

geração e seleção de casos de teste de interação.

Capítulo 6: Estudo de Caso Este capítulo apresenta uma demonstração prática do trabalho desenvolvido, que vai desde a especificação das interações até a geração e seleção de casos de teste. No final, os resultados obtidos são analisados.

Capítulo 7: Conclusão Neste capítulo final, o trabalho desenvolvido é concluído através da apresentação dos resultados alcançados e as perspectivas para trabalhos futuros são apontadas.

Capítulo 2

Fundamentação Teórica

Este capítulo tem como objetivo principal fornecer o embasamento teórico para os leitores com relação aos conceitos empregados neste trabalho e o estado da arte em trabalhos relacionados. Serão apresentados os principais conceitos relacionados a teste, destacando o teste baseado em modelo e também, conceitos inerentes à aplicações para celulares e LTS.

2.1 Teste de Software

Os sistemas de *software* estão cada vez mais complexos e onipresentes em nossa vida, seja por meio da Internet ou através de sistemas embarcados como, por exemplo, cartões inteligentes, computadores portáteis, celulares, etc. Com o crescimento da utilização desses sistemas, cresce também a exigência por produtos de qualidade. Teste é uma das técnicas de verificação de *software* mais utilizadas na prática [Kat99]. Se usado de forma efetiva, pode fornecer indicadores importantes sobre a qualidade e a confiabilidade de um produto.

O teste de *software* é uma atividade que consiste no esforço de encontrar defeitos introduzidos durante qualquer fase do desenvolvimento ou manutenção de sistemas de *software* e, de forma geral, esses defeitos podem ser decorrentes de omissões, inconsistências ou mau entendimento dos requisitos ou especificações por parte do desenvolvedor [MS01]. No contexto de teste de *software* há três termos muito utilizados na comunidade que são **falha**, **falta** e **erro**. Segundo Binder [Bin99], uma **falha** é a manifestação da inabilidade do software em executar de forma correta uma determinada funcionalidade, **falta** é definida como a ausência de código ou a presença de código incorreto no sistema, e **erro** é a ação humana que provoca

uma falta no sistema.

Teste é importante porque contribui substancialmente no processo que visa verificar se uma aplicação faz tudo o que era esperado que fizesse. Alguns esforços de teste vão além e visam assegurar que as aplicações não fazem nada além do especificado. De qualquer forma, o teste fornece meios de avaliar a existência de defeitos que poderiam resultar em vários prejuízos como perda de tempo, dinheiro, clientes ou até mesmo, no caso de sistemas críticos, de vidas [MS01].

Durante muito tempo, o teste de *software* foi definido dentro dos processos de desenvolvimento como sendo uma atividade a parte e que só era realizada ao final do desenvolvimento dos sistemas. Essa visão tradicional não se mostrou eficiente devido aos altos custos associados a correção de erros encontrados e manutenção do *software*. Isso contribuiu para a definição de métodos e técnicas sistemáticas de teste que constituíssem um processo a parte que pudesse ser aplicado em paralelo ao longo do processo de desenvolvimento [MS01].

Existem vários tipos de teste (e.g [Bei90; Bin99; MS01; Jor01]) que podem ser aplicados dependendo da propriedade do sistema (e.g interface, performance, segurança, etc.) a ser testada, bem como o seu tipo (e.g. sistemas orientados a objetos, distribuídos, reativos, etc.).

2.1.1 Casos de Teste

A essência do teste de *software* é determinar o conjunto de **casos de teste** para o sistema a ser testado. Um **caso de teste** deve conter pelo menos as seguintes informações [Jor01]:

- **Entradas**

- **Pré-condições:** condições que devem ser válidas antes da execução do caso de teste;
- **Dados de entrada:** dados utilizados na execução do caso de teste.

- **Saídas**

- **Pós-condições:** condições que devem ser satisfeitas após a execução do caso de teste;
- **Dados de saída:** o resultado esperado do sistema ao executar o caso de teste.

A utilização de cada caso de teste durante a atividade de teste engloba a definição das pré-condições necessárias, escolha dos dados de entrada, execução do caso de teste, observação dos resultados e, por fim, a comparação dos resultados obtidos com os resultados esperados para determinar se o teste passou ou não. O conjunto de casos de teste definido para um sistema de *software* é denominado de **suíte de teste**.

2.1.2 Tipos de Teste

As duas abordagens fundamentais de teste são conhecidas como teste estrutural e funcional [Jor01].

O teste estrutural, também conhecido como teste caixa-branca ou *white-box testing*, é um tipo de teste onde os casos de teste são identificados baseando-se na implementação do sistema. O objetivo do teste estrutural é testar detalhes procedimentais [Mye79]. Por basear-se na implementação, o teste estrutural consegue testar partes do sistema que não estão na especificação, mas por outro lado, o teste estrutural não consegue identificar comportamentos que estão na especificação, mas não foram implementados.

O teste funcional é um tipo de teste baseado na visão de que o *software* pode ser considerado como uma função que mapeia valores do domínio no seu contradomínio [Jor01], ou seja, é um tipo de teste realizado com o objetivo de observar se, para uma dada entrada, o *software* produz a saída correta. O teste funcional (também conhecido como caixa-preta ou *black-box testing*) é realizado baseando-se apenas na especificação do *software*.

Os casos de teste funcionais, por serem identificados baseando-se apenas na especificação, ao contrário dos casos de teste estruturais, são independentes da maneira como são implementados e por isso, se a implementação do sistema for alterada, os casos de teste continuam sendo úteis. Outra importante vantagem é que as atividades de teste podem ser executadas em paralelo ao desenvolvimento da aplicação, contribuindo para um melhor entendimento e correção dos modelos e especificações desde as etapas iniciais dos processos, evitando detecções de problemas tardiamente, diminuindo assim, o impacto e custos associados às mudanças.

Além da classificação de testes seguindo as abordagens fundamentais, estrutural e funcional, podemos fazer uma nova distinção com relação aos diferentes aspectos do comportamento do sistema a serem testados. Quando o teste é realizado com o objetivo de verificar

se o sistema possui as funcionalidades planejadas e se essas funcionalidades estão de acordo com a especificação, pode-se chamar de teste de conformidade [Tre96]. Quando a especificação é definida através de modelos, chama-se de teste baseado em modelos [EFW01]. Quando o objetivo é testar propriedades específicas do sistema através geração de casos de teste, baseando-se na especificação de propriedades desejáveis para o modelo, chama-se de teste baseado em propósitos [JJ05].

2.1.3 Teste Baseado em Modelo

Na última década, talvez pela popularização da orientação a objetos e utilização de modelos na engenharia de *software*, houve um grande desenvolvimento de uma técnica de teste funcional conhecida por teste baseado em modelo (TBM). TBM é um termo geral usado para denominar uma abordagem baseada no modelo da aplicação sendo testada para realizar atividades de teste [EFW01]. Tais atividades podem ser tanto geração de casos de teste como também avaliação de resultados de teste.

As principais atividades relacionadas ao teste baseado em modelo (ver Figura 2.1) são descritas a seguir:

- **Construir o modelo:** o modelo é construído a partir dos requisitos da aplicação;
- **Gerar casos de teste:** casos de teste são extraídos do modelo com o objetivo de verificar se a aplicação está de acordo com os requisitos;
- **Gerar saídas esperadas:** as saídas esperadas do teste são geradas partindo do modelo. Essas saídas devem indicar o comportamento esperado do sistema;
- **Executar os testes:** a aplicação é exercitada com os casos de teste gerados, produzindo novas saídas;
- **Comparar os resultados:** as saídas da aplicação, obtidas no passo anterior, são comparadas com as saídas esperadas geradas a partir do modelo.

O processo de teste baseado em modelo tem início quando os requisitos da aplicação estão definidos. A partir dos requisitos, o modelo que representa o comportamento esperado da aplicação é construído. Com o modelo pronto, o próximo passo é a geração de casos

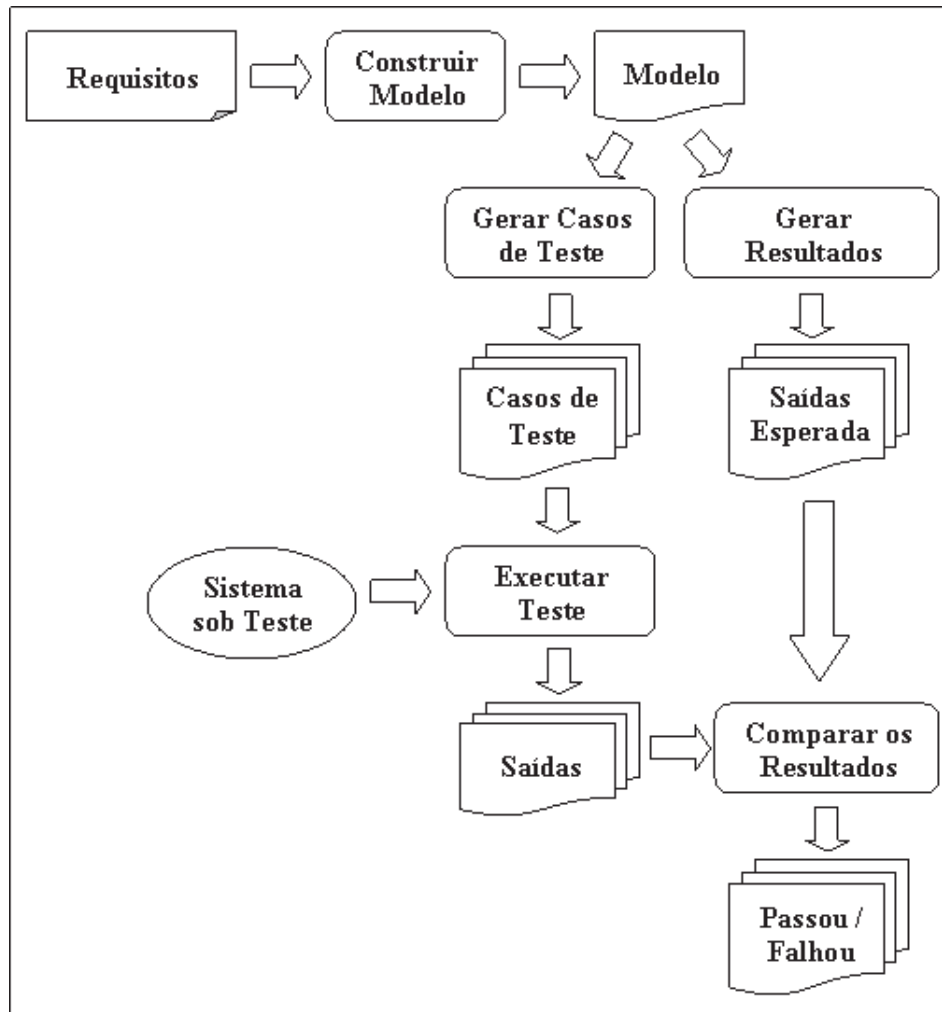


Figura 2.1: Atividades do Teste Baseado em Modelo

de teste. A especificação dos casos de teste inclui, dentre outras informações, entradas e saídas esperadas. Com essas entradas, executamos a aplicação e observamos o seu comportamento. O último passo é comparar as saídas obtidas com as saídas esperadas para avaliar se a aplicação está ou não de acordo com requisitos.

Uma das principais vantagens da utilização do TBM é que o modelo gerado pode servir como um ponto de referência para a comunicação entre todos os envolvidos no processo de desenvolvimento. Outra importante vantagem é que os modelos mais populares possuem um rico embasamento teórico que facilita a geração e automação do processo de teste [EFW01].

Uma das desvantagens da utilização do TBM é a necessidade de conhecimento da notação do modelo e do embasamento teórico a fim de aproveitar o máximo possível o tipo de modelo escolhido. Fazer com que a equipe adquira o conhecimento necessário implica em

investimentos em treinamentos e carência de tempo, além do tempo gasto com a construção do modelo [EFW01]. Outra desvantagem é a forte dependência com relação ao modelo, isto é, como as atividades de teste são realizadas baseando-se no modelo do sistema, a qualidade dos testes realizados está diretamente relacionada a qualidade do modelo.

2.1.4 Teste de Interação

Faltas e falhas devido a interações são vistas como um grande problema por parte dos testadores, visto que, esse tipo de defeito é tão sutil que se torna difícil reconhecê-lo ou até mesmo revelá-lo através de testes [Jor01]. O teste de interação é um tipo de teste que visa verificar a correta colaboração dos componentes de um sistema [MS01].

Faltas relacionadas a interações, geralmente, ocorrem como falhas em sistemas que já estão em uso há algum tempo [Jor01]. Tipicamente, esse tipo de falha tem baixa probabilidade de ocorrência e só aparece após muitas execuções.

Em muitas aplicações, nem sempre é possível executar toda a suíte de teste obtida a partir de uma geração exaustiva de casos de teste [FGMT02], devido a restrições de tempo e recursos. Dessa forma, a aplicação de estratégias de seleção de casos de teste surge como uma forma de amenizar o problema.

Uma estratégia para reduzir o tamanho da suíte de teste é a seleção de casos de teste baseada em propósitos de teste [JJ05]. Os propósitos de teste especificam que uma funcionalidade particular do sistema deve ser testada. Assim, dado o propósito de teste e o modelo da aplicação, é realizado um casamento de padrões entre eles, e com isso, somente casos de teste que atendam ao propósito de teste são gerados.

Uma outra forma de contornar o problema da restrição de tempo e recursos é priorizando os casos de teste. Trabalhos como [CGMC03; BC05] apresentam uma técnica onde casos de teste são gerados seguindo uma ordem de prioridade e visam garantir que as interações entre cada par de componentes sejam testadas pelo menos uma vez. Dessa forma, mesmo que não seja possível executar toda a suíte de teste, devido às restrições citadas, pode-se ao menos executar os casos de teste mais importantes. É importante salientar que o testador é quem escolhe os critérios de priorização dos casos de teste, ou seja, tanto essa técnica quanto aquela, baseada em propósitos, necessitam da experiência do testador durante o processo de geração de casos de teste.

Regehr [Reg05] apresenta uma abordagem para a geração aleatória de casos de teste para aplicações orientadas a interrupção visando testar interrupções inesperadas. Nessa abordagem, informações já obtidas são usadas na geração de novos casos de teste através da utilização de algoritmos genéticos.

2.2 Aplicações para Celulares

Os dispositivos de comunicação pessoal de hoje são muito mais que simples telefones móveis. Atualmente os celulares servem como uma plataforma para uma variedade de aplicações móveis, como por exemplo mensagens de texto, mensagens multimídia, gerência de informações pessoais, jogos, etc [RL03]. Cada aplicação desenvolvida para um celular é denominada *feature*. Uma *feature* é um conjunto de requisitos individuais que descreve uma unidade de funcionalidade coesa e identificável [TWFL98], como um exemplo, Mensagem é uma *feature* que possui, dentre outros requisitos, os requisitos de enviar e receber mensagens.

A Figura 2.2 mostra um ambiente com três *features* especificadas: Mensagem, Contatos e Agenda. Como podemos observar, cada *feature* é composta por seus requisitos. A *feature* Mensagem tem os requisitos de enviar, receber e arquivar mensagens, a *feature* Contatos possui os requisitos de adicionar e remover contatos e a *feature* Agenda tem como requisitos inserir, remover e remarcar compromissos. Como uma *feature* pode ou não interagir com outras *features*, ainda no mesmo exemplo, podemos observar que a *feature* Mensagem pode interagir com a *feature* Contatos.

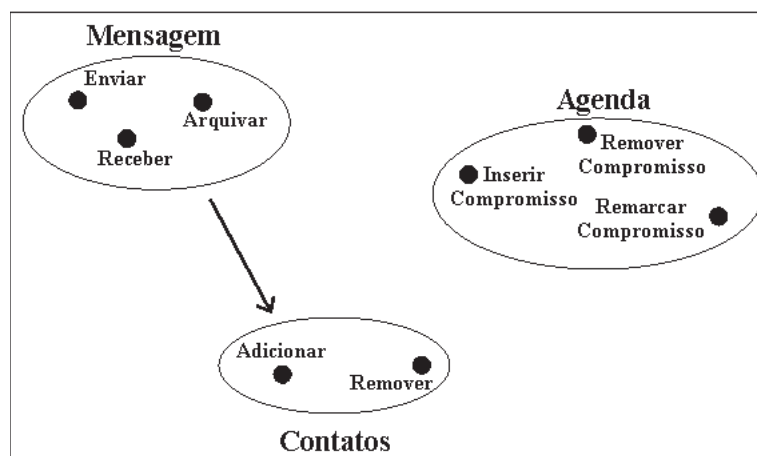


Figura 2.2: Conjunto de *Features*

Uma importante característica das aplicações para celulares é o de seu desenvolvimento se dar de maneira evolutiva, onde novos conjuntos de funcionalidades (*features*) são adicionados a versões previamente desenvolvidas. Cada *feature* é especificada de forma isolada, ou seja, sem o conhecimento de outras *features* com as quais pode ser agrupada [FN03].

Features são tipicamente aplicações reativas [dBORZ00]. Aplicações reativas são sistemas de *software* que interagem continuamente com o seu ambiente [dBORZ00]. Para cada entrada provida pelo ambiente, a aplicação reage emitindo uma saída.

2.3 Interação entre Features

Como foi visto na seção anterior, as *features* são especificadas de maneira isolada. Essa forma de trabalho tem suas vantagens e desvantagens. Como vantagem tem-se a modularização do sistema e a facilidade de implementação, já como desvantagem, têm-se os problemas causados pelas interações entre as *features*.

O termo interação entre *features* (do Inglês *feature interaction*) tem sido usado na comunidade para expressar os problemas causados por funcionalidades adicionadas aos sistemas que geram conflitos com outras já existentes [KK98; dBORZ00; PSC⁺02; CKMRM03]. Esse tipo de problema (do Inglês *feature interaction problem*) já vem sendo pesquisado desde o início da década de 90. Um exemplo que retrata bem o *feature interaction problem* é o cenário onde uma *feature* que redireciona para o aparelho C uma chamada originada de B quando o telefone A está ocupado, pode entrar em conflito com uma outra *feature* que mostra o número do telefone B para o usuário de A quando A está ocupado. Diz-se que nesse caso ocorre interação entre *features* uma vez que isoladamente as *features* funcionam, porém um mesmo sistema não poderá implementar ambas as *features*.

Em nosso contexto, são chamados de interações entre *features* os cenários onde as funcionalidades de uma *feature* dependem das funcionalidades de uma outra *feature* ou cenários onde há combinações de comportamentos independentes [LTX01]. Vamos utilizar a Figura 2.3 para diferenciar o que é estudado quando se fala, na literatura, em *feature interaction problem*, e quando se fala em interações entre *features* no nosso contexto. Para o termo *feature interaction* existe a preocupação em verificar as interações não intencionais procurando observar seus efeitos (Figura 2.3, quadrantes 3 e 4). E quando falamos em interações

entre features, estamos preocupados na verificação das interações intencionais procurando observar o correto funcionamento do sistema (Figura 2.3, quadrantes 1 e 2).

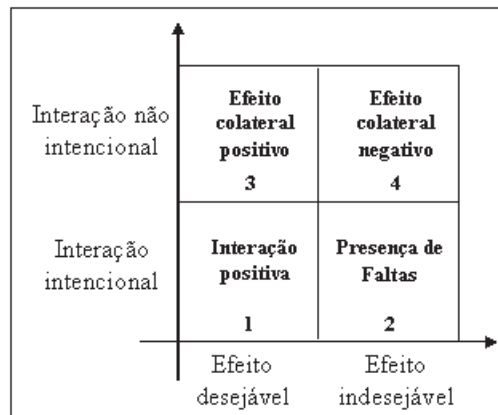


Figura 2.3: Classificação das interações entre *features*

Considerando apenas as interações intencionais e observando a forma como as *features* podem interagir entre si, dois tipos de interações podem ser encontrados: interações estáticas e dinâmicas. Interações estáticas são aquelas onde uma *feature* solicita um serviço ou dados de uma outra *feature*, ou seja, há certa dependência de uso. Neste caso, a primeira *feature* pára sua execução e invoca um serviço (ou dados) de uma outra *feature*, que quando termina sua execução, retorna para a primeira *feature*. Já as interações dinâmicas acontecem quando uma determinada *feature* ou serviço de rede, que está em segundo plano, interrompe a execução de uma *feature* que está em primeiro plano, ou seja, a execução das *features* é alternada. Esse último tipo de interação caracteriza uma combinação de comportamentos independentes.

Como exemplo de uma interação estática há o cenário onde ao terminar de digitar uma mensagem, o usuário acessa a lista de contatos para selecionar um contato e enviar a mensagem. Esse cenário caracteriza uma interação estática entre as *features* Mensagem e Contatos. O seguinte cenário apresenta um exemplo de interação dinâmica: quando o usuário está compondo uma mensagem de texto, uma ligação pode chegar em seu dispositivo, causando uma interação entre as *features* de enviar mensagem de texto e receber ligações. Nesses casos, identificamos o cenário que está executando (chamado de cenário principal) e um outro causado por uma interrupção. No último exemplo, o cenário de compor uma mensagem é o cenário principal e a interrupção é causada pelo cenário de recebimento de uma ligação.

Os exemplos de cenários acima não apresentam conflitos entre requisitos, como citado anteriormente para o termo *feature interaction*, porém, os cenários com interações estáticas, muitas vezes, são especificados de maneira superficial e os cenários com interações dinâmicas geralmente não são especificados. Isso ocorre devido às features serem desenvolvidas e testadas de forma isolada.

Uma característica importante das interações dinâmicas é que elas podem ocorrer em qualquer ponto durante a execução do sistema. Isso significa que o aumento do número de features no celular implica em uma explosão combinatorial do número de cenários com potenciais interações [CKMRM03].

Além da categorização das interações, Lorentsen et al [LTX01] apresenta modelos comportamentais usados na modelagem das mesmas. Redes de Petri Coloridas são usadas como ferramenta de modelagem para avaliações relacionadas a interface do sistema. Como desvantagens dessa abordagem pode-se citar o conhecimento necessário por parte dos testadores em Redes de Petri Coloridas e a inexistência de ferramentas que facilitem o processo de teste.

2.4 LTS

LTS é um formalismo que tem sido muito utilizado na definição de semânticas de especificações comportamentais [JJ05]. Essas especificações ou modelos comportamentais são descrições precisas e abstratas do comportamento esperado do sistema [UKM03].

LTSs são representados por grafos onde os estados representam as possíveis configurações do sistema e as arestas representam o movimento entre essas configurações através da ocorrência de ações.

Um LTS é uma 4-tupla $\langle Q, A, T, q_0 \rangle$, onde [Tre96]:

- Q é um conjunto finito e não-vazio de estados;
- A é um conjunto finito de ações;
- $T \subseteq Q \times A \times Q$ é a relação de transição;
- $q_0 \in Q$ é o estado inicial.

A Figura 2.4 apresenta um LTS que representa o comportamento de um caso de uso da *feature* Mensagens, onde o caso de uso apresentado expressa a possibilidade de remoção de uma mensagem que está na caixa de entrada.

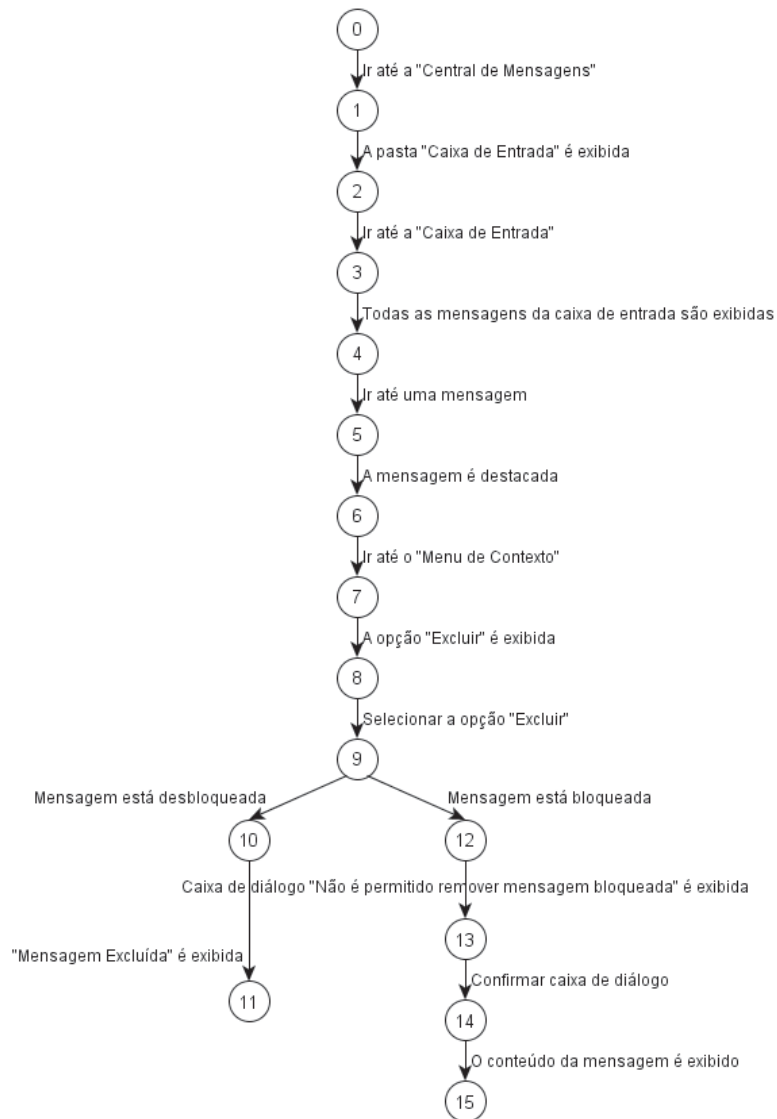


Figura 2.4: LTS que representa um caso de uso da *feature* Mensagens

2.4.1 LTS Anotado

Um LTS Anotado [Car06] é um LTS que segue basicamente a mesma definição apresentada na seção anterior. A diferença entre um LTS e um LTS Anotado está nos rótulos das transições.

Um LTS Anotado é uma 4-tupla $\langle Q, R, T, q_0 \rangle$, onde:

- Q é um conjunto finito e não-vazio de estados;
- $R = A \cup N$ é um conjunto finito de rótulos, onde A é um conjunto finito de ações e N é um conjunto finito de anotações;
- $T \subseteq Q \times R \times Q$ é a relação de transição;
- $q_0 \in Q$ é o estado inicial.

Anotações são inseridas no LTS com uma finalidade específica. No caso deste trabalho, estamos interessados na inserção de anotações com o propósito de facilitar o processo de geração de casos de teste.

A Figura 2.5 apresenta o LTS visto na Figura 2.4 anotado com algumas informações inerentes ao processo teste. Como pode ser observado na Figura 2.5, tais informações são: Passos, Resultados Esperados e Condições Iniciais. Essas anotações inseridas no LTS são utilizadas para separar as principais informações de um caso de teste: os passos ou ações a serem executados, os resultados esperados e as condições iniciais sob as quais as ações podem ser realizadas.

2.4.2 Geração e Seleção de Casos de Teste com TGV

TGV (*Test Generation with Verification technology*) [JJ05] é uma ferramenta de suporte a testes de conformidade que provê geração e seleção de casos de teste para sistemas reativos e não-determinísticos de forma automática. A geração de casos de teste é baseada em técnicas de verificação de modelos, tais como produto síncrono, verificação em tempo de execução e algoritmos de busca.

TGV é baseada na teoria *ioco* apresentada em [Tre96]. Esta teoria define uma relação de conformidade entre a implementação (*Implementation Under Test* - IUT) e sua respectiva especificação, na qual uma IUT é *ioco*-correta em relação a sua especificação se (i) a IUT nunca produz uma saída que não poderia ser produzida pela especificação após a mesma seqüência de entradas e saídas, ou (ii) a IUT pode tornar-se incapaz de evoluir (e.g. em *deadlock*) se a especificação também o pode. Tal incapacidade de evoluir é denominada quiescência (do Inglês *quiescence*), a qual define a ausência de saídas pela IUT.

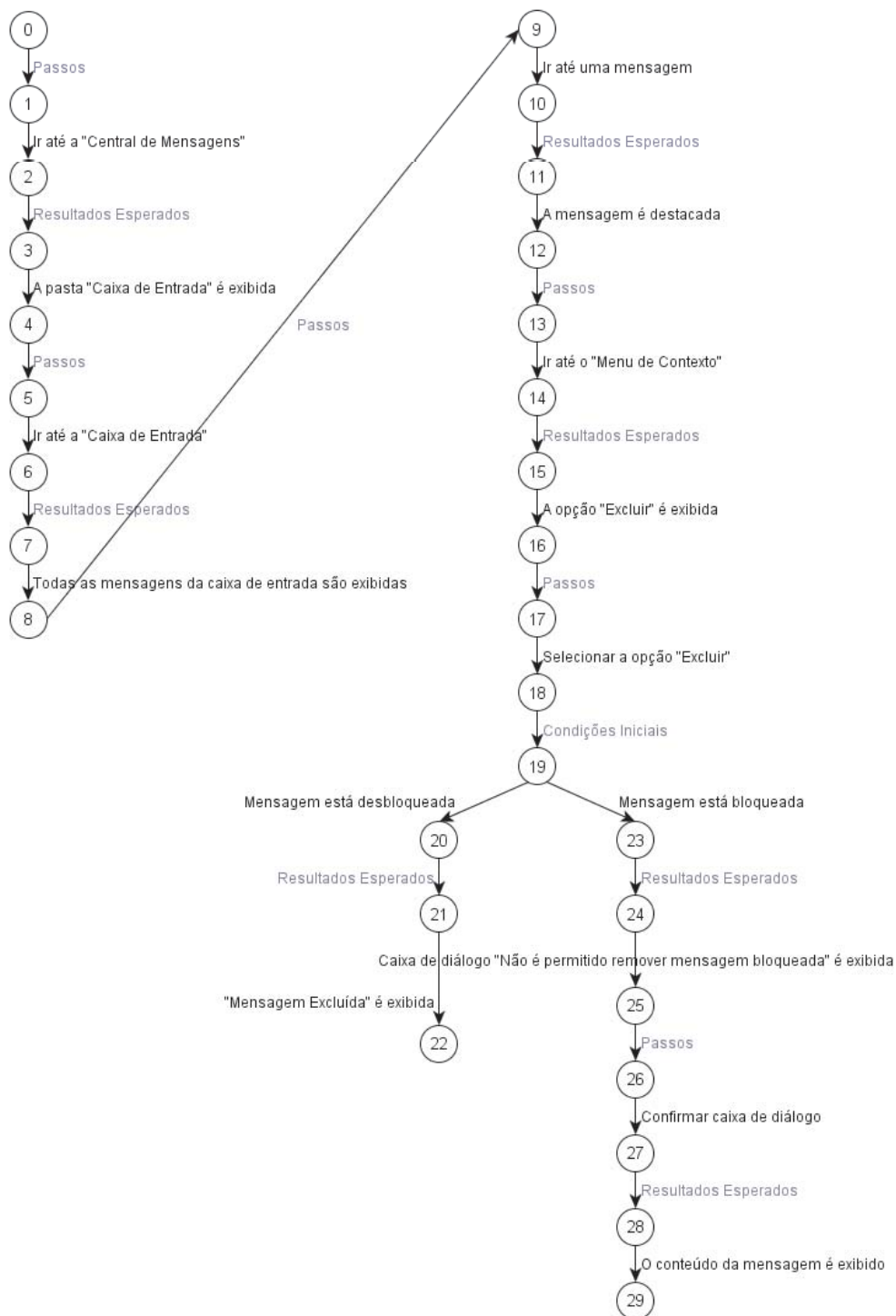


Figura 2.5: Exemplo de um LTS Anotado

A Figura 2.6 apresenta uma visão funcional da ferramenta TGV. Como podemos observar, TGV recebe como entrada a especificação do sistema e um propósito de teste, ambos fornecidos como uma variante do LTS clássico, chamado de sistemas de transições rotuladas

de entrada e saída (*Input-Output Labelled Transition Systems* - IOLTS). Um IOLTS faz distinção entre os eventos do sistema que são controláveis pelo ambiente (as entradas) e aqueles que são observáveis (as saídas). Ações internas do sistemas também podem ser representadas. A Figura 2.7 mostra um exemplo de um IOLTS utilizado pelo TGV. Um evento de entrada é definido através do símbolo “?” seguido do nome do evento (e.g. “?a” para um determinado evento de entrada “a”), já um evento de saída é definido através do símbolo “!” seguido do nome do evento (e.g. “!x” para um determinado evento de saída “x”).

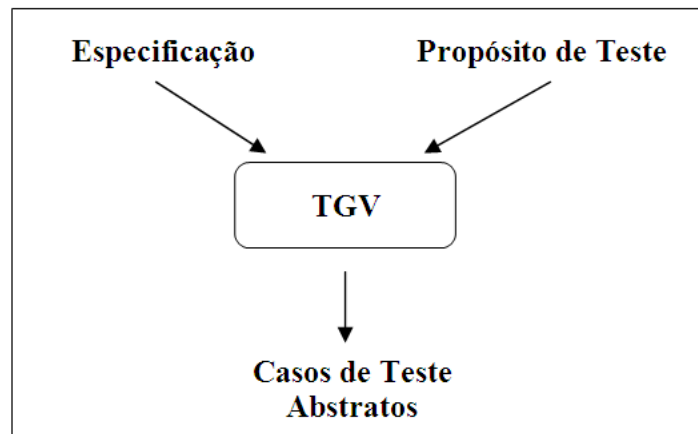


Figura 2.6: Visão Funcional da Ferramenta TGV

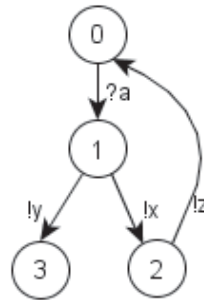


Figura 2.7: Exemplo de IOLTS Utilizado pelo TGV

Formalmente, um IOLTS é uma 4-tupla $\langle Q, A, T, q_0 \rangle$, onde [JJ05]:

- Q é um conjunto finito e não-vazio de estados;
- $A = A_I \cup A_O \cup I$ é um conjunto finito de ações, onde A_I é o conjunto finito de entradas, A_O é um conjunto finito de saídas e I é um conjunto finito de ações internas.

- $T \subseteq Q \times A \times Q$ é a relação de transição;
- $q_0 \in Q$ é o estado inicial.

Os propósitos de teste são equipados com estados especiais de aceitação e rejeição. Esses estados devem ser utilizados para guiar processos de seleção de casos de teste. Aceitação representa seqüências que devem ser selecionadas para os casos de teste, enquanto que rejeição representa as que não devem originar casos de teste. A Figura 2.8 mostra um exemplo de propósito de teste utilizado por TGV. Este objetivo de teste define que a seqüência de transições aceitas para a execução dos casos de teste deve ser composta do evento “a” seguido do evento “y”. Execuções em que o evento “x” ocorra não devem gerar casos de teste.

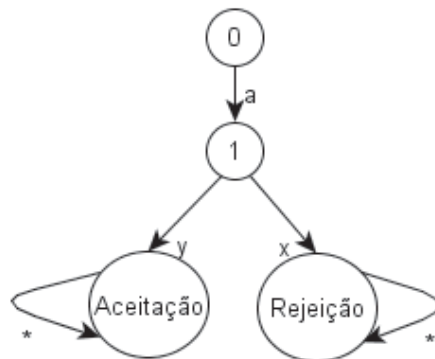


Figura 2.8: Exemplo de Propósito de Teste Utilizado pelo TGV

A Figura 2.9 mostra a arquitetura do TGV com as etapas no processo de geração de casos de teste. O primeiro passo do processo de geração é a realização do produto síncrono entre a especificação S do sistema e o propósito de teste TP , resultando em um terceiro IOLTS, chamado de SP . O produto síncrono é a etapa responsável pela identificação dos comportamentos aceitos ou rejeitados pelo propósito de teste.

O próximo passo consiste em gerar um novo IOLTS a partir de SP que contenha somente os comportamentos observáveis da especificação, ou seja, sem ações internas. Para tal, TGV provê meios de definir de maneira explícita e observável os estados quiescentes em um processo chamado de determinação.

Em seguida, a ferramenta constrói um IOLTS chamado CTG (Complete Test Graph) que representa um grafo que contém todos os casos de teste referentes ao propósito de teste fornecido inicialmente. Além disso, o CTG possui as entradas e saídas invertidas com relação

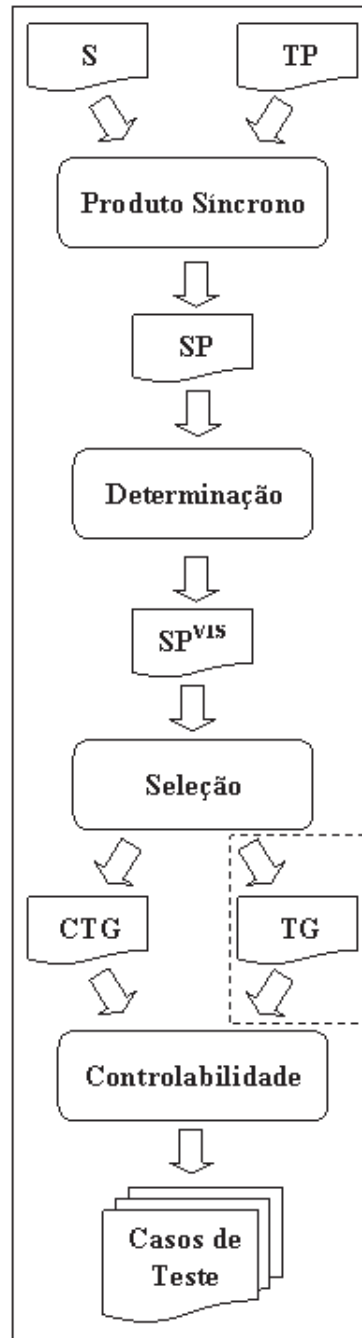


Figura 2.9: Arquitetura do TGV

a especificação do sistema. Essa inversão é realizada com intuito de associar a implementação aos casos de teste. Dessa forma, as saídas do CTG serão entradas para a implementação e as entradas serão saídas da implementação.

Concluindo o processo, os conflitos de controlabilidade são solucionados e os casos de teste são gerados. Conflitos de controlabilidade ocorrem quando, a partir de um determinado

estado, existem várias transições de saída ou várias transições de entrada e saída. Tais conflitos podem ser resolvidos através da escolha de uma das transições de saída ou todas as saídas são eliminadas e as entradas são mantidas. Essa escolha é feita de forma não determinística, ou seja, a cada seleção de casos de teste, um caso de teste diferente pode ser selecionado. Opcionalmente, um grafo de teste chamado *TG (Test Graph)* pode ser gerado durante a fase de seleção através da solução *on-the-fly* dos problemas de controlabilidade.

2.4.3 Geração e Seleção de Casos de Teste com LTS-BT

LTS-BT (*Labelled Transition System-Based Testing*) [CAN⁺07b] é uma ferramenta para geração e seleção de casos de teste funcionais voltados para o contexto das aplicações para celulares. LTS-BT foi desenvolvida para dar suporte ao teste de *feature* [Car06], onde os casos de teste são gerados e selecionados a partir de um LTS Anotado (Subseção 2.4.1).

A Figura 2.10 apresenta uma visão funcional da ferramenta LTS-BT. Como podemos observar, LTS-BT recebe como entrada uma especificação do sistema, que pode ser tanto um LTS Anotado quanto um conjunto de diagramas de seqüência; um propósito de teste, representado através de uma seqüência de transições; e uma porcentagem de cobertura de caminhos, usada na seleção do número de casos de teste de acordo com um critério de similaridade.

Quando a entrada se dá por meio de diagramas de seqüência, a ferramenta LTS-BT gera o modelo comportamental em LTS baseando-se num procedimento apresentado em [Bin99]. Com relação a geração de casos de teste, como LTS-BT é voltada para o teste funcional, os casos de teste gerados são compostos das seguintes anotações: passos ou ações do usuário, resultados esperados ou respostas do sistema e condições iniciais. Como cada caminho no LTS (partindo de um nó inicial até um nó final) é um caso de teste, se faz necessário a identificação de todos os caminhos do modelo para assegurar que todas as funcionalidades possam ser testadas. Esse processo de identificação de todos os caminhos é realizado através de um algoritmo de busca em profundidade.

Como a geração exaustiva de casos de teste pode não ser adequada ao contexto das aplicações para celulares, visto que, muitos casos de teste são executados manualmente, LTS-BT implementa duas estratégias de redução da suíte de testes: uma baseada na seleção de casos de teste usando propósitos de teste e outra baseada em similaridade.

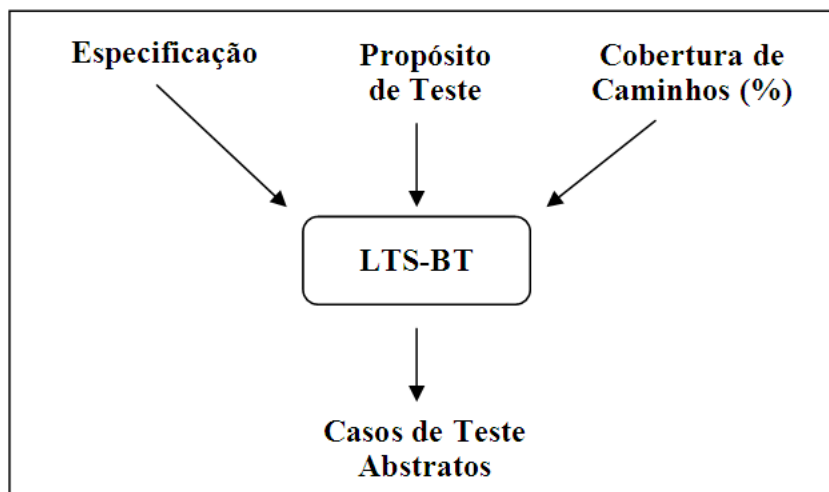


Figura 2.10: Visão Funcional da Ferramenta LTS-BT

LTS-BT usa uma notação bem simples para a definição de um propósito de teste, onde o mesmo é definido através de uma seqüência de transições. Nessa seqüência, um asterisco “*” pode aparecer para indicar que naquele ponto pode ocorrer qualquer transição. Um propósito de teste sempre termina com uma transição chamada “Aceitar” (indicando que todos os casos de teste precisam estar em conformidade com o propósito) ou com uma transição chamada “Rejeitar” (caso contrário). Como exemplo de um propósito usado por LTS-BT, vamos usar o LTS apresentado na Figura 2.5 com o objetivo de definir um propósito de teste para o cenário com um aviso informando que a mensagem não pode ser removida porque a mesma está bloqueada. Para esse cenário, o seguinte propósito poderia ser definido: “*;Caixa de diálogo "Não é permitido remover mensagem bloqueada"é exibida;*;Aceitar”.

Para a seleção de casos de teste baseada em similaridade, LTS-BT calcula o grau de similaridade entre dois casos de teste, analisando o tamanho do caminho e daí, aquele caso de teste que tem maior grau é eliminado.

A ferramenta LTS-BT se mostra bem mais atrativa que a ferramenta TGV visto que, o CTG gerado por TGV pode conter infinitos casos de teste e o formato da especificação dos propósitos de teste é muito geral, dificultando a exploração de técnicas de seleção específicas necessárias a alguns contextos como o de aplicações para celulares. Já o formato dos propósitos de teste utilizados por LTS-BT, que são bem mais simples, e a utilização de um modelo anotado facilitam o processo de adaptação da ferramenta para o contexto do teste de interação entre *features*.

2.5 Considerações Finais

Este capítulo apresentou o embasamento teórico necessário ao entendimento deste trabalho. Foram apresentados os principais conceitos relacionados a teste de *software*, os tipos de teste, uma abordagem de teste funcional baseada em modelo onde foram discutidas vantagens e desvantagens, e em seguida, foi apresentado um tipo específico de teste chamado de teste de interação.

Com relação a aplicações para celulares, foram apresentadas algumas características específicas e também foi introduzido o conceito de *feature*, muito utilizado na área de telecomunicações. Em seguida, houve uma discussão acerca da questão de interações entre *features*, onde o principal ponto apresentado foi a diferença existente entre o que a comunidade tem chamado de *feature interaction* e o conceito de interação entre *features* adotado neste trabalho.

Por fim, foram apresentados os conceitos de LTS e LTS Anotado, procurando explicitar a utilidade de cada um, e também, uma visão geral de duas ferramentas (TGV e LTS-BT) relacionadas a este trabalho, mencionando as vantagens e desvantagens de cada uma em relação ao contexto do teste de aplicações para celulares.

Capítulo 3

Especificação em Casos de Uso

Este capítulo tem como objetivo apresentar os *templates* usados na especificação dos requisitos de *features* [CS06; dFAM06]. Primeiramente, será apresentado o formato da especificação de *features* simples, isto é, *features* que não interagem com outras *features*. Em seguida, o procedimento de como se deve especificar interações estáticas e dinâmicas é descrito. Por fim, são apresentadas as contribuições relacionadas às alterações realizadas nos *templates* com o intuito de melhorar o processo de especificação de interações entre *features* juntamente com suas respectivas justificativas.

3.1 Especificando Features Simples

A especificação da cada *feature* se dá através de um único *template* [CS06]. Considerando todo o conjunto de *features* existentes, cada *feature* deve possuir uma identificação única. Cada *template* pode conter vários casos de uso, todos relacionados a mesma *feature*. Considerando o escopo do *template*, cada caso de uso deve possuir uma identificação única. Cada caso de uso é descrito através de um conjunto de passos e considerando o escopo do mesmo, cada passo também deve possuir uma identificação única.

Cada caso de uso de uma *feature* simples deve ser especificado através de duas seções: “Cenário Principal” (Figura 3.1) e “Cenários Alternativos” (Figura 3.2). A seção “Cenário Principal” descreve o fluxo principal do caso de uso, enquanto que a seção “Cenários Alternativos” descreve os possíveis fluxos alternativos. Caso haja mais de um cenário alternativo, cada cenário será especificado através de uma tabela diferente.

Cenário Principal

Descrição:

De:

Para:

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1M			
2M			
...			

Figura 3.1: Especificação do Cenário Principal

Cenários Alternativos

Descrição:

De:

Para:

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1A			
2A			
...			

Figura 3.2: Especificação dos Cenários Alternativos

Tanto a seção “Cenário Principal” quanto a seção “Cenários Alternativos” possuem os mesmos campos a serem preenchidos. O primeiro a ser preenchido é o campo “Descrição”, que contém uma explicação do fluxo que está sendo especificado. Em seguida, vêm os campos “De” e “Para” que contêm uma lista de identificadores de passos. O campo “De” indica que o fluxo descrito é uma possível continuação do passo especificado. O campo “Para” indica que o fluxo pode ser continuado a partir dos passos especificados.

Quando um fluxo não começa a partir de nenhum outro fluxo, o valor do campo “De” deve ser preenchido com o valor “INICIO”. Quando um fluxo não é continuado por nenhum outro fluxo, o valor do campo “Para” deve ser “FINAL”.

Como no escopo de um caso de uso cada passo tem uma identificação diferente, quando os campos “De” e “Para” referenciam um passo que está dentro do próprio caso de uso, basta indicar o identificador do passo. Quando houver uma referência a um passo que está fora do caso de uso em questão, então a referência deve conter a identificação do caso de uso e do passo referenciado, separados por “#”, como por exemplo “ID_CasoUso#ID_Passo”.

A última parte da especificação de um fluxo é uma tabela contendo os passos. Como pode-se observar, nas Figuras 3.1 e 3.2, essa tabela é igual tanto para a seção do cenário principal quanto para os cenários alternativos. Cada linha da tabela representa um passo do fluxo sendo especificado, e como já foi explicado, cada passo contém uma identificação.

Cada linha da tabela possui uma ação do usuário e a respectiva resposta do sistema. Além disso, cada passo possui uma condição (coluna “Estado do Sistema”) que determina se a resposta do sistema será realizada ou não.

O exemplo de utilização do *template* para a especificação de *features* simples será apresentado através de uma *feature* real, chamada Mensagens Prediletas. Essa *feature* especifica a adição de uma nova pasta no celular para que o usuário possa armazenar as suas mensagens prediletas. Como requisitos, o usuário poderá mover mensagens para a pasta de mensagens prediletas através de vários atalhos e também, realizar operações referentes a esta pasta, como por exemplo, apagar todas as mensagens contidas na mesma.

O caso de uso da *feature* Mensagens Prediletas escolhido para demonstrar o formato da especificação expressa a possibilidade da transferência de uma mensagem que está na caixa de entrada para a pasta de mensagens prediletas. O cenário principal e o cenário alternativo desse caso de uso estão especificados através das Figuras 3.3 e 3.4, respectivamente. Como podemos observar, os cenários são descritos através de passos que compreendem uma ação do usuário e a respectiva resposta do sistema. Por exemplo, o passo “5M” possui como ação do usuário a seleção da opção “Mover para Mensagens Prediletas” e a respectiva resposta do sistema é exibir uma mensagem avisando que a mensagem foi movida.

Cenário Principal

Descrição: A mensagem é movida para a pasta de mensagens prediletas

De: INICIO

Para: FINAL

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1M	Ir até a Central de Mensagens		A pasta "Mensagens Prediletas" é exibida
2M	Ir até a "Caixa de Entrada"		Todas as mensagens da "Caixa de Entrada" são exibidas
3M	Ir até uma mensagem		A mensagem é destacada
4M	Ir até o "Menu de Contexto"		A opção "Mover para Mensagens Prediletas" é exibida
5M	Selecionar a opção "Mover para Mensagens Prediletas"	A memória não está cheia	"Mensagem movida para a pasta Mensagens Prediletas" é exibida

Figura 3.3: Exemplo de Caso de Uso - Cenário Principal

Caso a condição especificada na coluna “Estado do Sistema” não seja satisfeita, um cenário alternativo deverá ser especificado, como temos o exemplo do passo “5M” do cenário principal, que possui um cenário alternativo (passos “1A” e “2A” da Figura 3.4).

Cenário Alternativo

Descrição: A mensagem não é movida para a pasta Mensagens Prediletas porque a memória está cheia

De: 5M

Para: FINAL

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1A	Selecionar a opção "Mover para Mensagens Prediletas"	A memória está cheia	Caixa de diálogo "Memória Insuficiente" é exibida
2A	Confirmar a caixa de diálogo de informação da memória		O conteúdo da mensagem é exibido

Figura 3.4: Exemplo de Caso de Uso - Cenário Alternativo

3.2 Especificando Interações Estáticas

Em alguns casos, precisamos especificar um comportamento através de vários casos de uso, como por exemplo, cenários onde ocorrem interações estáticas, isto é, cenários onde uma *feature* usa serviços ou dados de uma outra *feature*. Nesses casos, um comportamento é descrito por mais de uma *feature*, fazendo com que exista um *template* para cada uma. Para esse caso, a solução adotada foi permitir a referência entre *templates*, e assim, um determinado comportamento pode ser especificado através de vários *templates*.

Quando for necessário referenciar, no campo “De” ou no campo “Para”, um passo que está em outro *template*, primeiro se especifica o identificador da *feature*, depois o identificador do caso de uso e por último, o identificador do passo, tudo separado por “#”, como por exemplo: “ID_FEATURE#ID_CasoUso#ID_Passo”.

Para demonstrar como as interações estáticas podem ser especificadas, vamos usar duas novas *features*: Números Embutidos e Contatos. O comportamento descrito pela interação estática entre essas *features* permite que o usuário selecione um número telefônico que está embutido em uma mensagem de texto recebida e o armazene na sua lista de contatos.

A Figura 3.5 apresenta a especificação da primeira parte do comportamento desejado, ou seja, apresenta somente a especificação da *feature* Números Embutidos. Observe que o campo “Para” (Figura 3.5) é usado para referenciar o outro caso de uso, da *feature* Contatos, e dessa forma, a descrição do comportamento pode continuar em outro *template*. Por exemplo, o texto “CONTATOS#UC_01#1M”, em destaque no cenário principal e no cenário alternativo da Figura 3.5, é usado para referenciar o passo “1M” do caso de uso “UC_01” da *feature* Contatos, ou seja, após alcançar os passos “4M” ou “3A” da *feature* Números Em-

butidos, a descrição do comportamento continua a partir do passo “1M” da feature Contatos (Figuras 3.6 e 3.7).

Cenário Principal

Descrição: A mensagem contém um número telefônico.

De: INICIO

Para: CONTATOS#UC_01#1M

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1M	Abrir a "Caixa de Entrada"		A lista de mensagens é mostrada
2M	Selecionar uma mensagem		A mensagem é destacada
3M	Selecionar a opção "Ler"		A mensagem de texto é mostrada. A lista de opções é mostrada
4M	Selecionar a opção "Armazenar"	A mensagem contém um número telefônico	

Cenário Alternativo

Descrição: A mensagem contém mais de um número telefônico.

De: 4M

Para: CONTATOS#UC_01#1M

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1A	Selecionar a opção "Armazenar"	A mensagem contém mais de um número telefônico	A lista dos números telefônicos é mostrada
2A	Escolher um número		O número telefônico é destacado
3A	Selecionar a opção "Armazenar"		

Figura 3.5: Feature Números Embutidos

Cenário Principal

Descrição: Todos os campos obrigatórios são preenchidos.

De: NUMEROS_EMBUTIDOS#UC_01#4M, NUMEROS_EMBUTIDOS#UC_01#3A

Para: FINAL

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1M		O armazenamento padrão está configurado como sendo na memória de celular	A tela "Criar Contato" é mostrada
2M	Preencher todos os campos obrigatórios na tela "Criar Contato"		Todos os campos obrigatórios são preenchidos
3M	Selecionar a opção "Pronto"		O aviso "Número telefônico Armazenado" é mostrado

Figura 3.6: Feature Contatos - Cenário Principal

Observe que tanto podemos usar o campo “Para” quanto o campo “De” para unir *templates*. Por exemplo, na Figura 3.6 o campo “De”, em destaque, especifica que o comportamento está sendo continuado a partir do passo “4M” ou “3A” do caso de uso “UC_01” da

Cenários Alternativos

Descrição: Alguns campos obrigatórios não são preenchidos.

De: 2M

Para: 2M

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1A	Preencher alguns campos obrigatórios na tela "Criar Contato"		Alguns campos obrigatórios são preenchidos
2A	Selecionar a opção "Pronto"		A tela "Algum campo obrigatório não foi preenchido" é mostrada
3A	Selecionar a opção "Ok"		A tela "Criar Contato" é mostrada

Descrição: O usuário cancela a operação de criar um novo contato.

De: 2M

Para: FINAL

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
4A	Selecionar a opção "Cancelar"		A mensagem de texto é mostrada

Figura 3.7: *Feature* Contatos - Cenários Alternativos

feature Números Embutidos.

3.3 Especificando Interações Dinâmicas

Esta seção apresenta uma abordagem para a especificação de cenários que causam interações dinâmicas entre *features* (chamadas de interrupções) bem como para a especificação de cenários que não permitem ou redefinem os comportamentos dessas interrupções [dFAM06].

As interações entre *features* ocorrem quando uma *feature* está executando um determinado cenário e um cenário de uma outra *feature* o interrompe e, neste caso, inicia sua execução. Por exemplo, quando um usuário está compondo uma mensagem de texto uma outra mensagem pode chegar, caracterizando a interação entre as *features* de enviar mensagem e de chegada de mensagem.

Para que esse cenário esteja presente no modelo comportamental e, conseqüentemente, possam existir casos de teste que o exercite, é preciso que tanto o cenário que interrompe seja especificado em casos de uso como o cenário que é interrompido permita que tal interrupção ocorra. Dessa forma, sempre que uma *feature* possuir uma interrupção, ou seja, um cenário que é executado mesmo quando outro cenário esteja executando, um caso de uso deve ser especificado para tal interrupção. Por exemplo, a *feature* Mensagens possui a interrupção Chegada de uma Mensagem, então um caso de uso deve ser especificado para a interrupção

de chegada de uma mensagem.

Os casos de uso de interrupções são especificados através das mesmas seções do *template*, mostradas nas Figuras 3.1 e 3.2. Na especificação de interrupções, só existe uma regra que a diferencia da especificação de *features* simples: o campo “Estado do Sistema” do primeiro passo do cenário principal deve ser preenchido com o nome da *feature* responsável pela interrupção. Na Figura 3.8, temos um exemplo de um caso de uso para a interrupção de chegada de uma mensagem.

Cenário Principal

Descrição: O usuário decide ler a mensagem.

De: INICIO

Para: FINAL

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1M	Enviar uma mensagem de um outro celular para o celular sob teste	Chegada de uma Mensagem	Aviso "Nova Mensagem" é mostrado
2M	Selecionar a opção "Ler"		A mensagem é mostrada
3M	Selecionar a opção "Voltar"		Volta para a aplicação que estava executando antes da interrupção do aviso de chegada de uma mensagem

Cenário Alternativo

Descrição: O usuário decide não ler a mensagem.

De: 2M

Para: FINAL

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1A	Selecionar a opção "Cancelar"		Volta para a aplicação que estava executando antes da interrupção do aviso de chegada de uma mensagem

Figura 3.8: Caso de Uso para a Interrupção Chegada de Mensagem

O caso de uso da Figura 3.8 é especificado quando a *feature* relativa à interrupção for desenvolvida, e servirá como comportamento padrão para todas as outras *features* para quando essa interrupção ocorrer. Como já foi explicado, o primeiro passo do cenário principal do caso de uso contém uma informação de condição (“Estado do Sistema”) cujo valor é o nome da interrupção. O restante do fluxo é semelhante aos casos de uso normais, podendo, inclusive, possuir cenários alternativos, como mostra a Figura 3.8.

Uma vez que esse cenário de interrupção esteja especificado, assume-se que o mesmo pode ser executado a qualquer instante da execução de um outro cenário de caso de uso

qualquer. No entanto, alguns desses outros cenários podem não permitir que algumas interrupções ocorram ou até redefinir o comportamento para elas. Para tal, uma nova seção chamada *Interrupções* é adicionada ao caso de uso (Figura 3.9). Os campos “De” e “Para” da seção *Interrupções* indicam de onde o fluxo da interrupção está sendo continuado e a partir de que passo, o mesmo pode continuar, respectivamente. Note que esses campos não têm nenhuma relação com os passos das *features* que podem ser interrompidas. Os campos “De” e “Para” estão relacionados somente aos possíveis fluxos do comportamento da interrupção.

Interrupções

De:

Para:

Passos:

Interrupção:

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1I			

Figura 3.9: Redefinição de Comportamentos para Interrupções

Já a especificação dos passos para os quais o comportamento de uma interrupção está sendo alterado, é realizada através do campo “Passos”. Por fim, o campo “Interrupção” indica o nome da interrupção que está sendo redefinida. Com relação as colunas da tabela, todas elas têm a mesma semântica das colunas da tabela usada na especificação de *features* simples.

Para a especificação de cenários onde alguma interrupção não é permitida, preenche-se o campo “Interrupção” com os nomes das interrupções e na coluna “Resposta do Sistema” coloca-se o valor “NAO_PERMITIDA”. Se todas as interrupções não são permitidas, usa-se o valor “QUALQUER” no campo “Interrupção” e o valor “NAO_PERMITIDA” na coluna “Resposta do Sistema”.

Utilizando o caso de uso apresentado na Seção 3.1, especificamos alguns cenários de interrupção para o mesmo, como mostra a Figura 3.10.

Como foi explicado, podemos especificar interrupções que não são permitidas, seja especificamente uma ou todas, como no passo “2I” (Figura 3.10). Além disso, o comportamento para uma determinada interrupção pode ser redefinido, como ocorre no passo “1I” (Figura 3.10), onde, com a interrupção de chegada de uma mensagem, o comportamento do sistema será o de apresentar apenas uma tela informando que uma nova mensagem chegou. Esse comportamento é diferente do comportamento padrão (Figura 3.8), onde a opção de ler

Interrupções

De: INICIO

Para: FINAL

Passos: 5M

Interrupção: Chegada de uma Mensagem

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1I	Enviar uma mensagem de um outro celular para o celular sob teste		Aviso "Nova Mensagem" é mostrado

De: INICIO

Para: FINAL

Passos: 2A

Interrupção: QUALQUER

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
2I			NÃO PERMITIDA

Figura 3.10: Especificação de Comportamentos para Interrupções

a mensagem também é disponibilizada no momento da chegada da nova mensagem.

Na prática, com a especificação apresentada na Figura 3.10, sempre que uma nova mensagem chegar quando o usuário tiver selecionado a opção de mover uma mensagem para a pasta de mensagens prediletas, apenas um aviso de nova mensagem irá aparecer (Passo “1I” da primeira tabela da Figura 3.10). Além disso, quando a caixa de diálogo de memória insuficiente aparecer para o usuário, nenhuma interrupção poderá ocorrer antes que o usuário confirme a mensagem (Passo “2I” da segunda tabela da Figura 3.10).

Em resumo, a estratégia para especificação de interrupções em casos de uso consiste em:

- Especificar os cenários padrões para as interrupções na forma de casos de uso (Figura 3.8);
- Para cada passo dos casos de uso que podem sofrer interrupções, especificar na seção *Interrupções* (Figura 3.10) se as interrupções são permitidas ou não, e os comportamentos redefinidos para cada passo específico.

Com essa estratégia de especificação de interrupções, os comportamentos das diversas interrupções são definidos durante o desenvolvimento de cada *feature*, fazendo com que a forma com que as *features* são desenvolvidas, ou seja, de forma isolada, não seja alterada. Além disso, cada *feature* específica pode redefinir o comportamento de cada interrupção a cada passo dos casos de uso, podendo, inclusive, não permitir que elas ocorram.

3.4 Evolução dos *Templates*

Esta seção apresenta as alterações realizadas nos *templates* com o objetivo de agilizar o processo de especificação de interações e melhor adequar-se ao contexto do teste de interação em aplicações para celulares e, conseqüentemente, melhorar o processo de geração de casos de teste.

A primeira contribuição está relacionada a criação do campo “Passos” na seção *Interrupções*. Em [dFAM06], cada comportamento redefinido só poderia ser atribuído a um único passo, por exemplo, se quiséssemos redefinir um comportamento para vários passos de um caso de uso teríamos que replicar a redefinição do comportamento várias vezes. Com a nova modificação, por exemplo, se uma determinada *feature* redefine o comportamento da interrupção de chegada de uma nova mensagem para os passos “2M”, “3M” e “5M”, especificamos da forma que está em destaque na Figura 3.11.

Interrupções

De: INICIO

Para: FINAL

Passos: 2M, 3M, 5M

Interrupção: Chegada de uma Mensagem

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1I	Enviar uma mensagem de um outro celular para o celular sob teste		Aviso "Nova Mensagem" é mostrado

Figura 3.11: Exemplo da Primeira Alteração

A segunda contribuição diz respeito a padronização do primeiro passo da especificação de um caso de uso de interrupção. Em [dFAM06], o primeiro passo só continha a informação do campo “Estado do Sistema” e a respectiva “Resposta do Sistema”, já o campo “Ação do Usuário” deveria obrigatoriamente ficar vazio. A modificação proposta foi justamente preencher o campo “Ação do Usuário” (passo “1M” da Figura 3.8), pois, no contexto das aplicações para celulares, para que uma interrupção ocorra, algo precisa ser feito. Dessa forma, a especificação retrata melhor a realidade. A seguir, temos alguns exemplos de interrupções e seus respectivos eventos disparadores:

- Para que a interrupção de chegada de uma ligação ocorra, alguém tem que realizar uma chamada a partir de um outro aparelho telefônico;

- Para que a interrupção de chegada de uma mensagem ocorra, alguém tem que enviar uma mensagem a partir de um outro dispositivo;
- Para que a interrupção de um alarme do despertador ocorra, alguém tem que programá-lo antes;

A terceira contribuição está relacionada a não permissão da ocorrência de interrupções. Em [dFAM06], poderia-se negar a ocorrência somente de uma ou todas as interrupções. A segunda tabela da Figura 3.10 apresenta um cenário onde todas as interrupções não são permitidas. Com a alteração proposta, além de podermos especificar que uma ou todas as interrupções não são permitidas em um determinado passo, podemos também especificar que algumas interrupções não são permitidas, como mostra a Figura 3.12, onde somente as interrupções de chegada de uma ligação e chegada de uma mensagem não são permitidas.

Interrupções

De: INICIO

Para: FINAL

Passos: 2A

Interrupção: Chegada de uma Mensagem, Chegada de uma Ligação

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1I			NÃO PERMITIDA

Figura 3.12: Exemplo da Terceira Alteração

A quarta e última alteração proposta está relacionada a especificação dos pontos onde as interrupções podem ocorrer. Em [dFAM06], deveria-se especificar explicitamente se uma determinada interrupção deveria ocorrer depois da ação do usuário ou depois da resposta do sistema. Mas, na verdade, não se sabe exatamente em que tempo uma determinada interrupção pode ocorrer. Do ponto de vista do testador, só se pode observar uma interrupção após a resposta do sistema e como estamos considerando a abordagem funcional, só se pode avaliar o que pode ser observado, ou seja, a ocorrência de interrupções sempre após as respostas do sistema [JJ05]. Por isso, não é mais necessário especificar se uma interrupção deve ocorrer depois da ação do usuário ou depois da resposta do sistema.

3.5 Considerações Finais

Este capítulo apresentou a forma como os requisitos de *features* são especificados em casos de uso usando *templates*, desde a especificação de requisitos de *features* isoladas até os requisitos que envolvem mais de uma *feature*, como aqueles onde há interações estáticas e dinâmicas.

Como os *templates* apresentados são a informação de entrada para este trabalho, foram propostas algumas melhorias a fim de facilitar tanto o processo de especificação de interações quanto a geração do modelo LTS a partir dos mesmos. As melhorias apresentadas são fruto da experiência adquirida através da realização de estudos de caso.

Capítulo 4

Transformação de Casos de Uso em Modelos LTS

Este capítulo apresenta a estrutura do modelo comportamental LTS para *features* isoladas, obtido a partir da especificação de requisitos usando os *templates* apresentados no Capítulo 3. Esse modelo é usado para geração de casos de teste de *feature* usando o procedimento apresentado em [Car06], que foi automatizado através da ferramenta LTS-BT [CAN⁺07b].

Em seguida, é apresentada uma das principais contribuições deste trabalho, cujo objetivo é estender o modelo LTS usado pela ferramenta LTS-BT para que tanto as interações estáticas quanto as interações dinâmicas estejam presentes, e conseqüentemente, casos de teste possam ser gerados para verificar tais interações.

4.1 Modelos LTS de *Features* Simples

A estratégia geral para a geração de casos de teste de *feature* consiste em obter um modelo comportamental LTS a partir dos casos de uso da *feature*. Daí, usando uma ferramenta apropriada, casos de teste podem ser obtidos [Car06; CAN⁺07b].

O modelo obtido a partir do processamento de um caso de uso é uma variação de LTS, chamada LTS Anotado (Seção 2.4.1). As colunas do *template* “Ação do Usuário”, “Estado do Sistema” e “Resposta do Sistema” são mapeadas, respectivamente, para as seguintes anotações no modelo LTS: “Passos”, “Condições Iniciais” e “Resultados Esperados”. Daí, se $A = \langle Q, R, T, q_0 \rangle$ é um LTS Anotado obtido a partir de um caso de uso, então:

- Q é um conjunto finito e não-vazio de estados;
- $R = A \cup N$ é um conjunto finito de rótulos. A é o conjunto finito de informações retiradas do *template* (ações do usuário, estado do sistema e resposta do sistema). $N = \{\text{“Passos”}, \text{“Condições Iniciais”}, \text{“Resultados Esperados”}\}$ é o conjunto finito de anotações;
- $T \subseteq Q \times R \times Q$ é a relação de transição;
- $q_0 \in Q$ é o estado inicial.

A Figura 4.1 mostra o modelo LTS obtido a partir da especificação da *feature* Mensagens Prediletas (Figuras 3.3 e 3.4). Como podemos observar, o modelo obtido é um LTS Anotado, onde as colunas do *template* “Ação do Usuário”, “Estado do Sistema” e “Resposta do Sistema” foram mapeadas, respectivamente, para as seguintes anotações no modelo LTS: “Passos”, “Condições Iniciais” e “Resultados Esperados”. Como já foi mencionado na Seção 2.4.1, essas anotações são usadas para separar as informações inerentes ao processo de teste, facilitando assim, a geração automática de casos de teste.

No modelo LTS gerado, cada anotação é seguida pela sua respectiva informação, que está no *template*. Por exemplo, para o passo “1M” da Figura 3.3, tem-se a seguinte sequência de rótulos no modelo LTS: “Passos”, “Ir até a Central de Mensagens”, “Resultados Esperados” e “A pasta “Mensagens Prediletas” é mostrada”. Já a sequência dos vértices é definida através dos campos “De” e “Para” de cada tabela da especificação.

4.2 Modelos LTS para Interações Estáticas

Para mostrar como as interações estáticas são modeladas, as mesmas duas *features* apresentadas na Seção 3.2 serão utilizadas. Estamos interessados no cenário onde o usuário seleciona um número telefônico que está embutido em uma mensagem de texto recebida e o armazena em sua lista de contatos. Como já foi dito, a primeira parte desse cenário é especificado pela *feature* Números Embutidos e a segunda parte pela *feature* Contatos.

O procedimento para a representação da interação estática entre essas duas *features* é o seguinte: o LTS de cada *feature* é gerado separadamente e depois esses modelos são unidos.

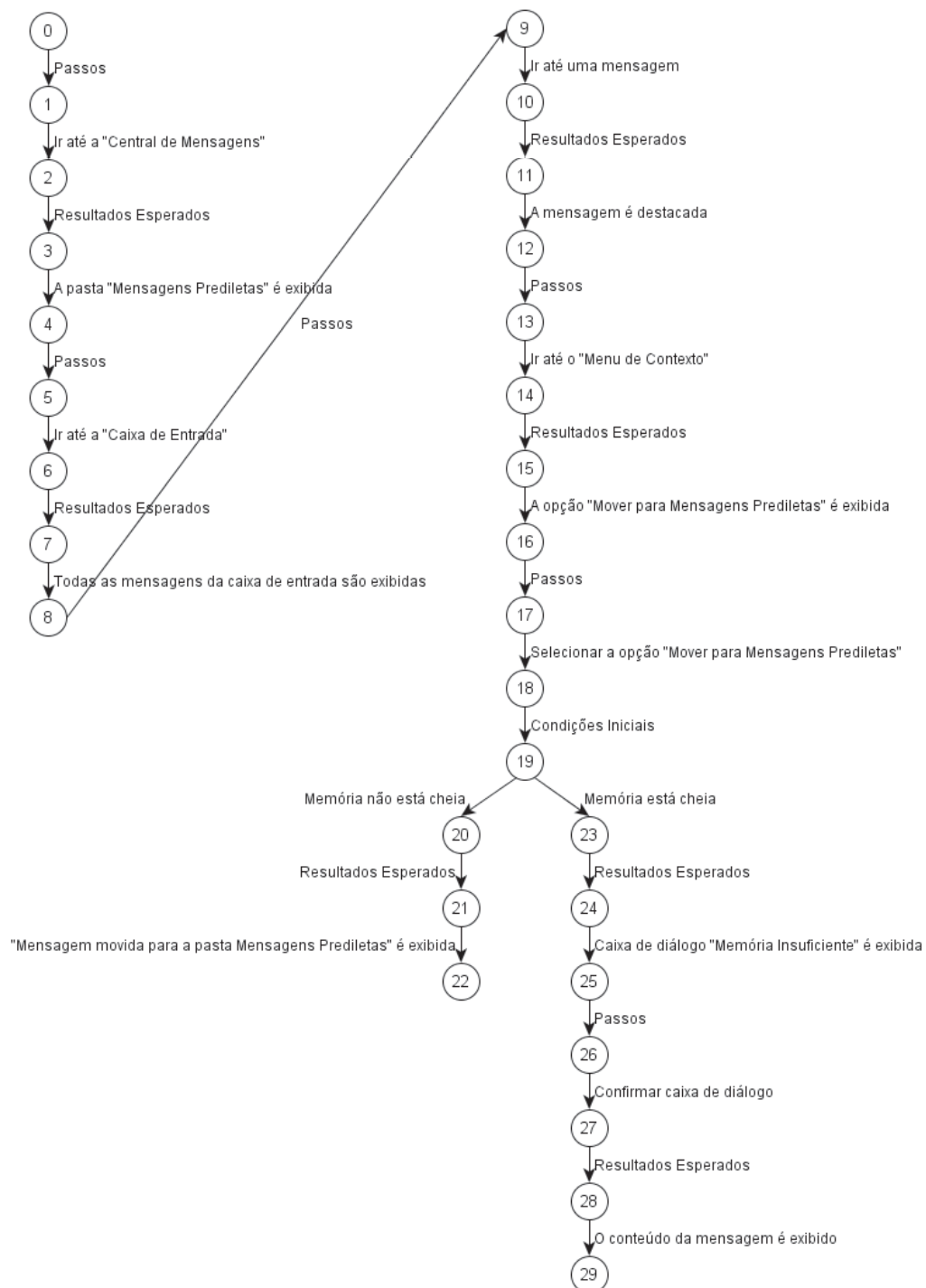


Figura 4.1: LTS que Representa um Caso de Uso da *Feature* Mensagens Prediletas

Para que seja possível gerar casos de teste de interação estática, os comportamentos das *features* envolvidas precisam estar representados num mesmo modelo LTS.

A Figura 4.2 modela a primeira parte do comportamento desejado, ou seja, a parte refe-

rente a *feature* Números Embutidos e a Figura 4.3 modela a segunda parte, referente a *feature* Contatos.

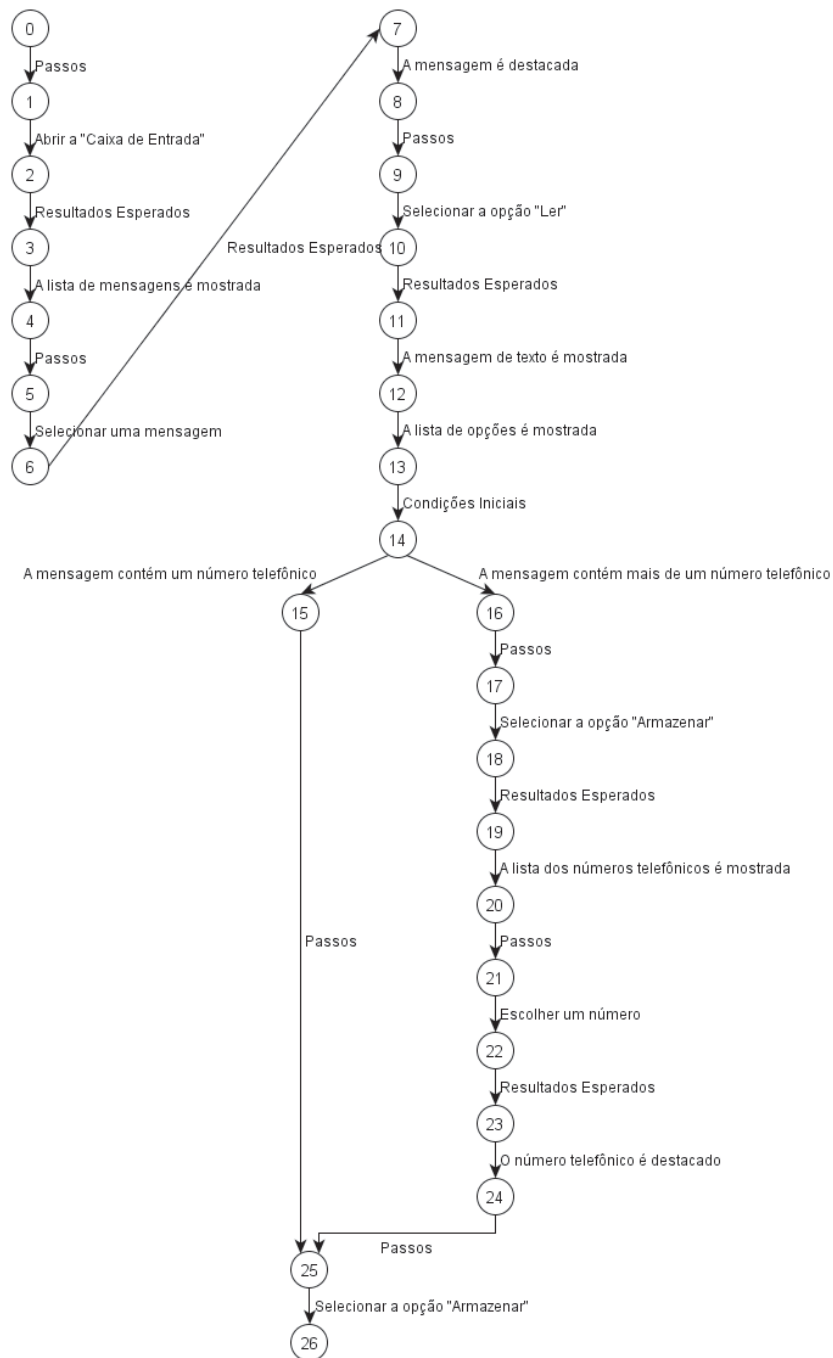


Figura 4.2: LTS da *Feature* Números Embutidos

O próximo passo é juntar os dois modelos LTS a partir do vértice de união. Esses vértices são identificados, usando os *templates*, a partir das referências que a *feature* Números

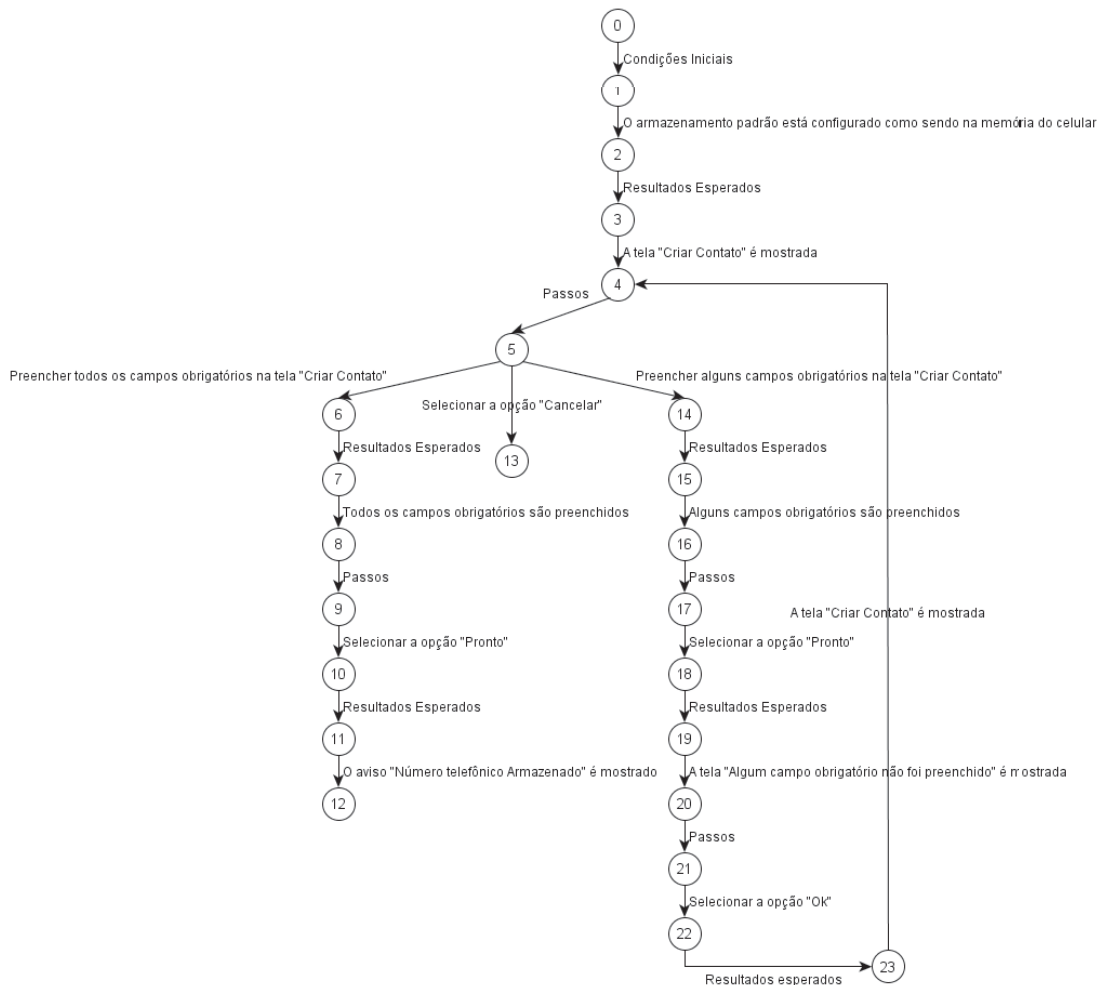


Figura 4.3: LTS que Representa a *Feature* Contatos

Embutidos faz a *feature* Contatos (campo “Para”, em destaque na Figura 3.5). Então o último vértice do LTS da Figura 4.2 (*feature* Números Embutidos) é o nosso vértice de união e ele precisa ser ligado ao primeiro vértice do LTS da Figura 4.3 (*feature* Contatos). Como um modelo LTS não pode conter vértices com números iguais, todos os vértices do LTS da *feature* Contatos (Figura 4.3) precisam ser renumerados. Para isto, soma-se o número do vértice de maior valor da primeira *feature* (Figura 4.2), que é 26, a todos os vértices da segunda *feature* (Figura 4.3). Daí, obtém-se o modelo LTS mostrado na Figura 4.4, onde os vértices de 0 a 23 da Figura 4.3 foram renumerados para 26 a 49.

Observando os modelos LTS das Figuras 4.2 e 4.4 podemos ver que os dois modelos ainda possuem um vértice com o mesmo número, o vértice 26. Então o próximo passo é eliminar esse vértice em comum e unir os dois modelos LTS da seguinte forma: o vértice 26

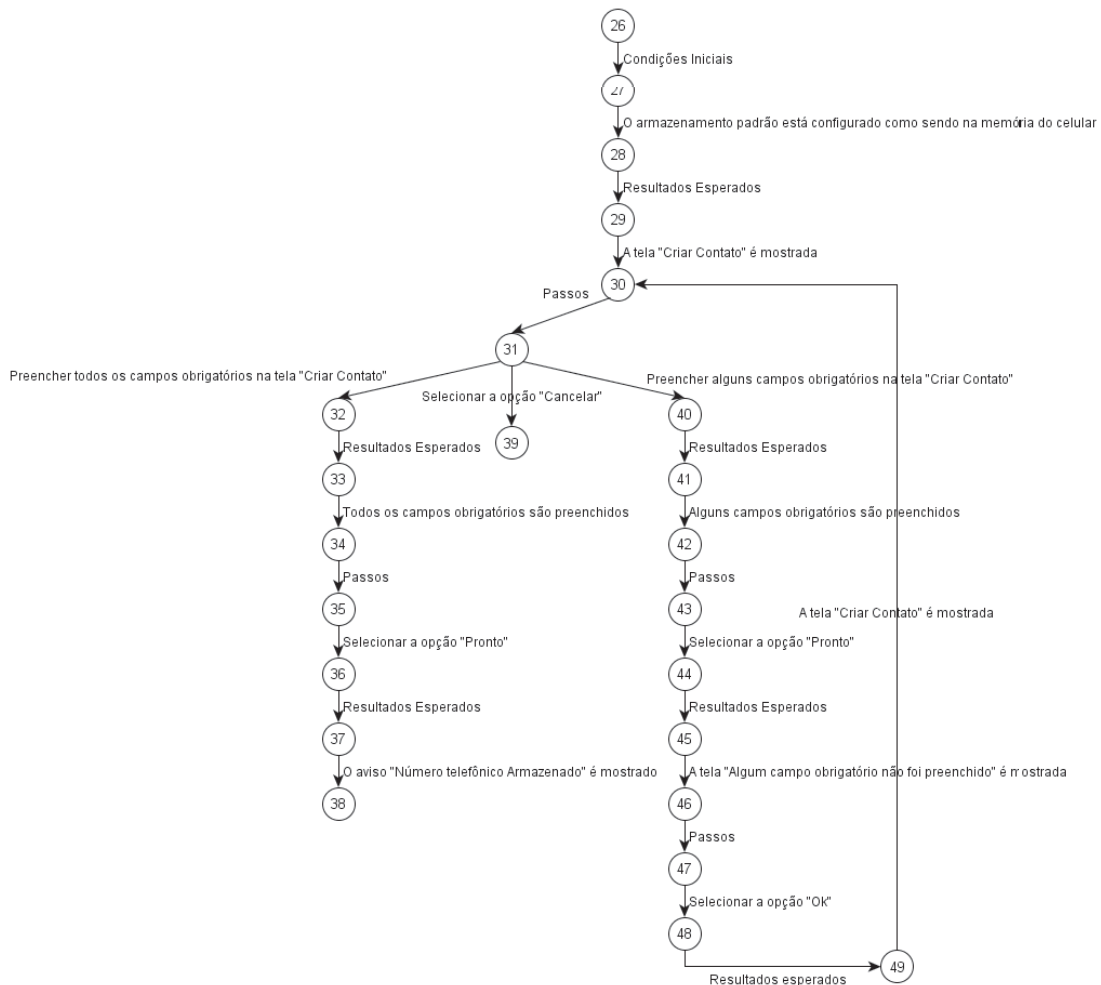


Figura 4.4: LTS da *Feature* Contatos com seus Vértices Renomeados

da *feature* Contatos (Figura 4.4) é removido e o rótulo que o conecta ao vértice 27 passa a ser o rótulo que conecta o vértice 26 da *feature* Números Embutidos (Figura 4.2) ao vértice 27 da *feature* Contatos (Figura 4.4). O resultado dessa manipulação pode ser visto na Figura 4.5. Dessa forma, obtemos um único modelo LTS que representa o cenário descrito no início da seção e contém os comportamentos das duas *features* envolvidas.

De forma geral, o procedimento para a geração de modelos LTS, com interações estáticas, especificados através de mais de um *template* é o seguinte:

1. Gerar o modelo LTS para cada *template* separadamente;
2. Renumerar os vértices do LTS que será anexado, somando-se a todos os seus vértices o número do vértice de maior valor do primeiro LTS;

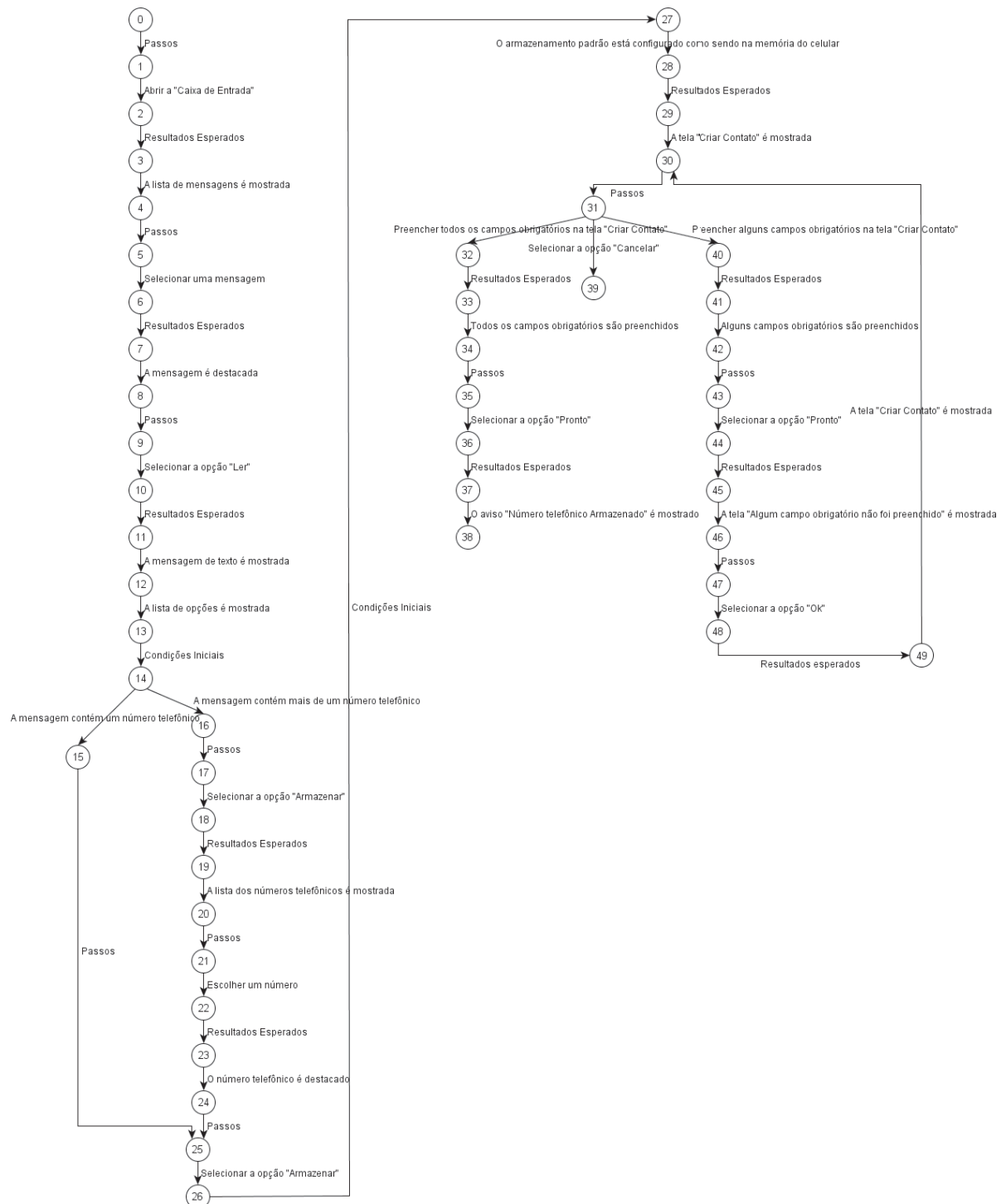


Figura 4.5: LTS Resultante

- Para conectar os dois modelos LTS, o primeiro vértice do LTS a ser anexado é removido e o rótulo que o conecta ao próximo vértice passa a ser o rótulo que conecta os dois modelos LTS, obtendo-se assim o LTS resultante;

4. Se ainda restarem modelos a serem anexados, isto é, se a interação se dá através de mais de duas *features*, então deve-se retornar ao passo 2 e considerar o modelo LTS resultante, obtido no passo 3, como sendo o primeiro LTS.

4.3 Modelos LTS para Interações Dinâmicas

O modelo LTS Anotado usado para representar um comportamento onde interrupções podem ocorrer, segue a mesma definição apresentada na Seção 4.1, com uma simples modificação: o conjunto de anotações é acrescido de mais dois tipos de anotações, onde uma é usada para indicar o início e a outra é usada para indicar o fim de uma interrupção.

A estrutura do modelo LTS que captura o comportamento das interações dinâmicas será apresentada através da *feature* Mensagens Prediletas, cuja especificação foi apresentada na Seção 3.1 e o respectivo modelo LTS na Seção 4.1, e uma nova *feature* responsável pelas interrupções. Essa nova *feature* especifica o comportamento de uma interrupção causada pela chegada de um aviso, um tipo específico de mensagem, onde o texto aparece numa caixa de diálogo para o usuário. A especificação e o modelo dessa interrupção são mostrados, respectivamente, nas Figuras 4.6 e 4.7.

Cenário Principal

Descrição: O aviso é exibido numa caixa de diálogo

De: INICIO

Para: FINAL

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1M	Enviar um aviso de um outro celular para o celular sob teste	Chegada de um Aviso	O aviso é exibido numa caixa de diálogo
2M	Selecionar a opção "Ok"		Volta para a aplicação que estava executando anteriormente

Figura 4.6: Especificação da Interrupção Causada pela Chegada de um Aviso

Como foi explicado na Seção 3.4, as interrupções só podem ser verificadas se ocorrerem entre os passos do caso de uso, ou seja, sempre depois da resposta do sistema e antes da próxima ação do usuário. Também foi dito na Seção 3.3 que se não há nenhuma tabela redefinindo o comportamento das interrupções nem proibindo a ocorrência das mesmas, então todas as interrupções especificadas podem ocorrer entre todos os passos do caso de uso. No entanto, como se trata de um modelo de testes, estaremos considerando apenas as *features*

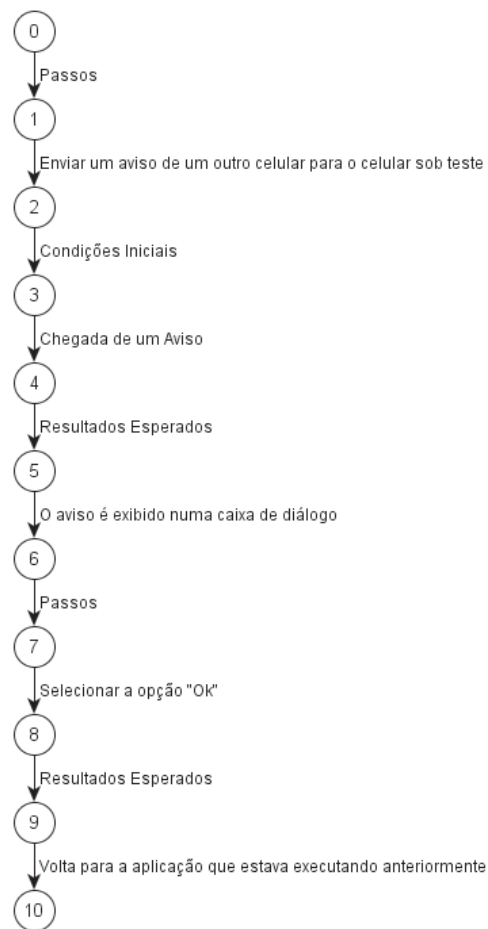


Figura 4.7: Modelo LTS da Interrupção Causada pela Chegada de um Aviso

de interesse e não qualquer feature existente. Dessa forma, temos um modelo de comportamento parcial.

Através das especificações das *features* Mensagens Prediletas (Figuras 3.3 e 3.4) e Chegada de um Aviso (Figura 4.6) é gerado o LTS apresentado na Figura 4.8. Os vértices numerados de 0 a 29 representam a *feature* principal (Mensagens Prediletas) e os vértices de 30 a 40 representam a interrupção (Chegada de um Aviso).

Como podemos ver, na Figura 4.8, duas novas anotações foram inseridas no modelo LTS: *Início da Interrupção X* e *Final da interrupção X*, onde X é um contador do tipo inteiro. Essas anotações são usadas para memorizar em que ponto a *feature* principal, Mensagens Prediletas, foi interrompida. Dessa forma, o fluxo de execução da *feature* principal pode continuar a partir do mesmo ponto em que houve a interrupção. Por exemplo, se a interrupção iniciar através do rótulo *Início da Interrupção 0*, então o fluxo de execução da interrupção

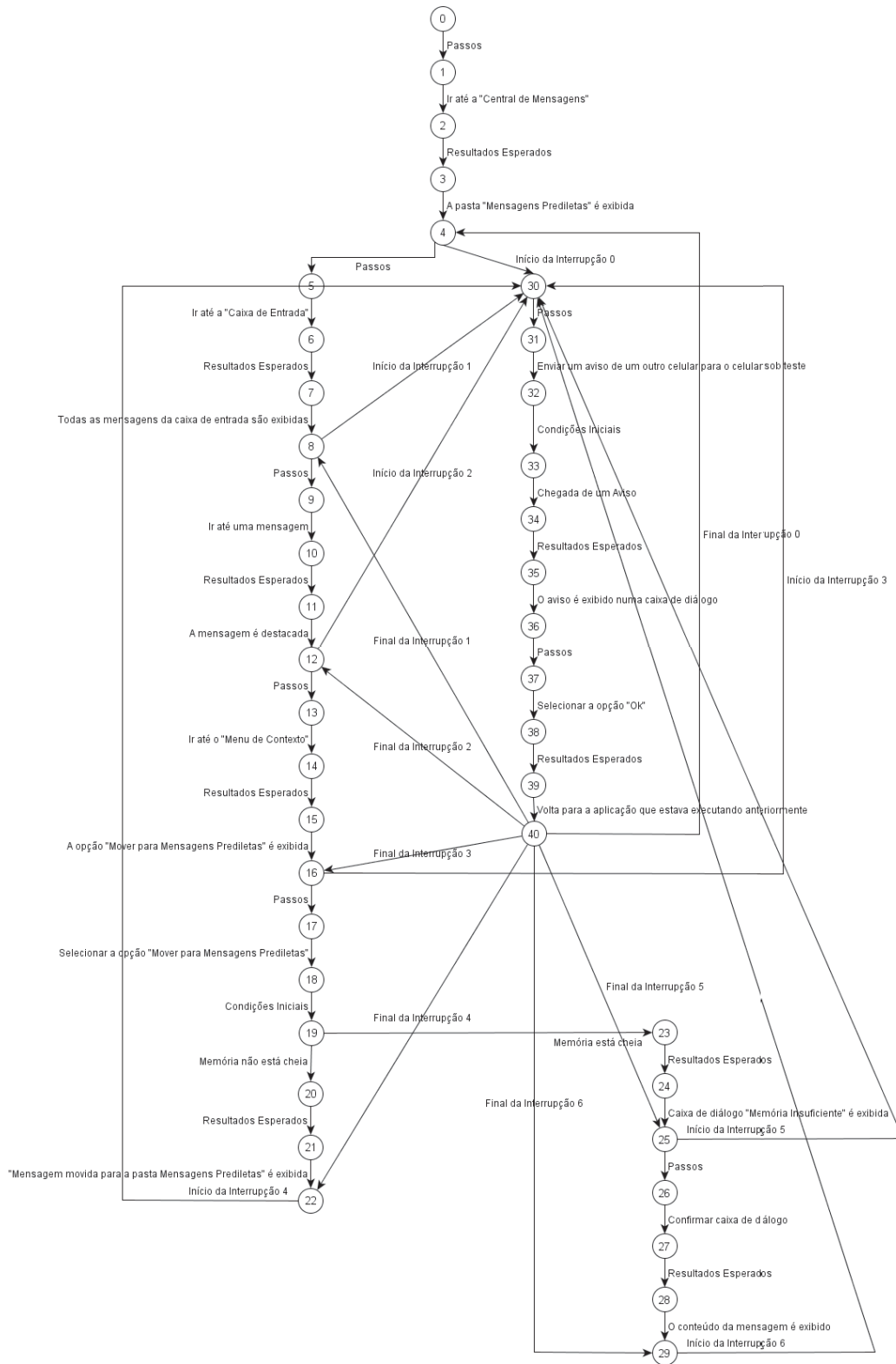


Figura 4.8: Modelo LTS com Interrupções

deve terminar através do rótulo *Final da Interrupção* 0; outro exemplo, se a interrupção iniciar através do rótulo *Início da Interrupção* 1, então o fluxo de execução deve terminar através do rótulo *Final da Interrupção* 1; e assim por diante.

Ainda utilizando a *feature* Mensagens Prediletas e a interrupção Chegada de um Aviso, serão apresentados alguns exemplos de cenários onde a seção *Interrupções*, do *template*, é utilizada, mostrando os seus respectivos modelos LTS gerados. No primeiro exemplo, considere um cenário onde não poderão ocorrer interrupções a partir do momento em que o usuário selecionar a opção de mover uma mensagem para pasta de mensagens prediletas com a memória do celular cheia. A Figura 4.9 mostra a seção *Interrupções* que especifica esse cenário e a Figura 4.10 apresenta o modelo LTS gerado.

Interrupções

De: INICIO

Para: FINAL

Passos: 1A, 2A

Interrupção: QUALQUER

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1I			NÃO_PERMITIDA

Figura 4.9: Seção *Interrupções* para o Primeiro Cenário

Comparando a Figura 4.8, onde as interrupções podem ocorrer em qualquer momento, com a Figura 4.10, podemos observar que nesta última não podem ocorrer interrupções partindo dos vértices 25 e 29, como foi especificado no primeiro exemplo de utilização da seção *Interrupções*.

No segundo cenário, sempre que o usuário estiver acessando o menu de contexto e chegar um novo aviso, apenas uma tela com o texto do aviso irá aparecer e depois desaparecerá automaticamente. Nesse segundo exemplo, o comportamento da interrupção é alterado em um único passo, o passo “4M” (Figura 4.11), enquanto que, para todos os outros passos o comportamento é o comportamento padrão, especificado na Figura 4.6. A Figura 4.12 mostra o modelo LTS que representa o segundo cenário. Note que o novo comportamento da interrupção está representado através dos vértices numerados de 41 a 47.

O terceiro e último cenário une o primeiro e o segundo exemplo, ou seja, a interrupção Chegada de um Aviso é redefinida para o passo “4M” e não é permitida nos passos “1A” e “2M”, como mostra a seção *Interrupções* na Figura 4.13. A Figura 4.14 mostra o modelo LTS gerado para esse terceiro cenário.

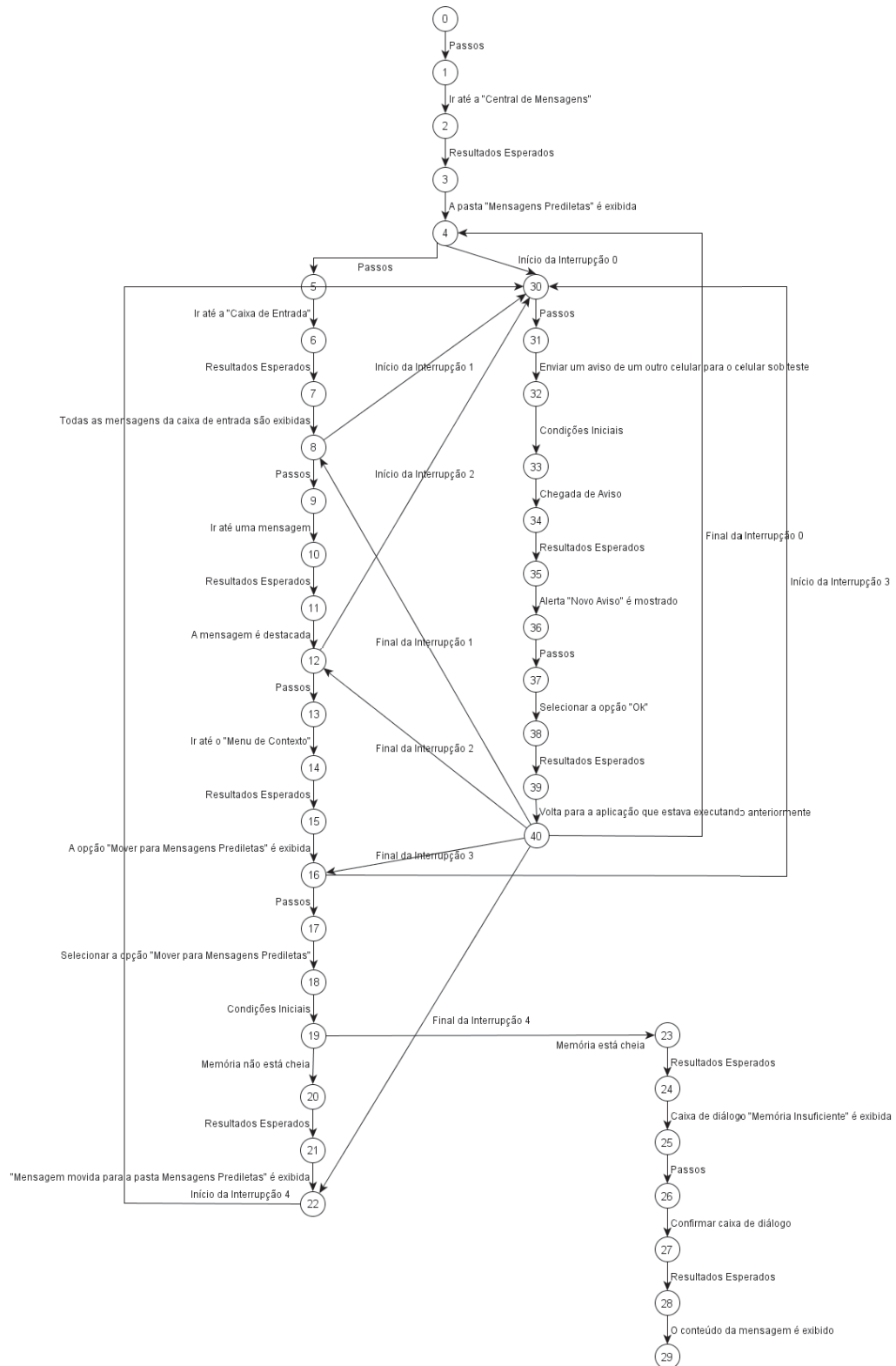


Figura 4.10: Modelo LTS do Primeiro Cenário

Interrupções

De: INICIO

Para: FINAL

Passos: 4M

Interrupção: Chegada de um Aviso

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1I	Enviar um aviso de um outro celular para o celular sob teste		O aviso é exibido numa caixa de mensagem temporária

Figura 4.11: Seção *Interrupções* para o Segundo Cenário

Em resumo, a estratégia para a geração de modelos LTS com interações dinâmicas é a seguinte:

1. Gerar o modelo LTS da *feature* principal e das interrupções separadamente;
2. Renumerar os vértices dos modelos das interrupções, de forma que não haja números repetidos no modelo LTS resultante;
3. Para cada vértice da *feature* principal onde é permitida a ocorrência de uma determinada interrupção, deve haver uma aresta conectando esse vértice ao vértice inicial do modelo da interrupção;
4. O rótulo da aresta que liga um vértice da *feature* principal a interrupção deve ser *Início da Interrupção X*, onde X é um número inteiro;
5. Cada aresta com o rótulo *Início da Interrupção X* deve possuir um valor diferente para X ;
6. Deve haver uma aresta conectando todos os vértices finais do modelo da interrupção ao mesmo vértice da *feature* principal a partir da qual a interrupção teve início;
7. O rótulo das arestas que conectam os vértices finais do modelo LTS da interrupção ao vértice da *feature* principal deve ser *Final da Interrupção X*, onde o valor de X deve ser igual ao valor utilizado no rótulo de início da interrupção. Por exemplo, se uma interrupção iniciar com o rótulo *Início da Interrupção 2*, então todos os seus vértices finais devem conter o rótulo *Final da Interrupção 2* e devem ser conectados ao mesmo vértice da *feature* principal de onde partiu a aresta com o rótulo *Início da Interrupção 2*.

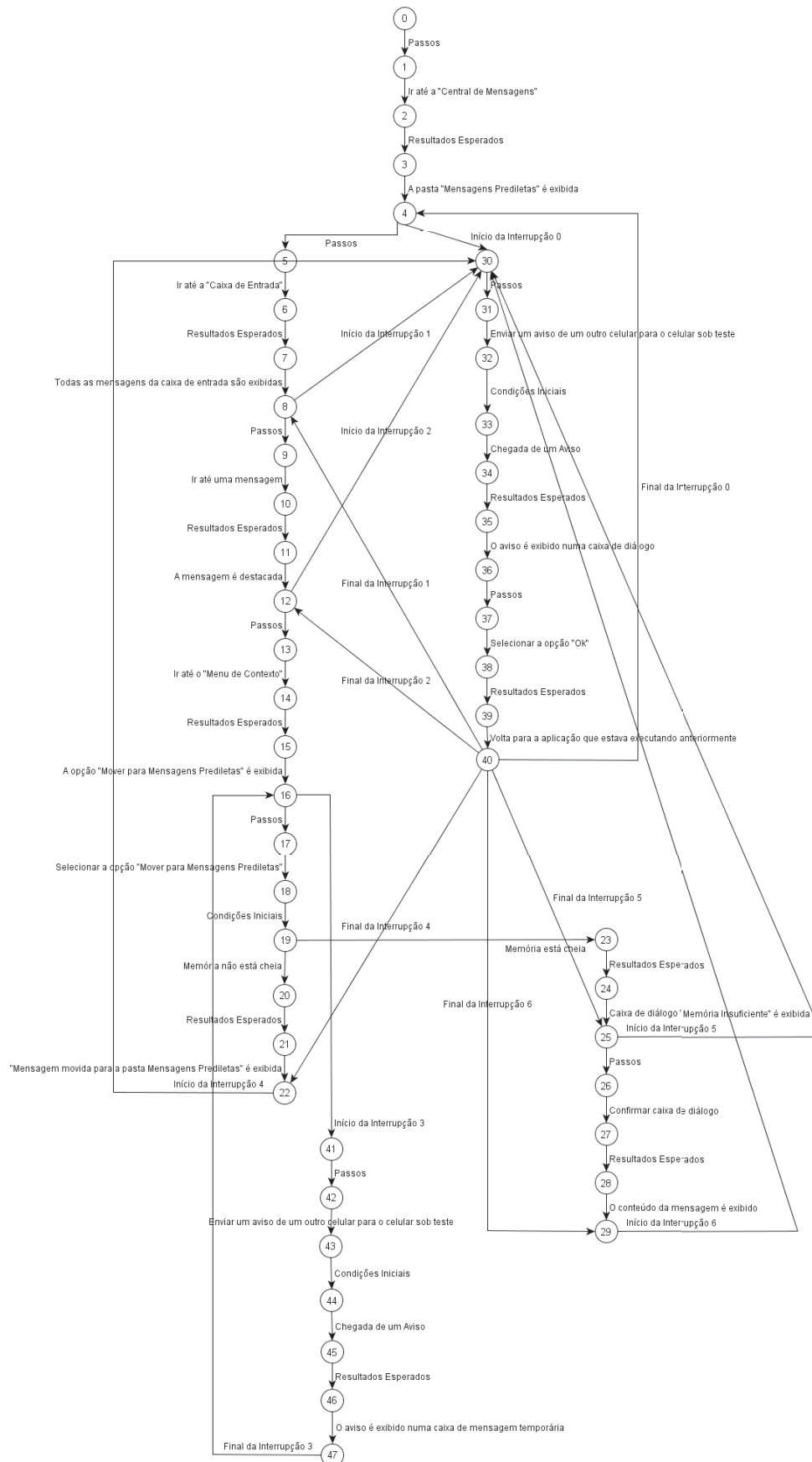


Figura 4.12: Modelo LTS do Segundo Cenário

Interrupções

De: INICIO

Para: FINAL

Passos: 4M

Interrupção: Chegada de um Aviso

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1I	Enviar um aviso de um outro celular para o celular sob teste		O aviso é exibido numa caixa de mensagem temporária

De: INICIO

Para: FINAL

Passos: 1A, 2A

Interrupção: QUALQUER

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
2I			NÃO_PERMITIDA

Figura 4.13: Seção *Interrupções* para o Terceiro Cenário

4.4 Considerações Finais

Este capítulo apresentou a estrutura do modelo LTS usado por LTS-BT para representar *features* isoladas e uma estratégia para a extensão desse modelo LTS, de tal forma que, tanto o comportamento expresso pelas interações estáticas quanto o das interações dinâmicas estejam presentes. Como o modelo LTS agora captura o comportamento das interações, casos de teste podem ser gerados com o objetivo de verificar tais interações.

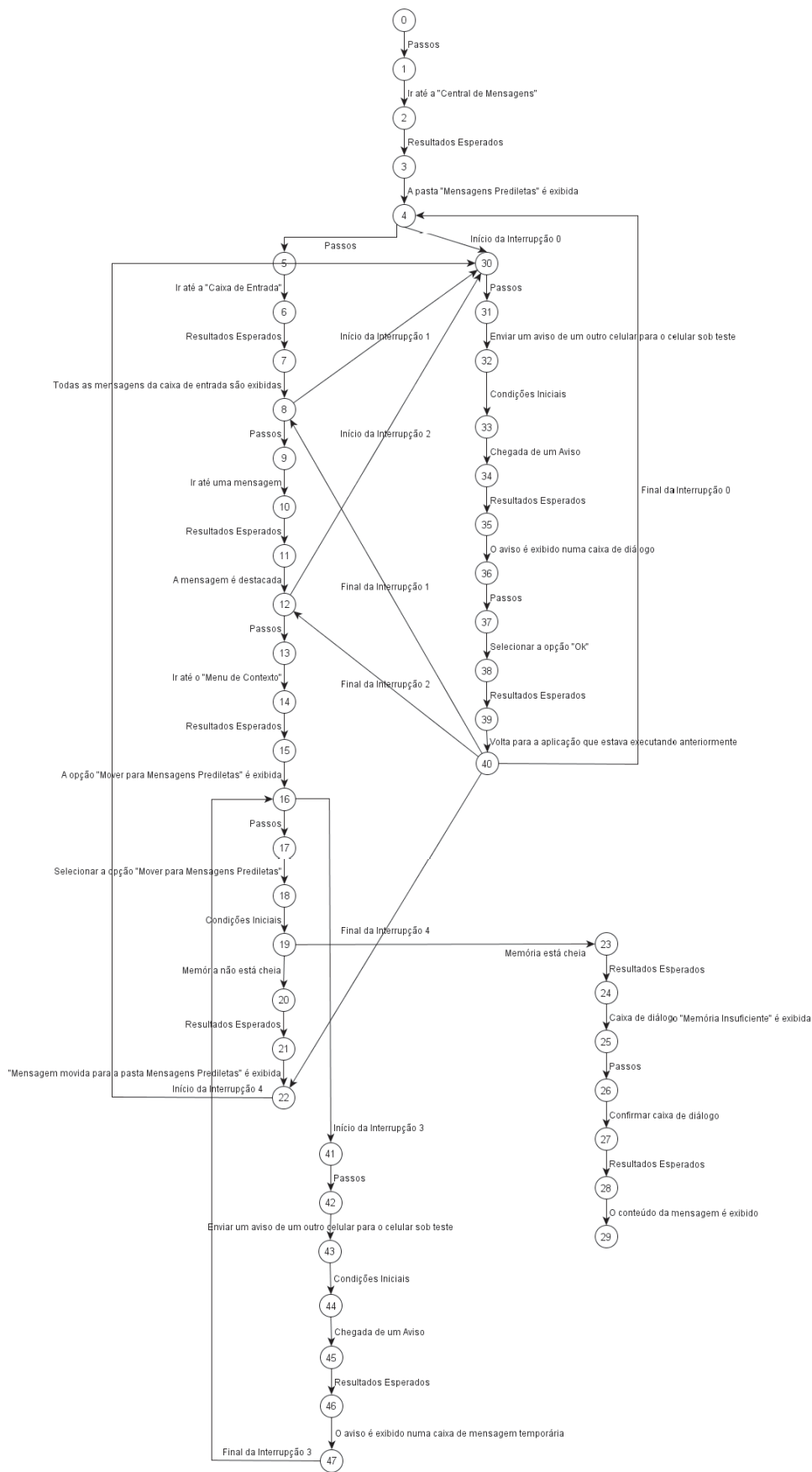


Figura 4.14: Modelo LTS do Terceiro Cenário

Capítulo 5

Geração e Seleção de Casos de Teste de Interação

Este capítulo apresenta mais uma contribuição deste trabalho, que é a extensão do procedimento sistemático apresentado em [Car06] visando a geração de casos de teste de interação a partir de um modelo LTS que segue a estrutura apresentada no capítulo anterior [CAN⁺07a].

Em seguida, uma estratégia de seleção de casos de teste de interação baseada em propósitos é apresentada [CAN⁺07a], visto que, o contexto das aplicações para celulares não permite a execução de toda a suíte de teste obtida a partir de uma geração exaustiva devido a restrições de tempo e recursos.

5.1 Casos de Teste de Interação

Casos de teste de interação têm algumas peculiaridades que os diferem dos casos de teste de *features* isoladas. A Figura 5.1 apresenta um caso de teste de *feature* isolada. O caso de teste da Figura 5.1 foi obtido a partir do modelo LTS da *feature* Mensagens Prediletas, apresentado na Seção 4.1. Como pode ser observado, na Figura 5.1, o caso de teste só contém ações referentes a *feature* Mensagens Prediletas.

Casos de teste de interação estática apresentam passos onde a ação do usuário pertence a uma *feature*, que solicita um serviço ou dados, e a respectiva resposta do sistema é de responsabilidade de outra *feature*, a *feature* que está sendo solicitada. A Figura 5.2 apresenta um caso de teste de interação estática. Nesse exemplo, a interação ocorre no último passo

Passos	Resultados Esperados
Ir até a Central de Mensagens	A pasta "Mensagens Prediletas" é exibida
Ir até a "Caixa de Entrada"	Todas as mensagens da "Caixa de Entrada" são exibidas
Ir até uma mensagem	A mensagem é destacada
Ir até o "Menu de Contexto"	A opção "Mover para Mensagens Prediletas" é exibida
Selecionar a opção "Mover para Mensagens Prediletas"	"Mensagem movida para a pasta Mensagens Prediletas" é exibida

Figura 5.1: Exemplo de um Caso de Teste Simples

do caso de teste, ou seja, a *Feature* Contatos solicita um serviço da *Feature* Ligação, onde o serviço solicitado é a realização de uma ligação.

Passos	Resultados Esperados
Ir até a Lista de Contatos	A Lista de Contatos é exibida
Selecionar um contato	Um contato é selecionado
Pressionar a tecla "Ligar"	A ligação é iniciada

Figura 5.2: Exemplo de Caso de Teste de Interação Estática

Uma característica interessante dos casos de teste de interação dinâmica é que suas execuções geralmente demandam dois celulares. Veja o exemplo do caso de teste de interação dinâmica da Figura 5.3. Além desse caso de teste verificar o comportamento descrito no caso de teste da Figura 5.1, há a ocorrência de uma ligação após o primeiro passo do caso de teste. No caso de teste da Figura 5.3, os passos 2,3 e 4 representam a interrupção.

5.2 Geração de Casos de Teste de Interação

A partir dos modelos LTS que capturam os comportamentos das interações, podemos gerar casos de teste de interação para as aplicações de celulares. Para isso, o procedimento apresentado em [Car06] foi estendido para dar suporte a geração de casos de teste de interação, assim como, a ferramenta LTS-BT [CAN⁺07b], desenvolvida para automatizar o procedimento apresentado em [Car06], também foi estendida. Cartaxo [Car06] apresenta um procedimento

Passos	Resultados Esperados
Ir até a Central de Mensagens	A pasta "Mensagens Prediletas" é exibida
Fazer uma ligação de um outro celular para o celular sob teste	Mostra alerta de Chegada de uma Ligação
Selecionar a opção "Atender"	A conversa é iniciada
Selecionar a opção "Parar"	Aviso "Ligação encerrada" é mostrado. Volta para a aplicação que estava executando antes da interrupção do aviso de chegada de uma ligação
Ir até a "Caixa de Entrada"	Todas as mensagens da "Caixa de Entrada" são exibidas
Ir até uma mensagem	A mensagem é destacada
Ir até o "Menu de Contexto"	A opção "Mover para Mensagens Prediletas" é exibida
Selecionar a opção "Mover para Mensagens Prediletas"	"Mensagem movida para a pasta Mensagens Prediletas" é exibida

Figura 5.3: Exemplo de Caso de Teste de Interação Dinâmica

para geração de casos de teste de *feature* usando o método de busca em profundidade (*Depth First Search* - DFS).

O critério de cobertura adotado é a cobertura de caminhos, para garantir que todos os comportamentos das aplicações possam ser testados. Portanto, um caso de teste é um caminho no modelo LTS partindo do vértice inicial e terminando em um vértice final.

A partir de agora, casos de teste de interação são obtidos através de modelos LTS que capturam os comportamentos das interações. Um caso de teste de interação pode ou não considerar as interrupções. As interrupções são consideradas no teste de interação dinâmica e podem ser usadas de maneira facultativa no teste de interação estática, dependendo do objetivo do testador. Um mesmo algoritmo básico será usado para geração de casos de teste, mas sua aplicação será diferente dependendo do tipo de teste a ser realizado (teste de interação estática ou dinâmica).

Dado que a abordagem considerada é a de teste funcional, a cobertura total é um objetivo razoável e viável para o teste de *features* isoladas visando garantir a verificação completa das funcionalidades da *feature* [MS01]. Entretanto, isso nem sempre é um objetivo alcançável para o teste de interação, principalmente nos casos que envolvem interrupções. Por isso, técnicas de seleção de casos de teste devem ser aplicadas. A Figura 5.4 apresenta o algoritmo

para geração de casos de teste.

```

Decompose(vertex, path, interruptionModel)

IF vertex.isLeaf() OR (vertex.isRoot() AND path ≠ ∅) THEN
    // End of a path
    recordTestCase(path)
    RETURN
FOR ∀ descendent ∈ vertex.getAdjacencies() DO
    Edge ← getEdgeBetween(vertex, descendent)
    IF edge ∉ path OR edge ∈ interruptionModel.getEdges() THEN
        path.add(edge)
        Decompose(descendent, path, interruptionModel) // DFS
    ELSE recordTestCase(path)
RETURN

```

Figura 5.4: Algoritmo de Geração de Casos de Teste

O algoritmo apresentado na Figura 5.4 utiliza três parâmetros de entrada:

- *vertex* é o vértice do modelo LTS a partir do qual o DFS será executado;
- *path* é um conjunto de arestas do modelo que representa o caminho visitado durante a execução do DFS;
- *interruptionModel* é um conjunto de arestas usado para separar o modelo LTS da *feature* principal do modelo LTS das interrupções.

A geração de casos de teste tem início a partir do vértice raiz (o vértice de número 0), verificando se o vértice atual (valor da variável *vertex*) é o fim de um caminho do modelo, indicando que um caso de teste foi extraído. Sendo assim, o caso de teste é armazenado. Se o vértice atual não é o fim de um caminho, então cada um dos vértices adjacentes serão visitados através do algoritmo DFS.

Para visitar cada vértice adjacente, a aresta entre o vértice atual e seu adjacente é analisada. A busca prossegue somente se: (i) a aresta não pertence ao caminho que está sendo extraído, ou (ii) se a aresta é uma das arestas do modelo LTS de uma interrupção (uma aresta com rótulo *Final da Interrupção*). Essa precaução é necessária devido ao fato de após a interrupção ser executada, a busca deve continuar a partir do vértice de onde a interrupção teve início. Baseando-se nas duas condições acima, dois cenários podem ser encontrados:

- As condições (i) e (ii) não são satisfeitas: a busca pára e o caminho percorrido é armazenado. Nesse caso, a recursão do algoritmo retorna para o próximo caminho a ser visitado, continuando a busca;
- As condições (i) ou (ii) são satisfeitas: a aresta entre o vértice atual e o seu descendente é adicionada ao caso de teste e o DFS continua até encontrar o final do caminho, que ocorre quando uma folha é encontrada ou quando uma aresta voltando para um vértice já visitado é encontrada.

Essas restrições impostas a geração de casos de teste são necessárias para evitar uma explosão de caminhos durante a execução do algoritmo de busca causado por eventuais laços no modelo LTS. Dessa forma, o número de casos de teste é reduzido, mas sem essas restrições, o número de casos de teste gerados torna-se impraticável, e mais, a maioria deles pode ser obtida através da combinação dos casos de teste gerados usando as restrições. Na prática, os casos de teste gerados sem as restrições adicionam redundância a suíte de teste.

Para demonstrar a aplicação do algoritmo apresentado na Figura 5.4, vamos gerar casos de teste de interação dinâmica para o modelo da Figura 4.10. As Figuras A.1 a A.9 (Apêndice A) apresentam somente os casos de teste gerados com uma única interrupção. Note que não há casos de teste com interrupção a partir do momento em que o usuário seleciona a opção de mover uma mensagem para pasta de mensagens prediletas com a memória do celular cheia, como especifica o modelo.

5.3 Seleção de Casos de Teste de Interação

A cobertura total para o teste de interação é impraticável devido a grande quantidade de casos de teste gerada e as características associadas ao contexto das aplicações de celulares, onde há fortes restrições de tempo e recursos, e além disso, a grande maioria dos casos de teste são executados manualmente.

A técnica empregada para a redução da suíte de testes é a seleção baseada em propósitos. Essa técnica foca na cobertura de um critério de seleção, o propósito de teste, visando testar uma funcionalidade particular do sistema. Considerando a utilização de propósitos de teste, os casos de teste de interação obtidos são consistentes no sentido de que um veredito é

negativo somente se a implementação não satisfaz o propósito definido [FMP03]. E mais, para se manter a consistência, considera-se que nenhuma interrupção não prevista no teste ocorra durante a execução do mesmo.

A mesma notação utilizada por LTS-BT [CAN⁺07b], apresentada na Seção 2.4.3, será utilizada para a definição de propósitos de teste com o objetivo de testar interações. A notação utilizada por LTS-BT é uma simplificação da notação utilizada por TGV [JJ05], que utiliza propósitos de teste definidos através de modelos LTS. Dado que o testador precisa manipular modelos LTS na definição de propósitos de teste em TGV, essa notação não é muito útil na prática. Modelos LTS são usados por LTS-BT como modelos extraídos dos casos de uso e como os propósitos são definidos em alto nível, o testador não precisa manipular nenhum modelo LTS.

A Figura 5.5 mostra o fluxo da atividade de geração de casos de teste usando propósitos. Para que a seleção de casos de teste possa ser realizada obedecendo a um determinado propósito de teste, o modelo LTS e o propósito devem ser dados como entrada. O propósito de teste é utilizado para limitar o modelo, depois o algoritmo de geração apresentado na Seção 5.2 é aplicado, para então, gerar os casos de teste.

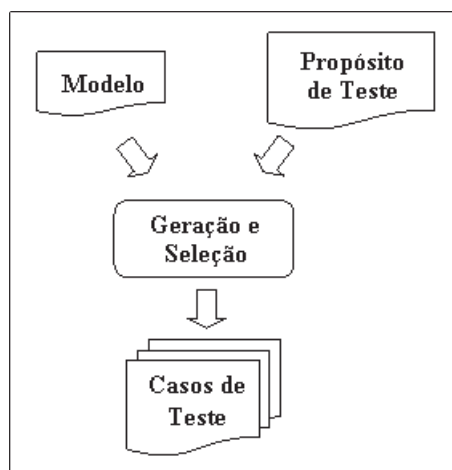


Figura 5.5: Fluxo da Seleção de Casos de Teste Baseada em um Propósito

A seguir, serão apresentados alguns exemplos de como definir propósitos de teste com o objetivo de testar interações estáticas e dinâmicas separadamente.

5.3.1 Seleção de Casos de Teste de Interação Estática

Como foi visto na Seção 4.2, modelos LTS que contém interações estáticas possuem mais de uma *feature*. A seleção de casos de teste de interação estática depende especificamente do propósito utilizado. Em interações estáticas, há a *feature* que solicita o serviço ou dados e há a *feature* que é solicitada, ou seja, a *feature* que provê o serviço ou dados. Se o testador escolher um propósito que especifica um cenário da *feature* que é solicitada, então as interações estáticas estarão presentes nos caminhos selecionados.

O número de interações estáticas num caso de teste depende do número de *features* do modelo. Por exemplo, se uma determinada *feature* A solicita um serviço de uma *feature* B, que por sua vez, solicita dados de uma outra *feature* C e o testador define um propósito relacionado a um cenário da *feature* C, o caso de teste conterá duas interações. Se porventura o testador quiser testar uma única interação, então o modelo deverá ser restringido.

Vamos usar o modelo gerado na Seção 4.2 (Figura 4.5) para demonstrar como definir propósitos de teste para verificar as interações estáticas presentes no modelo LTS. Duas *features* foram usadas na geração do modelo: Números Embutidos e Contatos. Observando o modelo gerado (Figura 4.5) podemos ver que o vértice de união das *features* é exatamente o vértice 26, então basta definir propósitos que levem a cenários definidos após o vértice 26. Para isso, basta escolher qualquer cenário da *feature* que é solicitada, ou seja, da *feature* Contatos.

O primeiro propósito será definido com o objetivo de testar os cenários que levam o usuário a visualizar a mensagem “Número telefônico Armazenado”. Para esse cenário, temos o seguinte propósito: “*;O aviso "Número telefônico Armazenado"é mostrado;*;Aceitar”. Após a definição do propósito, o modelo é limitado para satisfazê-lo, e assim, a aplicação do algoritmo de geração resultará somente em casos de teste que satisfazem o propósito definido inicialmente. A Figura 5.6 apresenta o modelo obtido a partir da aplicação do propósito definido e a Figuras 5.7 e 5.8 apresentam os casos de teste obtidos. Note que o ponto onde ocorre a interação estática está presente nos dois casos de teste gerados, ou seja, os casos de teste verificam se quando o usuário seleciona a opção “Armazenar” da *feature* Números Embutidos, a tela para a criação de um novo contato, da *feature* Contatos, é exibida.

No segundo exemplo de propósito de teste, vamos testar o cenário onde o usuário escolhe uma mensagem que possui apenas um número embutido e o armazena na lista de con-

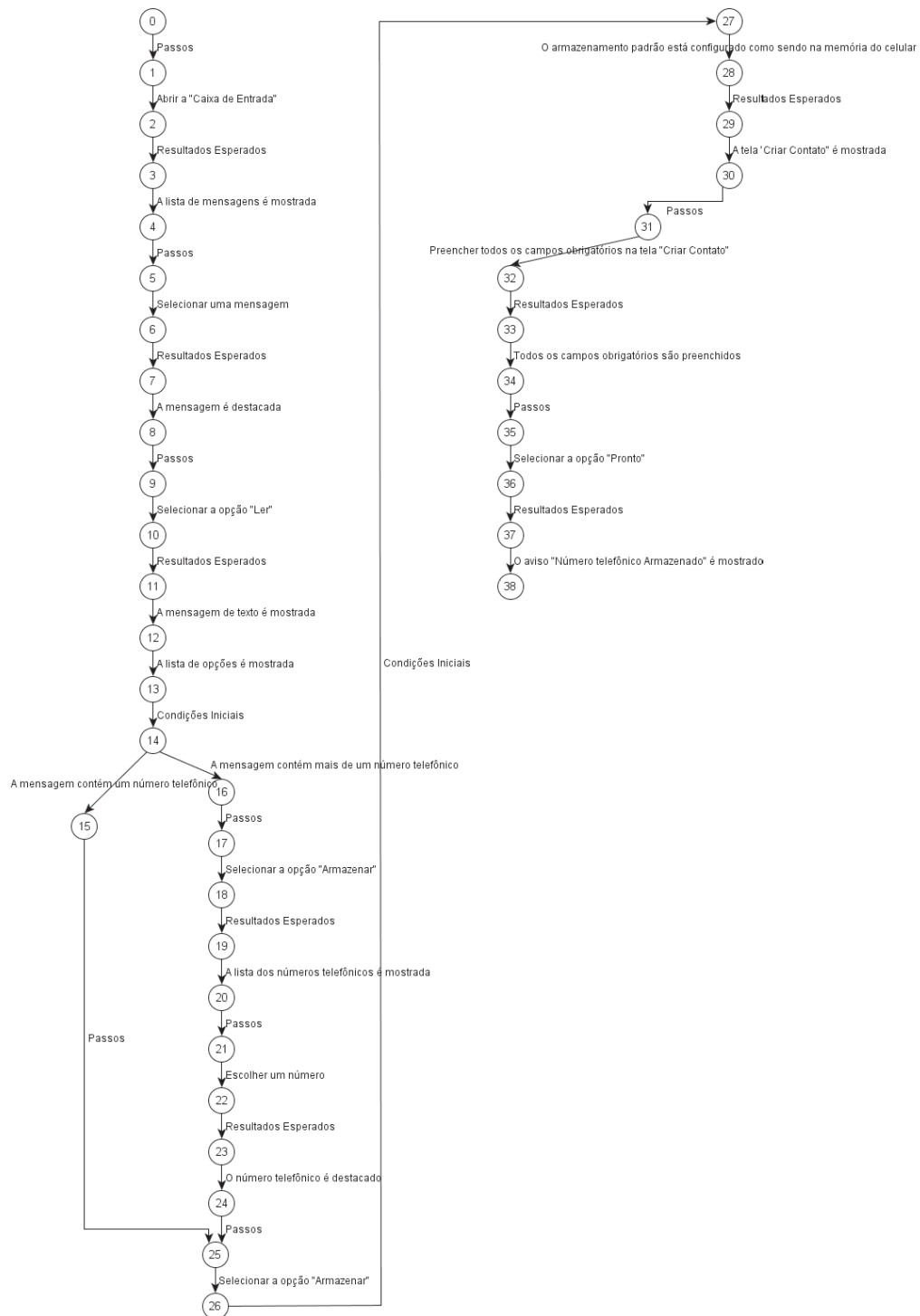


Figura 5.6: Modelo LTS Obtido com o Propósito: “*;O aviso "Número telefônico Armazenado" é mostrado*;Aceitar”

tatos, ou seja, a mensagem “Número telefônico Armazenado” é exibida. Para esse segundo cenário, temos o seguinte propósito: “*;A mensagem contém um número telefônico*;O

Passos	Resultados Esperados
Abrir a "Caixa de Entrada"	A lista de mensagens é mostrada
Selecionar uma mensagem	A mensagem é destacada
Selecionar a opção "Ler"	A mensagem de texto é mostrada. A lista de opções é mostrada
Selecionar a opção "Armazenar"	A tela "Criar Contato" é mostrada
Preencher todos os campos obrigatórios na tela "Criar Contato"	Todos os campos obrigatórios são preenchidos
Selecionar a opção "Pronto"	O aviso "Número telefônico Armazenado" é mostrado

Figura 5.7: Caso de Teste 01 - Obtido a partir do Modelo da Figura 5.6

Passos	Resultados Esperados
Abrir a "Caixa de Entrada"	A lista de mensagens é mostrada
Selecionar uma mensagem	A mensagem é destacada
Selecionar a opção "Ler"	A mensagem de texto é mostrada. A lista de opções é mostrada
Selecionar a opção "Armazenar"	A lista dos números telefônicos é mostrada
Escolher um número	O número telefônico é destacado
Selecionar a opção "Armazenar"	A tela "Criar Contato" é mostrada
Preencher todos os campos obrigatórios na tela "Criar Contato"	Todos os campos obrigatórios são preenchidos
Selecionar a opção "Pronto"	O aviso "Número telefônico Armazenado" é mostrado

Figura 5.8: Caso de Teste 02 - Obtido a partir do Modelo da Figura 5.6

aviso "Número telefônico Armazenado" é mostrado; *; Aceitar". A Figura 5.9 apresenta o modelo obtido a partir da aplicação do propósito definido acima e o único caso de teste gerado é o mesmo da Figura 5.7.

5.3.2 Seleção de Casos de Teste de Interação Dinâmica

A seleção de casos de teste de interação dinâmica também depende especificamente do propósito utilizado. Considerando o contexto do teste de interação dinâmica em aplicações

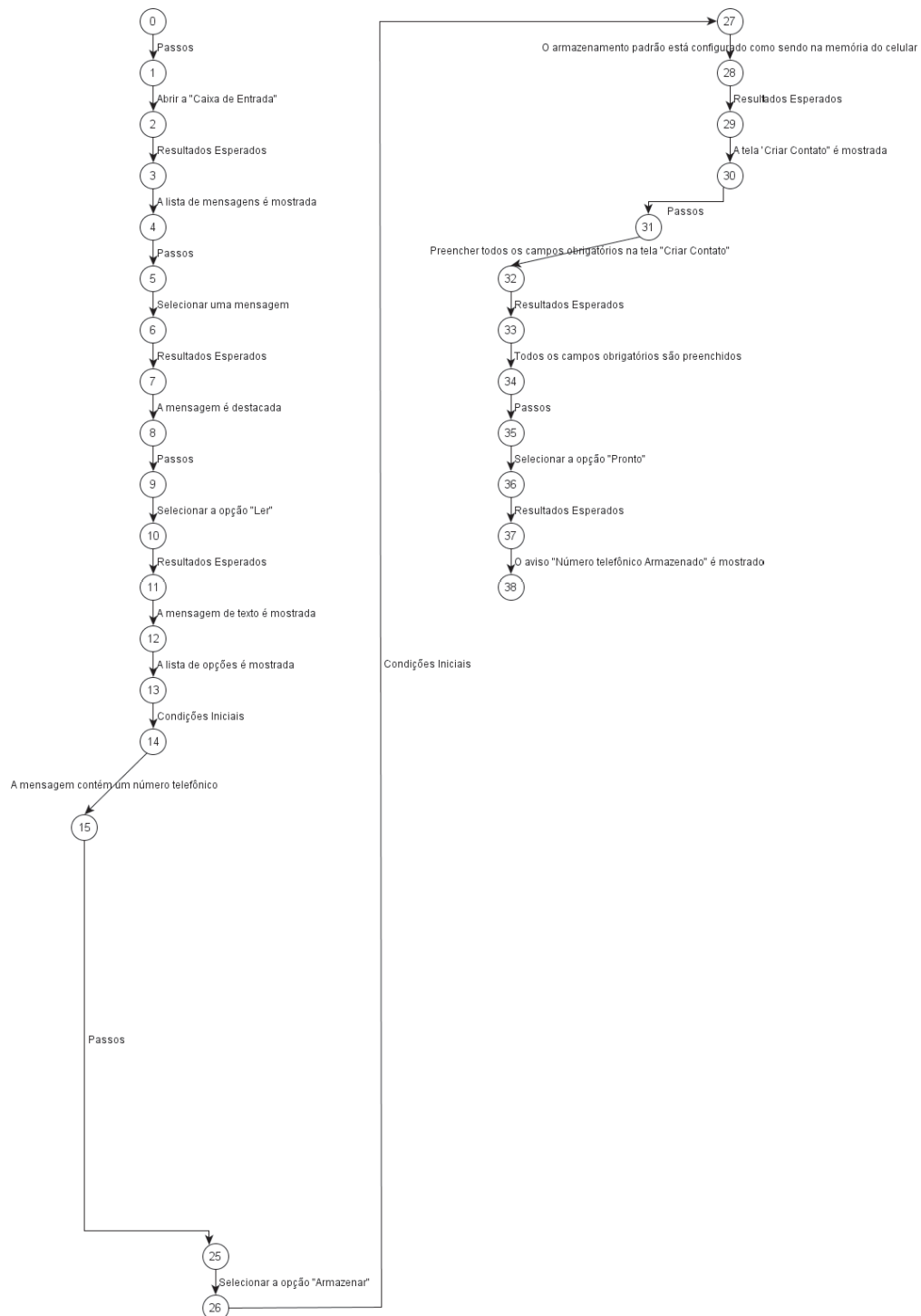


Figura 5.9: Modelo LTS Obtido com o Propósito: “*;A mensagem contém um número telefônico;*;O aviso "Número telefônico Armazenado"é mostrado;*;Aceitar”

para celulares cada caso de teste deve conter somente uma interrupção (Capítulo 6). Por isso, casos de teste de interação dinâmica só serão gerados se o nome da interrupção estiver no

propósito, e mais, cada propósito de teste deve conter um único nome de interrupção.

Se porventura um propósito de teste não possuir nenhuma interrupção, somente casos de teste de *feature* serão gerados, isto é, toda a porção do modelo referente ao comportamento das interrupções será eliminada durante a etapa de limitação do modelo.

O modelo obtido na Seção 4.3 (Figura 4.8) será usado para demonstrar como definir propósitos de teste para verificar as interações dinâmicas presentes no modelo LTS. Duas *features* foram usadas na geração do modelo: Mensagens Prediletas e Chegada de um Aviso.

Para o primeiro exemplo vamos definir um propósito visando testar o cenário onde, ao entrar na central de mensagens e visualizar a opção de acessar a pasta “Mensagens Prediletas”, um aviso chega ao celular. Esse cenário pode ser especificado através do seguinte propósito: “*;A pasta "Mensagens Prediletas" é exibida;Chegada de um aviso;*;Aceitar”. Note que o nome da interrupção precisa vir imediatamente depois do rótulo onde ela deve ocorrer. A Figura 5.10 apresenta o modelo obtido a partir da aplicação do propósito definido e as Figuras 5.11 e 5.12 apresentam os casos de teste obtidos. Note que nos dois casos de teste gerados, a interrupção ocorre após a visualização da pasta “Mensagens Prediletas”, como foi especificado no propósito. Além disso, os dois cenários possíveis da *feature* principal são cobertos. No caso de teste da Figura 5.11, uma interrupção ocorre no cenário onde a mensagem é movida, enquanto que, no caso de teste da Figura 5.12, uma interrupção ocorre no cenário onde a mensagem não é movida por causa da insuficiência de memória.

Se o propósito do primeiro exemplo, equivocadamente, tivesse sido especificado com um asterisco entre o rótulo escolhido e o nome da interrupção, então o significado do propósito seria o seguinte: um aviso deve chegar ao celular em qualquer passo, após o usuário entrar na central de mensagens e visualizar a opção de acessar a pasta “Mensagens Prediletas”. Daí, o ponto onde a interrupção pode ocorrer seria escolhido aleatoriamente, sendo possível também a geração de casos de teste com mais de uma interrupção.

Sabemos que uma interrupção só pode ser verificada após uma resposta do sistema, mas o testador pode especificar um propósito usando tanto a informação da ação do usuário quanto a resposta do sistema. Seja o cenário onde, quando o usuário está movendo o cursor em direção a seleção de uma mensagem, ocorre uma interrupção de chegada de uma aviso. Para esse cenário, o seguinte propósito também é válido: “*;Ir até uma mensagem;Chegada de um Aviso;*;Aceitar”. Mesmo definindo o propósito dessa forma, a interrupção só poderá

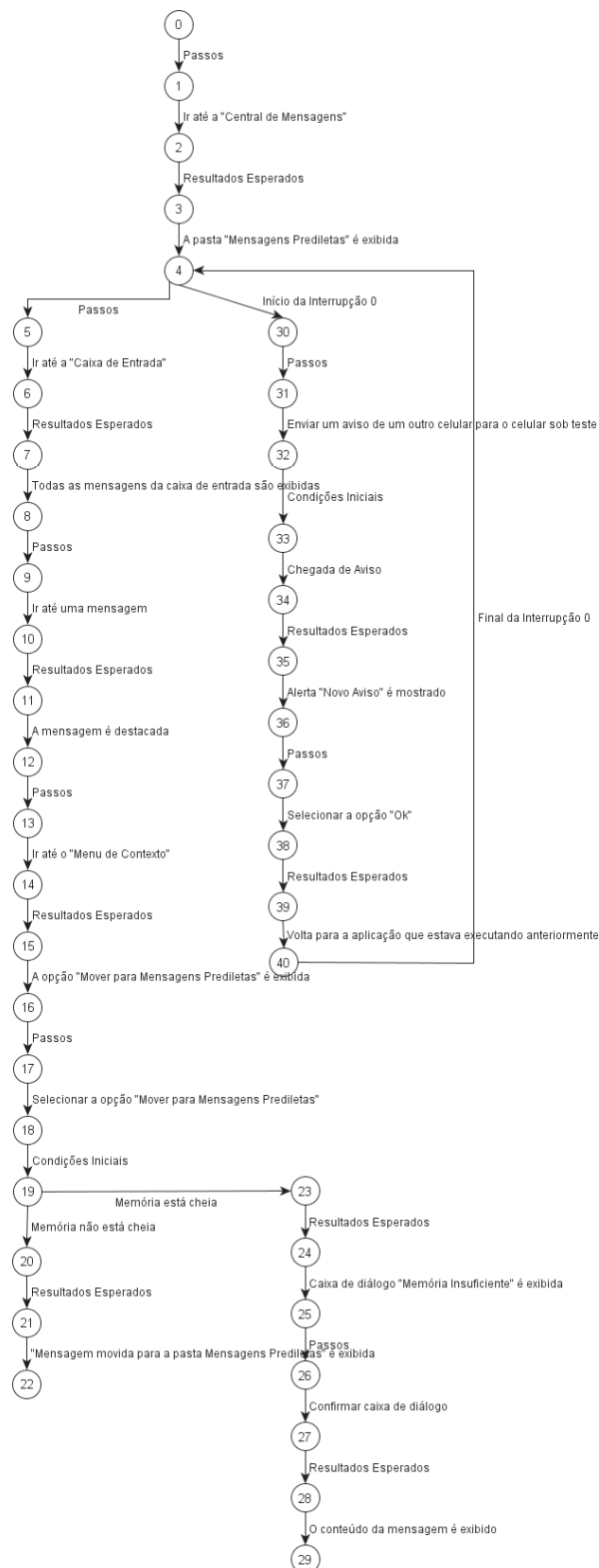


Figura 5.10: Modelo LTS Obtido com o Propósito: “*;A pasta "Mensagens Prediletas"é exibida;Chegada de um aviso;*;Aceitar”

Condição Inicial	A memória não está cheia
Passos	Resultados Esperados
Ir até a Central de Mensagens	A pasta "Mensagens Prediletas" é exibida
Enviar um aviso de um outro celular para o celular sob teste	O aviso é exibido numa caixa de diálogo
Selecionar a opção "Ok"	Volta para a aplicação que estava executando anteriormente
Ir até a "Caixa de Entrada"	Todas as mensagens da "Caixa de Entrada" são exibidas
Ir até uma mensagem	A mensagem é destacada
Ir até o "Menu de Contexto"	A opção "Mover para Mensagens Prediletas" é exibida
Selecionar a opção "Mover para Mensagens Prediletas"	"Mensagem movida para a pasta Mensagens Prediletas" é exibida

Figura 5.11: Caso de Teste 01 - Obtido a partir do Modelo da Figura 5.10

Condição Inicial	A memória está cheia
Passos	Resultados Esperados
Ir até a Central de Mensagens	A pasta "Mensagens Prediletas" é exibida
Enviar um aviso de um outro celular para o celular sob teste	O aviso é exibido numa caixa de diálogo
Selecionar a opção "Ok"	Volta para a aplicação que estava executando anteriormente
Ir até a "Caixa de Entrada"	Todas as mensagens da "Caixa de Entrada" são exibidas
Ir até uma mensagem	A mensagem é destacada
Ir até o "Menu de Contexto"	A opção "Mover para Mensagens Prediletas" é exibida
Selecionar a opção "Mover para Mensagens Prediletas"	Caixa de diálogo "Memória Insuficiente" é exibida
Confirmar a caixa de diálogo de informação da memória	O conteúdo da mensagem é exibido

Figura 5.12: Caso de Teste 02 - Obtido a partir do Modelo da Figura 5.10

ser verificada após a resposta do sistema associada a tal ação do usuário. A Figura 5.13 apresenta o modelo gerado e os casos de teste obtidos são apresentados nas Figuras 5.14 e 5.15.



Figura 5.13: Modelo LTS Obtido com o Propósito: “*;Ir até uma mensagem;Chegada de um Aviso*;Aceitar”

Passos	Resultados Esperados
Ir até a Central de Mensagens	A pasta "Mensagens Prediletas" é exibida
Ir até a "Caixa de Entrada"	Todas as mensagens da "Caixa de Entrada" são exibidas
Ir até uma mensagem	A mensagem é destacada
Enviar um aviso de um outro celular para o celular sob teste	O aviso é exibido numa caixa de diálogo
Selecionar a opção "Ok"	Volta para a aplicação que estava executando anteriormente
Ir até o "Menu de Contexto"	A opção "Mover para Mensagens Prediletas" é exibida
Selecionar a opção "Mover para Mensagens Prediletas"	"Mensagem movida para a pasta Mensagens Prediletas" é exibida

Figura 5.14: Caso de Teste 01 - Obtido a partir do Modelo da Figura 5.13

Passos	Resultados Esperados
Ir até a Central de Mensagens	A pasta "Mensagens Prediletas" é exibida
Ir até a "Caixa de Entrada"	Todas as mensagens da "Caixa de Entrada" são exibidas
Ir até uma mensagem	A mensagem é destacada
Enviar um aviso de um outro celular para o celular sob teste	O aviso é exibido numa caixa de diálogo
Selecionar a opção "Ok"	Volta para a aplicação que estava executando anteriormente
Ir até o "Menu de Contexto"	A opção "Mover para Mensagens Prediletas" é exibida
Selecionar a opção "Mover para Mensagens Prediletas"	Caixa de diálogo "Memória Insuficiente" é exibida
Confirmar a caixa de diálogo de informação da memória	O conteúdo da mensagem é exibido

Figura 5.15: Caso de Teste 02 - Obtido a partir do Modelo da Figura 5.13

5.4 Considerações Finais

Este capítulo apresentou o algoritmo usado para a geração de casos de teste, que foi implementado na ferramenta LTS-BT. O algoritmo apresentado gera tanto casos de teste de interação estática quanto casos de teste de interação dinâmica, ou seja, é um algoritmo só, com a possibilidade de aplicação nos dois tipos de teste de interação.

Outra contribuição apresentada está relacionada a seleção de casos de teste. Foi apresentado o procedimento para a definição de propósitos de teste com o objetivo de verificar as interações entre as *features*. Dessa forma, o tamanho da suíte de teste gerada poderá ser reduzida e focar precisamente numa funcionalidade ou interação específica, fazendo com que a estratégia apresentada neste trabalho seja de grande utilidade no contexto do teste de interação em aplicações para celulares.

Capítulo 6

Estudo de Caso

O objetivo deste capítulo é apresentar o estudo de caso realizado com a finalidade de avaliar a aplicação da abordagem proposta neste trabalho. Esse estudo de caso foi realizado com a cooperação de um time de teste de interação real, que faz parte do Motorola BTC, e as *features* foram escolhidas baseando-se nas *features* que o referido time de teste está trabalhando atualmente.

O estudo de caso está dividido em três etapas:

1. Especificar as *features* escolhidas usando os *templates* apresentados no Capítulo 3 e avaliar o custo associado a inserção dos comportamentos das interações no modelo LTS;
2. Avaliar a abordagem proposta para a geração e seleção de casos de teste de interação estática;
3. Avaliar a estratégia proposta para a geração e seleção de casos de teste com interrupções.

Por fim, serão apresentadas algumas considerações finais sobre os resultados obtidos.

6.1 Especificação das *Features* em Casos de Uso

As *features* escolhidas foram Compositor de Mensagens, Contatos e Chegada de uma Mensagem. A primeira *feature* especifica o comportamento de um editor que permite a criação de

mensagens de texto (SMS) e mensagens multimídia (MMS). O editor de mensagens interage estaticamente com a *feature* Contatos desde que um usuário pode escolher um contato da lista de contatos para enviar a mensagem. A última *feature* escolhida é a *feature* responsável pelas interrupções que ocorrem quando uma nova mensagem é recebida.

O estudo de caso foi focado no cenário onde o usuário cria uma nova mensagem a partir de uma opção de criação de novas mensagens pertencente ao menu “Central de Mensagens”. Como foi mencionado anteriormente, esse cenário é coberto por duas *features*. As Figuras 6.1 e 6.2 apresentam a primeira parte do cenário, que é especificada pela *feature* Compositor de Mensagens, e a Figura 6.3 mostra a segunda parte do cenário descrito, relativa a *feature* Contatos. A especificação da interrupção de chegada de uma mensagem já foi apresentada na Figura 3.8.

Cenário Principal

Descrição: O usuário cria uma nova mensagem de texto a partir da central de mensagens.

DE: INICIO

Para: CONTATOS#UC_02#1M

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1M	Ir até a Central de Mensagens		A opção "Criar Mensagem" é exibida
2M	Selecionar a opção "Criar Mensagem"		A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
3M	Digitar pelo menos um caractere na mensagem		O Contador de Caracteres é atualizado. A opção direita é "Enviar".
4M	Selecionar a opção "Enviar"		

Figura 6.1: *Feature* Compositor de Mensagens - Cenário Principal

O objetivo da primeira análise realizada é avaliar o custo da inserção do comportamento das interações no modelo LTS. Para isso, algumas métricas foram coletadas durante a etapa de especificação das *features*:

- **Entendimento da *feature*:** é o tempo gasto com a leitura dos documentos que descrevem o comportamento de uma determinada *feature*;
- **Planejamento:** é o tempo gasto no planejamento de como será organizado o caso de

Cenários Alternativos

Descrição: O usuário não digita nenhum caractere na mensagem.

De: 3M

Para: 2M

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1A	Selecionar a opção "Cancelar"		A Central de Mensagens é exibida

Descrição: O tipo da mensagem é alterado para MMS através das opções de envio.

De: 4M

Para: 4M

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
2A	Selecionar a opção "Opções"		A opção "Opções de Envio" é exibida
3A	Selecionar a opção "Opções de Envio"		A tela "Opções de Mensagens SMS" é exibida
4A	Preencher qualquer campo suportado somente por mensagens MMS		Qualquer campo suportado somente por mensagens MMS é preenchido
5A	Selecionar a opção "Pronto"		Caixa de diálogo para confirmação é exibida
6A	Selecionar a opção "Sim"		A aplicação "Compositor de Mensagens" é exibida. O valor do Contador de Caracteres é atualizado. A palavra "MMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Enviar"

Descrição: O usuário cancela a operação de alteração do tipo da mensagem.

De: 5A

Para: 4M

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
7A	Selecionar a opção "Cancelar"		A aplicação "Compositor de Mensagens" é exibida com a última mensagem

Descrição: O usuário não permite a alteração do tipo da mensagem.

De: 6A

Para: 4A

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
8A	Selecionar a opção "Não"		A tela "Opções de Mensagens SMS" é exibida

Figura 6.2: *Feature* Compositor de Mensagens - Cenários Alternativos

uso a ser especificado;

- **Criação do caso de uso:** é o tempo gasto no trabalho de preenchimento do *template*;
- **Revisão:** é o tempo gasto na revisão do documento gerado, contendo a especificação do caso de uso da *feature*.

As Figuras 6.4, 6.5 e 6.6 apresentam as métricas obtidas durante a especificação das *fea-*

Cenário Principal

Descrição: A mensagem é enviada.
 De: COMPOSITOR_MENSAGENS#UC_01#4M
 Para: FINAL

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1M			A tela "Enviar Mensagem" é exibida com os contatos da lista de contatos
2M	Escolher pelo menos um contato		Pelo menos um contato é escolhido
3M	Selecionar a opção "Enviar"		Aviso "Mensagem Enviada" é exibido

Cenário Alternativo

Descrição: O usuário cancela o envio da mensagem.
 De: 3M
 Para: COMPOSITOR_MENSAGENS#UC_01#4M

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1A	Selecionar a opção "Opções"		A tela "Opções" é exibida
2A	Selecionar a opção "Cancelar"		A aplicação "Compositor de Mensagens" é exibida com a última mensagem

Figura 6.3: *Feature* Contatos

tures Compositor de Mensagens, Contatos e Chegada de uma Mensagem, respectivamente. Observando as métricas obtidas, podemos concluir que o tempo de especificação total das *features* principais, como Compositor de Mensagens e Contatos, é relativamente igual ao tempo de especificação da interrupção. Na prática, o comportamento de uma interrupção tende a ser bem pequeno, de rápida especificação e dado que, os modelos LTS são gerados automaticamente, o único custo associado a inserção de interrupções no modelo está relacionado a própria especificação das interrupções, usando os *templates*.

O uso dos *templates* apresentados no Capítulo 3 tem muitas vantagens. A principal vantagem está relacionada a geração e seleção automática de casos de teste (Capítulo 5). Uma outra vantagem identificada foi que a padronização da documentação das *features* facilita a comunicação entre as pessoas da equipe.

6.2 Geração de Casos de Teste de Interação Estática

A segunda etapa do estudo de caso tem o objetivo de avaliar o procedimento usado na obtenção de casos de teste de interação estática. O modelo LTS (Figura 6.7) obtido a par-

Atividade	Total
Entendimento da <i>Feature</i>	02:00
Planejamento	00:20
Criação do Caso de Uso	00:40
Revisão	00:30
Total	03:30

Figura 6.4: Métricas de Especificação da *Feature* Compositor de Mensagens

Atividade	Total
Entendimento da <i>Feature</i>	00:20
Planejamento	00:05
Criação do Caso de Uso	00:15
Revisão	00:05
Total	00:45

Figura 6.5: Métricas de Especificação da *Feature* Contatos

Atividade	Total
Entendimento da <i>Feature</i>	00:20
Planejamento	00:05
Criação do Caso de Uso	00:25
Revisão	00:10
Total	01:00

Figura 6.6: Métricas de Especificação da *Feature* Chegada de uma Mensagem

tir das especificações das *features* Compositor de Mensagens (Figuras 6.1 e 6.2) e Contatos (Figura 6.3) será usado para a geração de casos de teste de interação estática. Na Figura 6.7, os vértices numerados de 18 a 34 correspondem a *feature* Contatos e os vértices restantes, a *feature* Compositor de Mensagens.

A interação entre as duas *features* está presente em dois pontos no modelo LTS gerado. A primeira interação (Figura 6.7, vértices 16 a 20) é descrita pelo passo onde o usuário seleciona a opção “Armazenar”, da *feature* Compositor de Mensagens, e o sistema exibe como resposta a tela com a lista de contatos, da *feature* Contatos. O outro cenário onde a segunda interação (Figura 6.7, vértices 31 a 34 e o vértice 16) ocorre é quando o usuário decide cancelar o envio da mensagem, operação referente a *feature* Contatos, e o fluxo de execução da funcionalidade volta para a edição da mensagem, referente a *feature* Compositor de Mensagens.

Realizando uma geração exaustiva de casos de teste, usando o algoritmo apresentado na

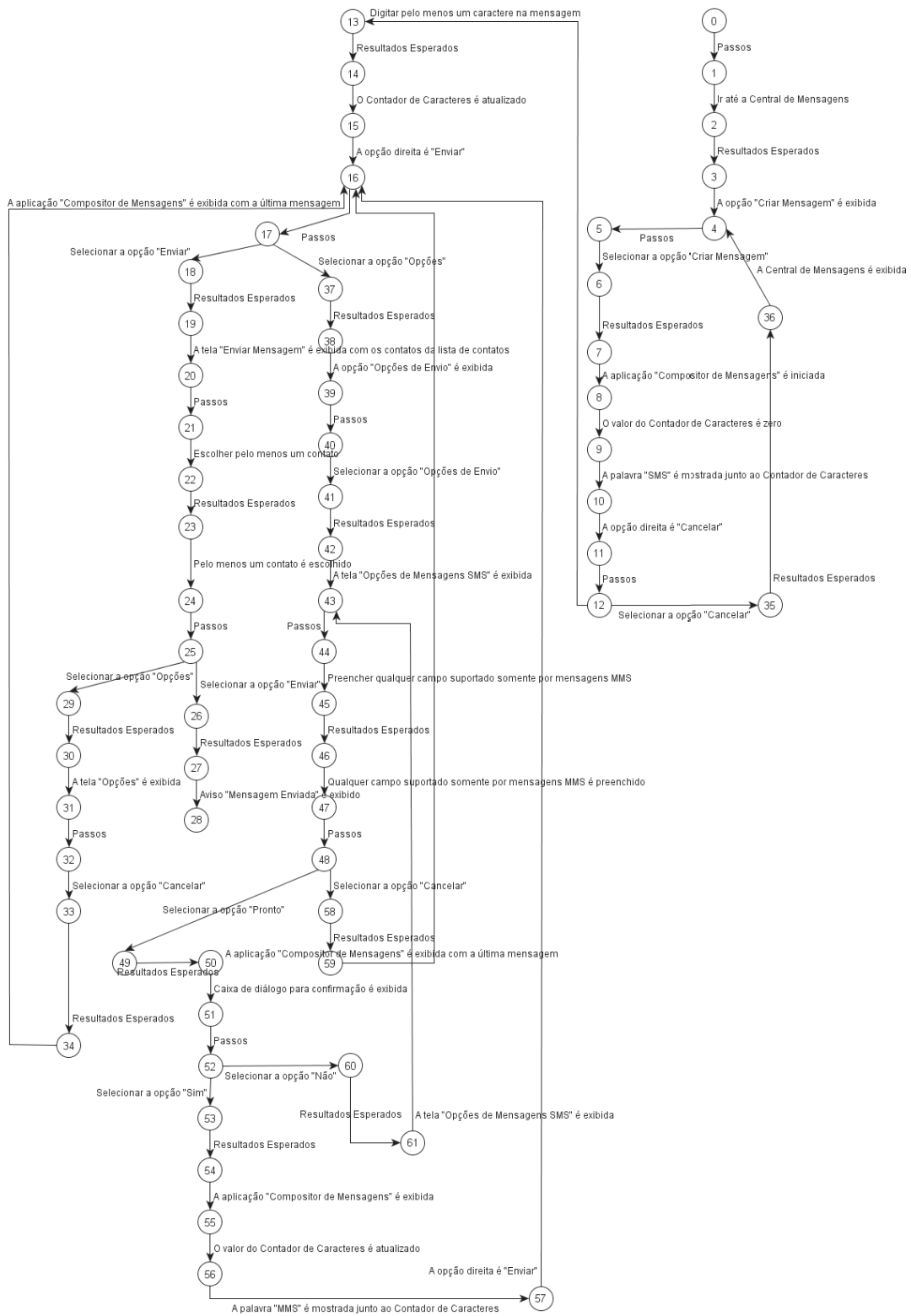


Figura 6.7: Modelo LTS Gerado

Seção 5.2, os casos de teste das Figuras A.10 a A.15 (Apêndice A) foram obtidos. Como podemos observar, somente dois casos de teste estão relacionados ao teste de interação estática (Figuras A.10 e A.11), já o restante, verifica somente o comportamento da *feature* Compositor de Mensagens.

Nesse caso, a estratégia de seleção de casos de teste, apresentada na Seção 5.3.1, deve ser utilizada para a obtenção dos casos de teste com um objetivo específico, isto é, com o objetivo de testar somente as interações estáticas. Portanto, propósitos de teste serão definidos para testar os seguintes cenários onde ocorrem tais interações:

1. O usuário envia uma mensagem para um contato da lista de contatos;
2. O usuário cancela o envio da mensagem.

Para o primeiro cenário a ser verificado, o seguinte propósito pode ser definido: “*;Aviso "Mensagem Enviada"é exibido*;Aceitar”. Utilizando esse propósito, somente um caso de teste é obtido (o da Figura A.10). Como LTS-BT permite a visualização gráfica dos modelos LTS e também a definição de propósitos através da seleção de rótulos no próprio modelo, a tarefa de seleção de casos de teste de interação fica extremamente fácil.

Para o segundo cenário, foi definido o propósito de teste a seguir: “*;A tela "Opções"é exibida*;Aceitar”. Com esse propósito, somente o caso de teste da Figura A.11 foi gerado.

Como foi possível observar, a estratégia apresentada permite que os integrantes do time de teste de interação realizem testes de interação estática de maneira altamente focada em seus objetivos, ou seja, sem a necessidade de geração de testes desnecessários, implicando no aumento de produtividade. Além disso, através de uma comparação realizada com os casos de teste já existentes, gerados manualmente pelo time de teste de interação, foi constatado que os casos de teste gerados automaticamente, através da estratégia apresentada neste trabalho, cobrem todos os casos de teste existentes. E mais, foi identificado também que não havia nenhum caso de teste verificando o segundo cenário, ou seja, o cenário onde o usuário cancela o envio da mensagem e volta ao compositor de mensagens. Portanto, a estratégia apresentada foi mais eficiente, no que diz respeito à cobertura das interações estáticas, do que a geração manual de casos de teste.

6.3 Geração de Casos de Teste de Interação Dinâmica

A terceira etapa do estudo de caso almeja avaliar o procedimento usado na obtenção de casos de teste de interação dinâmica. O mesmo modelo LTS apresentado na Figura 6.7 será utilizado, agora considerando que as interrupções de chegada de uma nova mensagem podem ocorrer. A Figura 6.8 mostra o modelo LTS da interrupção de chegada de uma mensagem, cuja especificação foi apresentada na Figura 3.8.

Para essa terceira etapa, também vamos considerar que o comportamento da interrupção de chegada de uma nova mensagem será redefinido em alguns pontos da *feature* Compositor de Mensagens. A especificação do comportamento alternativo para essa interrupção pode ser vista na Figura 6.9, onde, ao chegar uma nova mensagem, apenas um aviso temporário irá aparecer indicando que uma nova mensagem chegou. Esse comportamento alternativo é válido para os passos “2M” e “3M” da Figura 6.1 e o passo “6A” da Figura 6.2. Já a Figura 6.10, mostra o modelo LTS que representa o comportamento alternativo para a *feature* Chegada de uma Mensagem.

Devido a impossibilidade de visualização do modelo LTS obtido com todas as interrupções possíveis, ele não será apresentado. Mas, vamos considerar que o comportamento padrão da interrupção Chegada de uma Mensagem pode ocorrer a partir dos seguintes vértices do modelo LTS da Figura 6.7: 4, 20, 24, 28, 31, 39, 43, 47 e 51. Já o comportamento alternativo da interrupção, pode ocorrer a partir dos vértices: 11 e 16 (também da Figura 6.7).

Utilizando o modelo LTS com as interrupções especificadas e realizando uma geração exaustiva de casos de teste, usando o algoritmo apresentado na Seção 5.2, 6498 casos de teste foram obtidos. Dentre todos os casos de teste gerados, 6492 verificam cenários onde ocorrem interrupções. Dos 6492 casos de teste com interrupções, apenas 64 contém somente uma interrupção. Os únicos casos de teste com interrupções que realmente interessam ao time de teste de interação são aqueles onde há somente uma interrupção. Isso acontece por dois motivos: (1) é considerada a suposição apresentada em [Jor01] de que falhas raramente ocorrem como resultado da ocorrência simultânea de duas ou mais faltas; (2) bons casos de teste devem facilitar a localização de faltas [BFT06].

Por isso, podemos concluir que a geração exaustiva é inadequada ao contexto do teste de

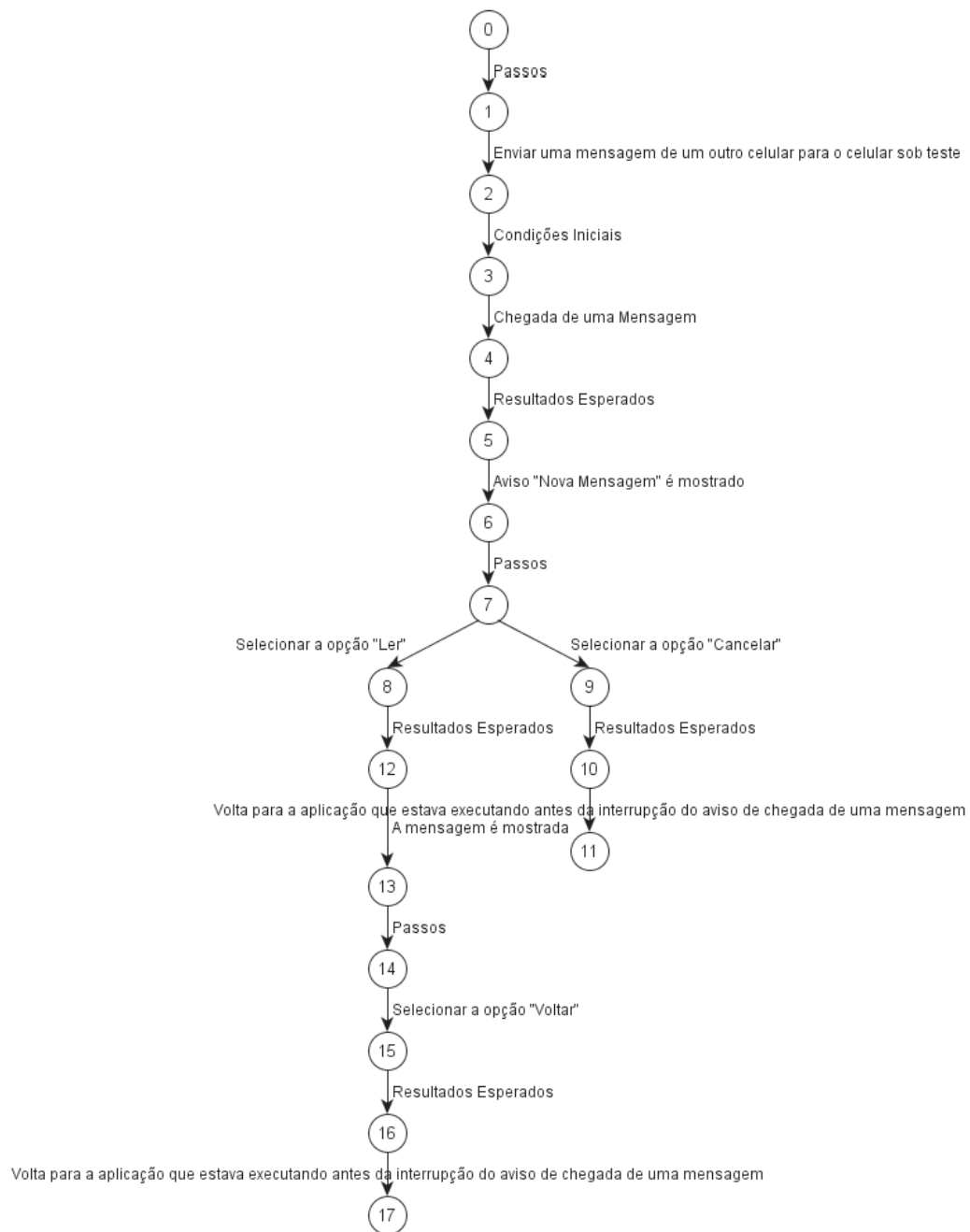


Figura 6.8: Modelo LTS da Interrupção Causada pela Chegada de uma Mensagem

interações em aplicações para celulares. Então, o procedimento para a obtenção de casos de teste de interação de forma eficiente deverá ser realizado através da utilização de propósitos de teste. Como a identificação de importantes cenários com interrupções a serem verificados depende da experiência do testador, alguns cenários foram indicados pelo time de teste de interação. Portanto, propósitos de teste serão definidos para testar os seguintes cenários

Interrupções

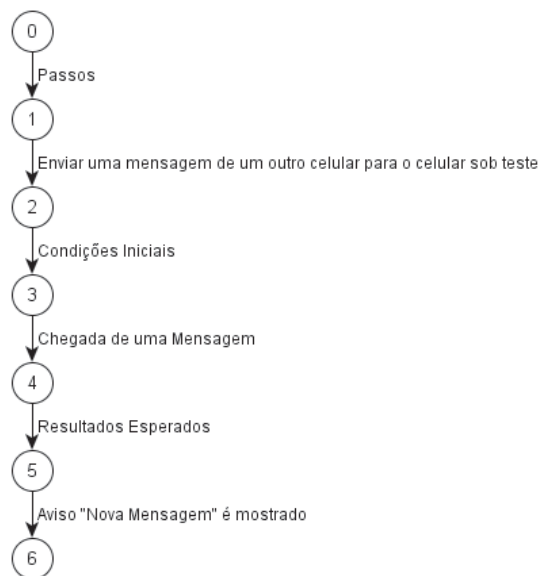
De: INICIO

Para: FINAL

Passos: 2M, 3M, 6A

Interrupção: Chegada de uma Mensagem

Id	Ação do Usuário	Estado do Sistema	Resposta do Sistema
1I	Enviar uma mensagem de um outro celular para o celular sob teste		Aviso "Nova Mensagem" é mostrado

Figura 6.9: Redefinição do Comportamento da *Feature* Chegada de uma MensagemFigura 6.10: Modelo LTS do Segundo Comportamento da *Feature* Chegada de uma Mensagem

indicados:

1. O usuário abre o editor de mensagens e, antes de começar a digitar a mensagem, uma nova mensagem chega ao celular;
2. O usuário começa a digitar uma mensagem e uma nova mensagem é recebida;
3. O usuário está escolhendo os contatos para enviar a mensagem digitada e uma nova mensagem chega;
4. O usuário está alterando o tipo da mensagem digitada e uma nova mensagem é recebida.

Casos de teste podem ser gerados para verificar o primeiro cenário através do seguinte propósito: “*;A opção direita é "Cancelar";Chegada de uma Mensagem;*;Aceitar”. Usando esse propósito, os casos de teste das Figuras A.16 a A.21 (Apêndice A) foram gerados. Note que cada caso de teste possui somente uma única interrupção e essa interrupção obedece ao comportamento redefinido na tabela *Interrupções* para o passo que representa o cenário em questão, o passo “2M” (Figura 6.9).

O propósito usado para verificar o segundo cenário é o seguinte: “*;Digitar pelo menos um caractere na mensagem;Chegada de uma Mensagem;*;Aceitar”. As Figuras A.22 a A.26 (Apêndice A) mostram os casos de teste obtidos através de tal propósito. Note que os casos de teste desse propósito também obedecem ao comportamento redefinido na tabela *Interrupções* para o passo referente ao cenário sendo verificado, o passo “3M” (Figura 6.9).

O terceiro cenário pode ser verificado através do propósito: “*;Pelo menos um contato é escolhido;Chegada de uma Mensagem;*;Aceitar”. As Figuras A.27 a A.30 (Apêndice A) apresentam os casos de teste gerados para esse cenário. Note que, como essa funcionalidade de escolher os contatos para enviar a mensagem pertence a *feature* Contatos, os casos de teste gerados verificam tanto a interação estática quanto a interação dinâmica. Observe também que o comportamento da interrupção para esse cenário é o comportamento padrão, visto que, a tabela *Interrupções* (Figura 6.9) é relativa somente a *feature* Compositor de Mensagens.

Para o quarto e último cenário, o seguinte propósito foi definido: “*;Qualquer campo suportado somente por mensagens MMS é preenchido;Chegada de uma Mensagem;*;Aceitar”. Os casos de teste obtidos para verificar esse cenário são os das Figuras A.31 a A.36 (Apêndice A).

Através dos exemplos utilizados, podemos ver que a geração exhaustiva de casos de teste de interação dinâmica é inviável, tornando a utilização de propósitos de teste imprescindível. Com relação a comparação com os casos de teste existentes, obtidos manualmente, foi constatado que os casos de teste obtidos através da estratégia apresentada cobrem todos os casos de teste existentes. Além disso, foi identificado que apenas o segundo cenário estava sendo verificado pelos casos de teste existentes, ou seja, a estratégia apresentada neste trabalho também foi mais eficiente no teste de interações dinâmicas.

Dada a facilidade e a velocidade com que os propósitos de teste podem ser definidos, a estratégia apresentada neste trabalho é de grande importância para o teste de interação

em aplicações para celulares, pois permite o foco em interações específicas. Isto facilita o processo de integração e evolução das aplicações: adição de novas interrupções ou *features* e modificação de comportamentos específicos de interrupções. Bem como reduz a suíte de teste, deixando a exaustão das funcionalidades para o teste de *feature*.

6.4 Considerações Finais

Um estudo de caso foi apresentado nesse capítulo com o objetivo de avaliar a estratégia de especificação de interações entre *features*, geração e seleção de casos de teste de interação. O estudo foi realizado com a colaboração de um time de teste de interação real, o que permitiu a avaliação da abordagem proposta na prática.

Com base na observação deste estudo de caso, temos que a introdução dos comportamentos das interações só acrescenta o custo da especificação das *features*, o que já é uma prática constante dentro da Motorola. Com relação a geração e seleção de casos de teste, vimos a importância da utilização de propósitos de teste, o que implica em uma maior efetividade na geração, visto que, casos de teste podem ser obtidos com o foco num objetivo específico. Já a definição de propósitos de teste não acrescentou nenhum custo a seleção, dado que os testadores podem definir propósitos através de uma interface intuitiva, sem a necessidade de manusear modelos LTS. É importante ressaltar que o estudo de caso foi completamente executado pelo pesquisador, porém dentro do ambiente de trabalho da equipe de teste de interação e em contato direto com os mesmos.

Embora a estratégia tenha sido aplicada em um estudo de caso simples, as atividades de geração e seleção foram facilitadas pelo uso da ferramenta estendida (LTS-BT), tornando as atividades realizadas mais simples e menos sujeita à inserção de erros, devido a automação. Além disso, os casos de teste gerados atendem as expectativas da equipe no tocante ao formato, foco e cobertura.

Capítulo 7

Conclusão

O trabalho apresentado neste documento propõe uma estratégia para dar suporte ao teste de interação entre *features* em aplicações para celulares. Casos de uso de *features* e suas interações são especificados através de um *template* proposto. Os casos de uso são transformados em um modelo LTS a partir do qual casos de teste podem ser obtidos automaticamente. Após a geração dos casos de teste, técnicas de seleção são aplicadas para reduzir o tamanho da suíte de teste visando testar funcionalidades específicas.

Como suporte à estratégia, a ferramenta LTS-BT foi estendida. O modelo LTS utilizado pela ferramenta foi estendido para capturar os comportamentos das interações. O algoritmo de geração de casos de teste foi modificado para gerar casos de teste de interação e uma estratégia para a seleção de casos de teste de interação baseada em propósitos de teste foi apresentada.

Um estudo de caso foi realizado com o intuito de avaliar o funcionamento da estratégia proposta. O estudo de caso considerou desde a etapa de especificação das *features* e suas interações usando os templates até a geração e seleção de casos de teste de interação. Foi verificada a importância da aplicação de uma estratégia de seleção de casos de teste no contexto do teste de interação, o que permite o foco do teste em interações específicas. Como resultados obtidos, temos que o único custo associado a geração de casos de teste de interação está relacionado a especificação das próprias interações usando o *template* proposto e que os casos de teste obtidos realmente atendem as expectativas da equipe de teste de interação.

7.1 Contribuições

Considerando que existem poucos procedimentos sistemáticos para a geração de casos de teste funcional para aplicações de celulares e que o teste de interação é muito pouco explorado, especialmente no contexto do desenvolvimento baseado em *features*, a principal contribuição deste trabalho caracteriza-se pela definição de uma estratégia para geração de casos de teste de interação no referido contexto.

Para a geração de casos de teste usando propósitos de teste, ao contrário de outras ferramentas (e.g. TGV) onde o testador precisa manipular modelos formais, através da utilização da estratégia proposta neste trabalho, o testador só precisa especificar as *features* e suas interações usando os *templates* apresentados. Em seguida, os propósitos de teste podem ser definidos através de uma notação bem simples ou através da interface da ferramenta LTS-BT, que possibilita a visualização gráfica do modelo LTS e a definição de propósitos através da seleção de rótulos no próprio modelo. Dessa forma, o testador não precisa construir modelos formais para definir propósitos de teste, facilitando a implantação da estratégia proposta.

O procedimento apresentado neste trabalho não adiciona custos extras ao processo de teste, visto que, somente artefatos existentes são utilizados. Muito pelo contrário, custos podem ser reduzidos dado que, quanto mais cedo um erro for encontrado, menos onerosa será a correção. Além disso, este trabalho pode contribuir para a melhoria da qualidade dos produtos vendidos pelas empresas de telecomunicações aos usuários finais, visto que, novos cenários de teste (cenários que causam interações entre as *features*) podem ser incluídos no conjunto de casos de teste.

7.2 Trabalhos Futuros

Com a conclusão deste trabalho, surgem várias oportunidades que o complementam. A seguir, temos a descrição de alguns desses trabalhos propostos:

Extensão da ferramenta LTS-BT para lidar com quiescência: o tratamento da questão da quiescência é um dos principais pontos necessários a definição da consistência dos casos de teste gerados;

Investigação de estratégias de controle do ambiente de execução: a investigação de es-

estratégias de controle do ambiente de execução dos testes possibilitará a garantia de consistência das suítes de teste. Para isso, as suítes de teste precisam ser enriquecidas com informações/dependências de contexto;

Investigação de outras estratégias de seleção: essa investigação é necessária para permitir que o teste foque em interações específicas quando o modelo contiver um grande número de features, para, por exemplo, dar suporte a interação big-bang e ao teste de regressão;

Provisionamento de embasamento formal para a abordagem apresentada: a intenção é prover embasamento formal que permita a análise de propriedades de suítes de teste como consistência e completude. Principalmente pelo fato de estar lidando com modelos parciais;

Extensão da proposta para outros contextos: estender a estratégia proposta para dar suporte ao teste de interação em sistemas embarcados em geral e sistemas de tempo real;

Realização de outros estudos de caso: o estudo de caso realizado permitiu uma visão prática da estratégia proposta. A realização de outros estudos de casos e estudos experimentais precisam ser realizados para tentar mensurar o retorno obtido com a adoção da estratégia apresentada.

Bibliografia

- [BC05] Renée C. Bryce and Charles J. Colbourn. Test prioritization for pairwise interaction coverage. In *A-MOST '05: Proceedings of the first international workshop on Advances in model-based testing*, pages 1–7, New York, NY, USA, 2005. ACM Press.
- [Bei90] B. Beizer. *Software Testing Techniques*. Van Nostrand Reinhold, second edition, 1990.
- [Bei95] Boris Beizer. *Black-box testing: techniques for functional testing of software and systems*. John Wiley & Sons, Inc., New York, NY, USA, 1995.
- [BFT06] Benoit Baudry, Franck Fleurey, and Yves Le Traon. Improving test suites for efficient fault localization. In *ICSE '06: Proceeding of the 28th international conference on Software engineering*, pages 82–91, New York, NY, USA, 2006. ACM Press.
- [Bin99] Robert V. Binder. *Testing object-oriented systems: models, patterns, and tools*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
- [BS02] Kurt Bittner and Ian Spence. *Use Case Modeling*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1 edition, 2002.
- [CAN⁺07a] E. G. Cartaxo, W. L. Andrade, L. H. O. Nascimento, F. G. O. Neto, and P. D. L. Machado. Generating and selecting feature and feature interaction functional test cases for mobile phone applications. Submitted to International Symposium on Software Testing and Analysis - ISSTA'2007, 2007.

- [CAN⁺07b] E. G. Cartaxo, W. L. Andrade, L. H. O. Nascimento, F. G. O. Neto, and P. D. L. Machado. Lts-bt: A tool to generate and select functional test cases for mobile phone applications. Submitted to Automation of Software Testing - AST'2007, 2007.
- [Car06] Emanuela Gadelha Cartaxo. Geração de casos de teste funcional para aplicações de celulares. Master's thesis, Universidade Federal de Campina Grande, oct 2006.
- [CGMC03] Myra B. Cohen, Peter B. Gibbons, Warwick B. Mugridge, and Charles J. Colbourn. Constructing test suites for interaction testing. In *ICSE '03: Proceedings of the 25th International Conference on Software Engineering*, pages 38–48, Washington, DC, USA, 2003. IEEE Computer Society.
- [CKMRM03] Muffy Calder, Mario Kolberg, Evan H. Magill, and Stephan Reiff-Marganiec. Feature interaction: a critical review and considered forecast. *Comput. Networks*, 41(1):115–141, 2003.
- [Coc00] Alistair Cockburn. *Writing Effective Use Cases*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2000.
- [CS06] Gustavo Cabral and Augusto Sampaio. Formal specification generation from requirement documents. In *Brazilian Symposium on Formal Methods (SBMF)*, pages 217–232, Natal, Brazil, sep 2006.
- [dBORZ00] L. du Bousquet, F. Ouabdesselam, J.-L. Richier, and N. Zuanon. Feature interaction detection using a synchronous approach and testing. *Comput. Networks*, 32(4):419–431, 2000.
- [dFAM06] André L. L. de Figueiredo, Wilkerson L. Andrade, and Patrícia D. L. Machado. Generating interaction test cases for mobile phone systems from use case specifications. *SIGSOFT Softw. Eng. Notes*, 31(6):1–10, 2006.
- [dVT00] Rene G. de Vries and Jan Tretmans. On-the-fly conformance testing using SPIN. *International Journal on Software Tools for Technology Transfer*, 2(4):382–393, 2000.

- [EFW01] I. K. El-Far and J. A. Whittaker. Model-based software testing. *Encyclopedia on Software Engineering*, 2001.
- [FGMT02] Loe M. G. Feijs, Nicolae Goga, Sjouke Mauw, and Jan Tretmans. Test selection, trace distance and heuristics. In *TestCom '02: Proceedings of the IFIP 14th International Conference on Testing Communicating Systems XIV*, pages 267–282, Deventer, The Netherlands, The Netherlands, 2002. Kluwer, B.V.
- [FMP03] Jean-Claude Fernandez, Laurent Mounier, and Cyril Pachon. Property oriented test case generation. In *Proceedings of FATES'03 (Satellite workshop of ASE'03), Montreal, Canada*, 2003.
- [FN03] Amy P. Felty and Kedar S. Namjoshi. Feature specification and automated conflict detection. *ACM Trans. Softw. Eng. Methodol.*, 12(1):3–27, 2003.
- [JJ05] Claude Jard and Thierry Jéron. Tgv: theory, principles and algorithms: A tool for the automatic synthesis of conformance test cases for non-deterministic reactive systems. *Int. J. Softw. Tools Technol. Transf.*, 7(4):297–315, 2005.
- [Jor01] Paul Jorgensen. *Software Testing: A Craftman's Approach*. CRC Press, Inc., Boca Raton, FL, USA, 2001.
- [Kat99] Joost-Pieter Katoen. *Concepts, algorithms, and tools for model checking*. Alexander-Universitaet Erlangen-Nuernberg, 1999.
- [KK98] Dirk O. Keck and Paul J. Kuehn. The feature and service interaction problem in telecommunications systems: A survey. *IEEE Trans. Softw. Eng.*, 24(10):779–796, 1998.
- [LTX01] L. Lorentsen, A.-P. Tuovinen, and J. Xu. Modelling feature interactions in mobile phones. In *Feature Interaction in Composed Systems (ECOOP 2001)*, pages 7–13, Budapest, Hungary, 2001.
- [LW03] Dean Leffingwell and Don Widrig. *Managing Software Requirements: A Use Case Approach*. Pearson Education, 2003.

- [MS01] John D. McGregor and David A. Sykes. *A practical guide to testing object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.
- [Mye79] Glenford J. Myers. *Art of Software Testing*. John Wiley & Sons, Inc., New York, NY, USA, 1979.
- [PIO05] Fredrik Pettersson, Martin Ivarsson, and Peter Ohman. Automotive use case standard for embedded systems. In *SEAS '05: Proceedings of the second international workshop on Software engineering for automotive systems*, pages 1–6, New York, NY, USA, 2005. ACM Press.
- [PSC⁺02] Elke Pulvermueller, Andreas Speck, James Coplien, Maja D'Hondt, and Wolfgang De Meuter. Feature interaction in composed systems. In *ECOOP '01: Proceedings of the Workshops on Object-Oriented Technology*, pages 86–97, London, UK, 2002. Springer-Verlag.
- [Reg05] John Regehr. Random testing of interrupt-driven software. In *EMSOFT '05: Proceedings of the 5th ACM international conference on Embedded software*, pages 290–298, New York, NY, USA, 2005. ACM Press.
- [RL03] Alexander Ran and Raimondas Lencevicius. Making sense of runtime architecture for mobile phone software. In *ESEC/FSE-11: Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering*, pages 367–370, New York, NY, USA, 2003. ACM Press.
- [SW01] Geri Schneider and Jason P. Winters. *Applying Use Cases: A Practical Guide*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2 edition, 2001.
- [TB02] J. Tretmans and E. Brinksma. Côte de Resyste – Automated Model Based Testing. In M. Schweizer, editor, *Progress 2002 – 3rd Workshop on Embedded Systems*, pages 246–255, Utrecht, The Netherlands, October 24 2002. STW Technology Foundation.

-
- [Tre96] Jan Tretmans. Test generation with inputs, outputs, and quiescence. In *TACAs '96: Proceedings of the Second International Workshop on Tools and Algorithms for Construction and Analysis of Systems*, pages 127–146, London, UK, 1996. Springer-Verlag.
- [TWFL98] C. Reid Turner, Alexander L. Wolf, Alfonso Fuggetta, and Luigi Lavazza. Feature engineering. In *IWSSD '98: Proceedings of the 9th international workshop on Software specification and design*, page 162, Washington, DC, USA, 1998. IEEE Computer Society.
- [UKM03] Sebastian Uchitel, Jeff Kramer, and Jeff Magee. Behaviour model elaboration using partial labelled transition systems. In *ESEC/FSE-11: Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering*, pages 19–27, New York, NY, USA, 2003. ACM Press.

Apêndice A

Casos de Teste

Neste apêndice serão apresentados os casos de teste gerados a partir da Seção 5.2 e os casos de teste obtidos durante o estudo de caso.

A.1 Casos de Teste de Interação Dinâmica Gerados na Seção 5.2

Condição Inicial	A memória não está cheia
------------------	--------------------------

Passos	Resultados Esperados
Ir até a Central de Mensagens	A pasta "Mensagens Prediletas" é exibida
Enviar um aviso de um outro celular para o celular sob teste	O aviso é exibido numa caixa de diálogo
Selecionar a opção "Ok"	Volta para a aplicação que estava executando anteriormente
Ir até a "Caixa de Entrada"	Todas as mensagens da "Caixa de Entrada" são exibidas
Ir até uma mensagem	A mensagem é destacada
Ir até o "Menu de Contexto"	A opção "Mover para Mensagens Prediletas" é exibida
Selecionar a opção "Mover para Mensagens Prediletas"	"Mensagem movida para a pasta Mensagens Prediletas" é exibida

Figura A.1: Caso de Teste 01 - Obtido a partir do Modelo da Figura 4.10

Passos	Resultados Esperados
Ir até a Central de Mensagens	A pasta "Mensagens Prediletas" é exibida
Ir até a "Caixa de Entrada"	Todas as mensagens da "Caixa de Entrada" são exibidas
Enviar um aviso de um outro celular para o celular sob teste	O aviso é exibido numa caixa de diálogo
Selecionar a opção "Ok"	Volta para a aplicação que estava executando anteriormente
Ir até uma mensagem	A mensagem é destacada
Ir até o "Menu de Contexto"	A opção "Mover para Mensagens Prediletas" é exibida
Selecionar a opção "Mover para Mensagens Prediletas"	"Mensagem movida para a pasta Mensagens Prediletas" é exibida

Figura A.2: Caso de Teste 02 - Obtido a partir do Modelo da Figura 4.10

Passos	Resultados Esperados
Ir até a Central de Mensagens	A pasta "Mensagens Prediletas" é exibida
Ir até a "Caixa de Entrada"	Todas as mensagens da "Caixa de Entrada" são exibidas
Ir até uma mensagem	A mensagem é destacada
Enviar um aviso de um outro celular para o celular sob teste	O aviso é exibido numa caixa de diálogo
Selecionar a opção "Ok"	Volta para a aplicação que estava executando anteriormente
Ir até o "Menu de Contexto"	A opção "Mover para Mensagens Prediletas" é exibida
Selecionar a opção "Mover para Mensagens Prediletas"	"Mensagem movida para a pasta Mensagens Prediletas" é exibida

Figura A.3: Caso de Teste 03 - Obtido a partir do Modelo da Figura 4.10

Passos	Resultados Esperados
Ir até a Central de Mensagens	A pasta "Mensagens Prediletas" é exibida
Ir até a "Caixa de Entrada"	Todas as mensagens da "Caixa de Entrada" são exibidas
Ir até uma mensagem	A mensagem é destacada
Ir até o "Menu de Contexto"	A opção "Mover para Mensagens Prediletas" é exibida
Enviar um aviso de um outro celular para o celular sob teste	O aviso é exibido numa caixa de diálogo
Selecionar a opção "Ok"	Volta para a aplicação que estava executando anteriormente
Selecionar a opção "Mover para Mensagens Prediletas"	"Mensagem movida para a pasta Mensagens Prediletas" é exibida

Figura A.4: Caso de Teste 04 - Obtido a partir do Modelo da Figura 4.10

Passos	Resultados Esperados
Ir até a Central de Mensagens	A pasta "Mensagens Prediletas" é exibida
Ir até a "Caixa de Entrada"	Todas as mensagens da "Caixa de Entrada" são exibidas
Ir até uma mensagem	A mensagem é destacada
Ir até o "Menu de Contexto"	A opção "Mover para Mensagens Prediletas" é exibida
Selecionar a opção "Mover para Mensagens Prediletas"	"Mensagem movida para a pasta Mensagens Prediletas" é exibida
Enviar um aviso de um outro celular para o celular sob teste	O aviso é exibido numa caixa de diálogo
Selecionar a opção "Ok"	Volta para a aplicação que estava executando anteriormente

Figura A.5: Caso de Teste 05 - Obtido a partir do Modelo da Figura 4.10

Condição Inicial	A memória está cheia
Passos	Resultados Esperados
Ir até a Central de Mensagens	A pasta "Mensagens Prediletas" é exibida
Enviar um aviso de um outro celular para o celular sob teste	O aviso é exibido numa caixa de diálogo
Selecionar a opção "Ok"	Volta para a aplicação que estava executando anteriormente
Ir até a "Caixa de Entrada"	Todas as mensagens da "Caixa de Entrada" são exibidas
Ir até uma mensagem	A mensagem é destacada
Ir até o "Menu de Contexto"	A opção "Mover para Mensagens Prediletas" é exibida
Selecionar a opção "Mover para Mensagens Prediletas"	Caixa de diálogo "Memória Insuficiente" é exibida
Confirmar a caixa de diálogo de informação da memória	O conteúdo da mensagem é exibido

Figura A.6: Caso de Teste 06 - Obtido a partir do Modelo da Figura 4.10

Condição Inicial	A memória está cheia
Passos	Resultados Esperados
Ir até a Central de Mensagens	A pasta "Mensagens Prediletas" é exibida
Ir até a "Caixa de Entrada"	Todas as mensagens da "Caixa de Entrada" são exibidas
Enviar um aviso de um outro celular para o celular sob teste	O aviso é exibido numa caixa de diálogo
Selecionar a opção "Ok"	Volta para a aplicação que estava executando anteriormente
Ir até uma mensagem	A mensagem é destacada
Ir até o "Menu de Contexto"	A opção "Mover para Mensagens Prediletas" é exibida
Selecionar a opção "Mover para Mensagens Prediletas"	Caixa de diálogo "Memória Insuficiente" é exibida
Confirmar a caixa de diálogo de informação da memória	O conteúdo da mensagem é exibido

Figura A.7: Caso de Teste 07 - Obtido a partir do Modelo da Figura 4.10

Condição Inicial	A memória está cheia
Passos	Resultados Esperados
Ir até a Central de Mensagens	A pasta "Mensagens Prediletas" é exibida
Ir até a "Caixa de Entrada"	Todas as mensagens da "Caixa de Entrada" são exibidas
Ir até uma mensagem	A mensagem é destacada
Enviar um aviso de um outro celular para o celular sob teste	O aviso é exibido numa caixa de diálogo
Selecionar a opção "Ok"	Volta para a aplicação que estava executando anteriormente
Ir até o "Menu de Contexto"	A opção "Mover para Mensagens Prediletas" é exibida
Selecionar a opção "Mover para Mensagens Prediletas"	Caixa de diálogo "Memória Insuficiente" é exibida
Confirmar a caixa de diálogo de informação da memória	O conteúdo da mensagem é exibido

Figura A.8: Caso de Teste 08 - Obtido a partir do Modelo da Figura 4.10

Condição Inicial	A memória está cheia
Passos	Resultados Esperados
Ir até a Central de Mensagens	A pasta "Mensagens Prediletas" é exibida
Ir até a "Caixa de Entrada"	Todas as mensagens da "Caixa de Entrada" são exibidas
Ir até uma mensagem	A mensagem é destacada
Ir até o "Menu de Contexto"	A opção "Mover para Mensagens Prediletas" é exibida
Enviar um aviso de um outro celular para o celular sob teste	O aviso é exibido numa caixa de diálogo
Selecionar a opção "Ok"	Volta para a aplicação que estava executando anteriormente
Selecionar a opção "Mover para Mensagens Prediletas"	Caixa de diálogo "Memória Insuficiente" é exibida
Confirmar a caixa de diálogo de informação da memória	O conteúdo da mensagem é exibido

Figura A.9: Caso de Teste 09 - Obtido a partir do Modelo da Figura 4.10

A.2 Casos de Teste Gerados no Estudo de Caso

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Enviar"	A tela "Enviar Mensagem" é exibida com os contatos da lista de contatos
Escolher pelo menos um contato	Pelo menos um contato é escolhido
Selecionar a opção "Enviar"	Aviso "Mensagem Enviada" é exibido

Figura A.10: Caso de Teste 01 - Obtido a partir do Modelo da Figura 6.7

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Enviar"	A tela "Enviar Mensagem" é exibida com os contatos da lista de contatos
Escolher pelo menos um contato	Pelo menos um contato é escolhido
Selecionar a opção "Opções"	A tela "Opções" é exibida
Selecionar a opção "Cancelar"	A aplicação "Compositor de Mensagens" é exibida com a última mensagem

Figura A.11: Caso de Teste 02 - Obtido a partir do Modelo da Figura 6.7

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Opções"	A opção "Opções de Envio" é exibida
Selecionar a opção "Opções de Envio"	A tela "Opções de Mensagens SMS" é exibida
Preencher qualquer campo suportado somente por mensagens MMS	Qualquer campo suportado somente por mensagens MMS é preenchido
Selecionar a opção "Pronto"	Caixa de diálogo para confirmação é exibida
Selecionar a opção "Sim"	A aplicação "Compositor de Mensagens" é exibida. O valor do Contador de Caracteres é atualizado. A palavra "MMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Enviar"

Figura A.12: Caso de Teste 03 - Obtido a partir do Modelo da Figura 6.7

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Opções"	A opção "Opções de Envio" é exibida
Selecionar a opção "Opções de Envio"	A tela "Opções de Mensagens SMS" é exibida
Preencher qualquer campo suportado somente por mensagens MMS	Qualquer campo suportado somente por mensagens MMS é preenchido
Selecionar a opção "Pronto"	Caixa de diálogo para confirmação é exibida
Selecionar a opção "Não"	A tela "Opções de Mensagens SMS" é exibida

Figura A.13: Caso de Teste 04 - Obtido a partir do Modelo da Figura 6.7

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Opções"	A opção "Opções de Envio" é exibida
Selecionar a opção "Opções de Envio"	A tela "Opções de Mensagens SMS" é exibida
Preencher qualquer campo suportado somente por mensagens MMS	Qualquer campo suportado somente por mensagens MMS é preenchido
Selecionar a opção "Cancelar"	A aplicação "Compositor de Mensagens" é exibida com a última mensagem

Figura A.14: Caso de Teste 05 - Obtido a partir do Modelo da Figura 6.7

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Selecionar a opção "Cancelar"	A Central de Mensagens é exibida

Figura A.15: Caso de Teste 06 - Obtido a partir do Modelo da Figura 6.7

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Enviar"	A tela "Enviar Mensagem" é exibida com os contatos da lista de contatos
Escolher pelo menos um contato	Pelo menos um contato é escolhido
Selecionar a opção "Enviar"	Aviso "Mensagem Enviada" é exibido

Figura A.16: Caso de Teste de Interação Dinâmica 01 - Primeiro Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Enviar"	A tela "Enviar Mensagem" é exibida com os contatos da lista de contatos
Escolher pelo menos um contato	Pelo menos um contato é escolhido
Selecionar a opção "Opções"	A tela "Opções" é exibida
Selecionar a opção "Cancelar"	A aplicação "Compositor de Mensagens" é exibida com a última mensagem

Figura A.17: Caso de Teste de Interação Dinâmica 02 - Primeiro Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Opções"	A opção "Opções de Envio" é exibida
Selecionar a opção "Opções de Envio"	A tela "Opções de Mensagens SMS" é exibida
Preencher qualquer campo suportado somente por mensagens MMS	Qualquer campo suportado somente por mensagens MMS é preenchido
Selecionar a opção "Pronto"	Caixa de diálogo para confirmação é exibida
Selecionar a opção "Sim"	A aplicação "Compositor de Mensagens" é exibida. O valor do Contador de Caracteres é atualizado. A palavra "MMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Enviar"

Figura A.18: Caso de Teste de Interação Dinâmica 03 - Primeiro Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Opções"	A opção "Opções de Envio" é exibida
Selecionar a opção "Opções de Envio"	A tela "Opções de Mensagens SMS" é exibida
Preencher qualquer campo suportado somente por mensagens MMS	Qualquer campo suportado somente por mensagens MMS é preenchido
Selecionar a opção "Pronto"	Caixa de diálogo para confirmação é exibida
Selecionar a opção "Não"	A tela "Opções de Mensagens SMS" é exibida

Figura A.19: Caso de Teste de Interação Dinâmica 04 - Primeiro Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Opções"	A opção "Opções de Envio" é exibida
Selecionar a opção "Opções de Envio"	A tela "Opções de Mensagens SMS" é exibida
Preencher qualquer campo suportado somente por mensagens MMS	Qualquer campo suportado somente por mensagens MMS é preenchido
Selecionar a opção "Cancelar"	A aplicação "Compositor de Mensagens" é exibida com a última mensagem

Figura A.20: Caso de Teste de Interação Dinâmica 05 - Primeiro Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Selecionar a opção "Cancelar"	A Central de Mensagens é exibida

Figura A.21: Caso de Teste de Interação Dinâmica 06 - Primeiro Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Selecionar a opção "Enviar"	A tela "Enviar Mensagem" é exibida com os contatos da lista de contatos
Escolher pelo menos um contato	Pelo menos um contato é escolhido
Selecionar a opção "Enviar"	Aviso "Mensagem Enviada" é exibido

Figura A.22: Caso de Teste de Interação Dinâmica 01 - Segundo Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Selecionar a opção "Enviar"	A tela "Enviar Mensagem" é exibida com os contatos da lista de contatos
Escolher pelo menos um contato	Pelo menos um contato é escolhido
Selecionar a opção "Opções"	A tela "Opções" é exibida
Selecionar a opção "Cancelar"	A aplicação "Compositor de Mensagens" é exibida com a última mensagem

Figura A.23: Caso de Teste de Interação Dinâmica 02 - Segundo Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Selecionar a opção "Opções"	A opção "Opções de Envio" é exibida
Selecionar a opção "Opções de Envio"	A tela "Opções de Mensagens SMS" é exibida
Preencher qualquer campo suportado somente por mensagens MMS	Qualquer campo suportado somente por mensagens MMS é preenchido
Selecionar a opção "Pronto"	Caixa de diálogo para confirmação é exibida
Selecionar a opção "Sim"	A aplicação "Compositor de Mensagens" é exibida. O valor do Contador de Caracteres é atualizado. A palavra "MMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Enviar"

Figura A.24: Caso de Teste de Interação Dinâmica 03 - Segundo Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Selecionar a opção "Opções"	A opção "Opções de Envio" é exibida
Selecionar a opção "Opções de Envio"	A tela "Opções de Mensagens SMS" é exibida
Preencher qualquer campo suportado somente por mensagens MMS	Qualquer campo suportado somente por mensagens MMS é preenchido
Selecionar a opção "Pronto"	Caixa de diálogo para confirmação é exibida
Selecionar a opção "Não"	A tela "Opções de Mensagens SMS" é exibida

Figura A.25: Caso de Teste de Interação Dinâmica 04 - Segundo Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Selecionar a opção "Opções"	A opção "Opções de Envio" é exibida
Selecionar a opção "Opções de Envio"	A tela "Opções de Mensagens SMS" é exibida
Preencher qualquer campo suportado somente por mensagens MMS	Qualquer campo suportado somente por mensagens MMS é preenchido
Selecionar a opção "Cancelar"	A aplicação "Compositor de Mensagens" é exibida com a última mensagem

Figura A.26: Caso de Teste de Interação Dinâmica 05 - Segundo Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Enviar"	A tela "Enviar Mensagem" é exibida com os contatos da lista de contatos
Escolher pelo menos um contato	Pelo menos um contato é escolhido
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Selecionar a opção "Ler"	A mensagem é mostrada
Selecionar a opção "Voltar"	Volta para a aplicação que estava executando antes da interrupção do aviso de chegada de uma mensagem
Selecionar a opção "Enviar"	Aviso "Mensagem Enviada" é exibido

Figura A.27: Caso de Teste de Interação Dinâmica 01 - Terceiro Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Enviar"	A tela "Enviar Mensagem" é exibida com os contatos da lista de contatos
Escolher pelo menos um contato	Pelo menos um contato é escolhido
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Selecionar a opção "Cancelar"	Volta para a aplicação que estava executando antes da interrupção do aviso de chegada de uma mensagem
Selecionar a opção "Enviar"	Aviso "Mensagem Enviada" é exibido

Figura A.28: Caso de Teste de Interação Dinâmica 02 - Terceiro Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Enviar"	A tela "Enviar Mensagem" é exibida com os contatos da lista de contatos
Escolher pelo menos um contato	Pelo menos um contato é escolhido
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Selecionar a opção "Ler"	A mensagem é mostrada
Selecionar a opção "Voltar"	Volta para a aplicação que estava executando antes da interrupção do aviso de chegada de uma mensagem
Selecionar a opção "Opções"	A tela "Opções" é exibida
Selecionar a opção "Cancelar"	A aplicação "Compositor de Mensagens" é exibida com a última mensagem

Figura A.29: Caso de Teste de Interação Dinâmica 03 - Terceiro Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Enviar"	A tela "Enviar Mensagem" é exibida com os contatos da lista de contatos
Escolher pelo menos um contato	Pelo menos um contato é escolhido
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Selecionar a opção "Cancelar"	Volta para a aplicação que estava executando antes da interrupção do aviso de chegada de uma mensagem
Selecionar a opção "Opções"	A tela "Opções" é exibida
Selecionar a opção "Cancelar"	A aplicação "Compositor de Mensagens" é exibida com a última mensagem

Figura A.30: Caso de Teste de Interação Dinâmica 04 - Terceiro Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Opções"	A opção "Opções de Envio" é exibida
Selecionar a opção "Opções de Envio"	A tela "Opções de Mensagens SMS" é exibida
Preencher qualquer campo suportado somente por mensagens MMS	Qualquer campo suportado somente por mensagens MMS é preenchido
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Selecionar a opção "Ler"	A mensagem é mostrada
Selecionar a opção "Voltar"	Volta para a aplicação que estava executando antes da interrupção do aviso de chegada de uma mensagem
Selecionar a opção "Pronto"	Caixa de diálogo para confirmação é exibida
Selecionar a opção "Sim"	A aplicação "Compositor de Mensagens" é exibida. O valor do Contador de Caracteres é atualizado. A palavra "MMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Enviar"

Figura A.31: Caso de Teste de Interação Dinâmica 01 - Quarto Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Opções"	A opção "Opções de Envio" é exibida
Selecionar a opção "Opções de Envio"	A tela "Opções de Mensagens SMS" é exibida
Preencher qualquer campo suportado somente por mensagens MMS	Qualquer campo suportado somente por mensagens MMS é preenchido
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Selecionar a opção "Cancelar"	Volta para a aplicação que estava executando antes da interrupção do aviso de chegada de uma mensagem
Selecionar a opção "Pronto"	Caixa de diálogo para confirmação é exibida
Selecionar a opção "Sim"	A aplicação "Compositor de Mensagens" é exibida. O valor do Contador de Caracteres é atualizado. A palavra "MMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Enviar"

Figura A.32: Caso de Teste de Interação Dinâmica 02 - Quarto Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Opções"	A opção "Opções de Envio" é exibida
Selecionar a opção "Opções de Envio"	A tela "Opções de Mensagens SMS" é exibida
Preencher qualquer campo suportado somente por mensagens MMS	Qualquer campo suportado somente por mensagens MMS é preenchido
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Selecionar a opção "Ler"	A mensagem é mostrada
Selecionar a opção "Voltar"	Volta para a aplicação que estava executando antes da interrupção do aviso de chegada de uma mensagem
Selecionar a opção "Pronto"	Caixa de diálogo para confirmação é exibida
Selecionar a opção "Não"	A tela "Opções de Mensagens SMS" é exibida

Figura A.33: Caso de Teste de Interação Dinâmica 03 - Quarto Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Opções"	A opção "Opções de Envio" é exibida
Selecionar a opção "Opções de Envio"	A tela "Opções de Mensagens SMS" é exibida
Preencher qualquer campo suportado somente por mensagens MMS	Qualquer campo suportado somente por mensagens MMS é preenchido
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Selecionar a opção "Cancelar"	Volta para a aplicação que estava executando antes da interrupção do aviso de chegada de uma mensagem
Selecionar a opção "Pronto"	Caixa de diálogo para confirmação é exibida
Selecionar a opção "Não"	A tela "Opções de Mensagens SMS" é exibida

Figura A.34: Caso de Teste de Interação Dinâmica 04 - Quarto Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Opções"	A opção "Opções de Envio" é exibida
Selecionar a opção "Opções de Envio"	A tela "Opções de Mensagens SMS" é exibida
Preencher qualquer campo suportado somente por mensagens MMS	Qualquer campo suportado somente por mensagens MMS é preenchido
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Selecionar a opção "Ler"	A mensagem é mostrada
Selecionar a opção "Voltar"	Volta para a aplicação que estava executando antes da interrupção do aviso de chegada de uma mensagem
Selecionar a opção "Cancelar"	A aplicação "Compositor de Mensagens" é exibida com a última mensagem

Figura A.35: Caso de Teste de Interação Dinâmica 05 - Quarto Cenário

Passos	Resultados Esperados
Ir até a Central de Mensagens	A opção "Criar Mensagem" é exibida
Selecionar a opção "Criar Mensagem"	A aplicação "Compositor de Mensagens" é iniciada. O valor do Contador de Caracteres é zero. A palavra "SMS" é mostrada junto ao Contador de Caracteres. A opção direita é "Cancelar"
Digitar pelo menos um caractere na mensagem	O Contador de Caracteres é atualizado. A opção direita é "Enviar"
Selecionar a opção "Opções"	A opção "Opções de Envio" é exibida
Selecionar a opção "Opções de Envio"	A tela "Opções de Mensagens SMS" é exibida
Preencher qualquer campo suportado somente por mensagens MMS	Qualquer campo suportado somente por mensagens MMS é preenchido
Enviar uma mensagem de um outro celular para o celular sob teste	Aviso "Nova Mensagem" é mostrado
Selecionar a opção "Cancelar"	Volta para a aplicação que estava executando antes da interrupção do aviso de chegada de uma mensagem
Selecionar a opção "Cancelar"	A aplicação "Compositor de Mensagens" é exibida com a última mensagem

Figura A.36: Caso de Teste de Interação Dinâmica 06 - Quarto Cenário