



TÉCNICA DE LEITURA PARA INSPEÇÃO DE DIAGRAMAS DE ESTADOS COM  
BASE EM DIAGRAMAS DE ATIVIDADES ESPECIFICANDO OS CASOS DE USO DO  
SOFTWARE

Karen Miyuki Nakazato

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Guilherme Horta Travassos

Rio de Janeiro  
Setembro de 2013

TÉCNICA DE LEITURA PARA INSPEÇÃO DE DIAGRAMAS DE ESTADOS COM  
BASE EM DIAGRAMAS DE ATIVIDADES ESPECIFICANDO OS CASOS DE USO DO  
SOFTWARE

Karen Miyuki Nakazato

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO  
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE)  
DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS  
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM  
CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

---

Prof. Guilherme Horta Travassos, D.Sc.

---

Prof. Toacy Cavalcante de Oliveira, D.Sc.

---

Prof. Márcio de Oliveira Barros, D.Sc.

RIO DE JANEIRO, RJ – BRASIL  
SETEMBRO DE 2013

Nakazato, Karen Miyuki

Técnica de Leitura para Inspeção de Diagramas de Estados com base em Diagramas de Atividades Especificando os Casos de Uso do Software / Karen Miyuki Nakazato – Rio de Janeiro: UFRJ/COPPE, 2013.

XII, 149 p.: il.; 29,7 cm.

Orientador: Guilherme Horta Travassos.

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2013.

Referências Bibliográficas: p. 113-118.

1. Inspeção de Software. 2. Diagrama de Estados. 3. Diagrama de Atividades. 4. Engenharia de Software Experimental. I. Travassos, Guilherme Horta II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Aos meus pais, pela dedicação e exemplo de vida.

Aos meus avós pela compreensão.

Aos meus irmãos pelo carinho.

## **Agradecimentos**

Dedico o meu eterno agradecimento à todos que estiveram ao meu lado nesses anos. A todos que de alguma forma contribuíram para a minha formação não só profissional, mas também pessoal.

Em especial, agradeço aos meus pais Ademir e Nancy pelo apoio e amor incondicional em todos os momentos. Pelo incentivo e o esforço que foram cruciais para o término do mestrado.

À Batchan e Ditchan, meus avós, que me acolheram durante o mestrado, pela compreensão e incentivo.

Aos meus irmãos, Junji e Akio, pelo carinho e companheirismo.

Ao meu orientador, prof. Guilherme Horta Travassos, pela dedicação e apoio na minha pesquisa, sem o qual não seria possível realizar o mesmo trabalho.

Aos professores Márcio Barros e Toacy Cavalcante, por participarem de minha banca de defesa de mestrado.

Aos meus amigos e companheiros da COPPE, alguns já egressos, Victor Vidigal, Arthur Siqueira, Jobson Massollar, Breno França, Rafael de Mello, Rafael Espirito Santo, Leonardo Mota, Wallace Martinho, Paulo Sérgio Medeiros, Vitor Lopes, Rodrigo Spínola, Talita Ribeiro, Thiago de Souza, Verônica Taquette e Ivens Portugal pelos conselhos e excelentes momentos vividos durante esses anos.

À Taísa Guidini Gonçalves, sempre disposta e prestativa.

Aos meus amigos de Campo Grande que me apoiaram e incentivaram, e os amigos do Rio de Janeiro pela receptividade e companheirismo.

Aos alunos que participaram dos estudos que compõem esta dissertação.

Ao CNPQ pelo apoio financeiro.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

TÉCNICA DE LEITURA PARA INSPEÇÃO DE DIAGRAMAS DE ESTADOS COM  
BASE EM DIAGRAMAS DE ATIVIDADES ESPECIFICANDO OS CASOS DE USO DO  
SOFTWARE

Karen Miyuki Nakazato

Setembro/2013

Orientador: Guilherme Horta Travassos

Programa: Engenharia de Sistemas e Computação

Este trabalho apresenta *Shiô*, uma técnica de leitura para inspecionar diagramas de estados com o apoio de diagramas de atividades que especificam os casos de uso do software. A técnica foi desenvolvida com base em evidência. Seu desenvolvimento se justifica pela necessidade de uma técnica deste tipo para os projetos de software atuais e, ao mesmo tempo, a carência de tecnologias para a inspeção de diagramas de fluxo de atividades envolvendo diagramas de estado, conforme indicado pelos resultados de uma *quasi-revisão* sistemática (estudo secundário). A primeira versão de *Shiô* foi avaliada através de dois estudos de viabilidade *in-vitro* que indicaram sua capacidade em identificar defeitos, principalmente aqueles usualmente não capturados por inspeção *ad-hoc*, apesar do tempo dispendido pelos inspetores ser maior. Entretanto, estes resultados permitiram evoluir a técnica *Shiô*, simplificando sua aplicação e aprimorando sua organização visando aumentar sua capacidade para detecção de defeitos nos diagramas de estados.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

A READING TECHNIQUE FOR INSPECTION OF STATE DIAGRAMS BASED ON  
ACTIVITY DIAGRAMS DESCRIBING SOFTWARE USE

Karen Miyuki Nakazato

September/2013

Advisor: Guilherme Horta Travassos

Department: Computer Science and Systems Engineering

This work introduces *Shiô*, a reading technique to inspect state diagrams based on activity diagrams describing the software use cases. The technique had been developed through an evidence based methodology. Its development can be justified by the need of such type of techniques for supporting quality in contemporary software projects considering the lack of inspection technologies to support the reading of flow based diagrams as indicated in the result of a *quasi*-systematic review (secondary study). The first version of *Shiô* was evaluated by two *in vitro* feasibility studies, which indicated its capacity of detecting defects, mainly those ones not usually captured by *ad-hoc* reading, although the increasing of time reading. These results motivated the evolution of *Shiô*, aiming at to simplify its descriptions and reorganization to improve the capacity of inspectors on identifying defects in state diagrams.

# ÍNDICE

1	Introdução.....	1
1.1	Motivação e Contexto: Descrição do Problema .....	1
1.2	Objetivo da Solução Proposta .....	5
1.3	Metodologia de Trabalho.....	6
1.4	Organização do Trabalho .....	8
2	Fundamentação Teórica .....	9
2.1	Introdução .....	9
2.2	Diagrama de Atividades .....	11
2.2.1	Possíveis Usos de Diagramas de Atividades .....	12
2.2.2	Estruturas.....	12
2.3	Diagrama de Estados.....	15
2.3.1	Estruturas.....	16
2.4	Inspeção de Software.....	21
2.4.1	Processo de Inspeção .....	22
2.4.2	Taxonomia de defeitos .....	24
2.4.3	Exemplos de Técnicas de Inspeção .....	24
2.5	Conclusão .....	29
3	Técnicas de Inspeção para Diagramas de Fluxos de Atividades.....	31
3.1	Introdução .....	31
3.2	Planejamento da <i>quasi</i> -Revisão Sistemática .....	32
3.3	Execução da <i>quasi</i> -Revisão Sistemática .....	35
3.3.1	Seleção dos artigos .....	36
3.3.2	Extração das Informações .....	37
3.3.3	Avaliação da Qualidade dos Artigos .....	40
3.4	Resultados encontrados.....	42
3.4.1	Tanriöver e Bilgen (2007) .....	43
3.4.2	Cooper <i>et al.</i> (2007).....	43
3.4.3	de Mello <i>et al.</i> (2010).....	44
3.5	Atualização da <i>quasi</i> -Revisão Sistemática .....	45
3.6	Ameaças à validade .....	47
3.7	Conclusão .....	48
4	Técnica de Leitura para inspecionar Diagramas de Estados com referência nos Diagramas de Atividades .....	49
4.1	Introdução .....	49
4.2	Modelos Inspeccionados.....	51
4.3	Premissas para Aplicação da Técnica.....	54
4.4	Relação entre diagramas de estados e diagramas de atividades .....	54
4.5	A Técnica <i>Shiô</i> .....	58
4.5.1	Estrutura da Técnica <i>Shiô</i> .....	58
4.5.2	Primeira Versão da Técnica.....	65
4.5.3	Relatório de Discrepância.....	69
4.6	Conclusão .....	71
5	Estudos Experimentais .....	72
5.1	Introdução .....	72
5.2	Primeiro Estudo.....	73
5.2.1	Planejamento .....	74
5.2.2	Projeto Experimental .....	75
5.2.3	Instrumentação.....	77



5.2.4	Execução.....	77
5.2.5	Análise Preliminar dos Resultados .....	81
5.3	Segundo Estudo.....	88
5.3.1	Planejamento e Projeto Experimental.....	89
5.3.2	Instrumentação e Execução .....	91
5.3.3	Análise Preliminar dos Resultados .....	92
5.4	Análise Qualitativa dos Estudos .....	93
5.5	Lições Aprendidas com os Estudos.....	102
5.6	Ameaças à validade .....	103
5.7	Segunda Versão da Técnica .....	104
5.8	Conclusão .....	109
6	Conclusão e Trabalhos Futuros .....	110
6.1	Considerações Finais.....	110
6.2	Contribuições da Pesquisa.....	111
6.3	Limitações .....	111
6.4	Trabalhos Futuros .....	112
	REFERÊNCIAS BIBLIOGRÁFICAS.....	113
	APÊNDICE A – Formulários Utilizados nos Estudos.....	119
A.1.	Formulário de Consentimento .....	119
A.2.	Formulário de Caracterização .....	121
	APÊNDICE B – Diagramas utilizados nos Estudos.....	124
B.1.	Modelos e Regras de Negócio sobre Conta Utilizados na Inspeções .....	124
B.2	Modelos e Regras de Negócio sobre Movimento Utilizados na Inspeções .....	128
	APÊNDICE C – Relatórios de Discrepância Utilizados nos Estudos.....	145
C.1.	Relatório de Discrepância da Inspeção <i>Ad-hoc</i> .....	145
C.2.	Relatório de Discrepância da Inspeção com a Técnica <i>Shiô</i> .....	146
	APÊNDICE D – Questionário de Avaliação da Técnica de Leitura Shiô.....	147
D.1.	Questionário de Avaliação da Técnica <i>Shiô</i> .....	147

# ÍNDICE DE FIGURAS

Figura 1-1 Diagrama de atividades que representa a descrição do caso de uso, conforme MASSOLLAR (2011).	5
Figura 1-2 Metodologia de pesquisa adotada.	7
Figura 2-1 Diagramas estruturais e comportamentais da UML (Adaptado da OMG, 2011).	9
Figura 2-2 Exemplos de Técnicas de Inspeção em Diagramas da UML (Adaptado de TRAVASSOS, 2002).	11
Figura 2-3 Exemplo de duas ações.	12
Figura 2-4 Exemplo da atividade Separar Produtos.	13
Figura 2-5 Nós inicial e final de atividade.	14
Figura 2-6 Nó de decisão (Adaptado da OMG, 2011).	15
Figura 2-7 Exemplo de estado.	16
Figura 2-8 Diagrama de estados para Automóvel, tendo como foco o movimento (Adaptado de SILVA, 2007).	17
Figura 2-9 Diagrama de estados para Automóvel no contexto de uma oficina mecânica (Adaptado de SILVA, 2007).	17
Figura 2-10 (a) Estado inicial (b) Estado final.	18
Figura 2-11 Diagrama de estados de Venda.	19
Figura 2-12 Modelagem de estados para Automóvel, com estado composto (Adaptado de SILVA, 2007).	20
Figura 2-13 Modelagem alternativa para Automóvel, com estado composto (Adaptado de SILVA, 2007).	20
Figura 2-14 Modelagem de Automóvel com estado submáquina (SILVA, 2007).	21
Figura 2-15 Processo de inspeção de software de FAGAN (de acordo com WONG, 2006).	22
Figura 3-1 Artigos distintos retornados pelas máquinas de busca.	36
Figura 4-1 Representação gráfica da Técnica de Leitura de Diagramas de Estados com Diagramas de Atividades dentro do contexto de OORTs.	50
Figura 4-2 Diagrama de atividades de Realizar Venda.	55
Figura 4-3 Diagrama de estados de Venda.	55
Figura 4-4 Visão geral da técnica proposta.	58
Figura 4-5 Diagrama de estados de Venda utilizado para ilustrar a aplicação da técnica <i>Shiô</i> .	59
Figura 4-6 Diagrama de estados - Passo 1 da Técnica <i>Shiô</i> .	60
Figura 4-7 Diagrama de atividades - Passo 1 da Técnica <i>Shiô</i> .	61
Figura 4-8 Diagrama de atividades - Passo 2 da Técnica <i>Shiô</i> .	62
Figura 4-9 Diagrama de atividades - Passo 3 da Técnica <i>Shiô</i> .	63
Figura 4-10 Diagrama de estados - Passo 3 da Técnica <i>Shiô</i> .	64
Figura 4-11 Diagrama de atividades - Passo 4 da Técnica <i>Shiô</i> .	65
Figura 5-1 Quantidade de defeitos encontrados pela inspeção <i>ad-hoc</i> e por <i>Shiô</i> no conjunto de modelos de Conta.	84
Figura 5-2 Quantidade de defeitos encontrados pela inspeção <i>ad-hoc</i> e <i>Shiô</i> no conjunto de modelos de Movimento.	85
Figura 5-3 Grau de confiabilidade estatística (Gardner e Altman, 1989).	97
Figura 5-4 Respostas do questionário de avaliação da técnica referente à adesão da técnica.	98
Figura 5-5 Respostas do questionário de avaliação da técnica referente ao grau de dificuldade da técnica.	98
Figura 5-6 Respostas do questionário de avaliação da técnica referente ao auxílio da técnica durante a inspeção.	98
Figura 5-7 Respostas do questionário de avaliação da técnica referente ao uso em oportunidades futuras de inspeção.	99

Figura 5-8 Respostas do questionário de avaliação da técnica referente à qualidade do treinamento aplicado.....	99
Figura 5-9 Respostas do questionário de avaliação da técnica referente à dificuldade em entender os modelos inspecionados. ....	100
Figura 5-10 Respostas do questionário de avaliação da técnica referente à dificuldade em entender o domínio dos modelos inspecionados.....	100
Figura 5-11 Respostas do questionário de avaliação da técnica referente à complexidade dos modelos inspecionados. ....	101
Figura 5-12 Respostas do questionário de avaliação da técnica referente ao modelo mais fácil de inspecionar.....	101

## ÍNDICE DE TABELAS

Tabela 2-1 Tipos de defeitos de software (Adaptado de TRAVASSOS, 2001).....	24
Tabela 2-2 Técnicas de Leitura OORTs (Adaptado de TRAVASSOS, 2002). ....	27
Tabela 2-3 Técnicas de Leitura para apoiar o processo de desenvolvimento ProDeS/UML (Adaptado de MARUCCI <i>et al.</i> , 2002). ....	28
Tabela 3-1 Máquinas de buscas e quantidade de artigos retornados. ....	36
Tabela 3-2 Informações extraídas dos resultados.....	38
Tabela 3-3 Artigos incluídos após os critérios de inclusão e exclusão. ....	39
Tabela 3-4 Informações extraídas dos artigos selecionados.....	39
Tabela 3-5 Conceitos com relação à qualidade dos artigos selecionados.....	41
Tabela 3-6 Máquinas de buscas e quantidade de artigos retornados na reexecução da <i>quasi-revisão sistemática</i> .....	45
Tabela 3-7 Informações extraídas de MASSOLLAR <i>et al.</i> (2011). ....	46
Tabela 4-1 Recursos sintáticos do diagrama de atividades explorados na abordagem de (MASSOLLAR, 2011).....	52
Tabela 4-2 Relação dos conceitos entre a descrição de casos de uso e diagramas de atividades (MASSOLLAR, 2011).....	52
Tabela 5-1 Objetivos específicos do primeiro estudo.....	74
Tabela 5-2 Arranjo do primeiro estudo.....	76
Tabela 5-3 Distribuição dos participantes e dos artefatos da turma da pós-graduação. .....	78
Tabela 5-4 Distribuição dos participantes e dos artefatos da turma da graduação.....	79
Tabela 5-5 <i>Outliers</i> da turma da pós-graduação.....	81
Tabela 5-6 <i>Outliers</i> da turma da graduação.....	82
Tabela 5-7 Dados da inspeção <i>ad-hoc</i> . ....	83
Tabela 5-8 Dados da inspeção com o uso da técnica <i>Shiô</i> .....	83
Tabela 5-9 Média e desvio padrão de defeitos e tempo no conjunto de Conta. ....	86
Tabela 5-10 Média e desvio padrão de defeitos e tempo no conjunto de Movimento..	86
Tabela 5-11 Inspetores não <i>outliers</i> por rodada e por conjunto de modelos. ....	88
Tabela 5-12 Objetivos específicos do segundo estudo. ....	89
Tabela 5-13 Arranjo do segundo estudo. ....	90
Tabela 5-14 Distribuição dos participantes e dos artefatos do segundo estudo. ....	91
Tabela 5-15 Inspetores da graduação que encontraram defeitos não encontrados pela inspeção <i>ad-hoc</i> .....	94

# 1 Introdução

*Neste capítulo são apresentados o contexto e a motivação para esta pesquisa. Além disso, são definidos os objetivos da pesquisa, a metodologia de pesquisa adotada e a organização do texto desta dissertação.*

## 1.1 Motivação e Contexto: Descrição do Problema

As características gerais do software, incluindo suas funcionalidades e eventuais restrições, são normalmente descritas em forma textual na fase de especificação de requisitos de um projeto de software. Estas características podem ser realizadas por representações de casos de uso que fornecem um formato semiestruturado para essa descrição. Após essa fase e de posse da especificação de requisitos, diferentes modelos de projeto podem ser construídos de acordo com a necessidade do software. Estes modelos, por sua vez, podem ser descritos através de diferentes notações, incluindo a UML (OMG, 2011), que oferece representações apropriadas para diferentes diagramas de projeto relacionados às propriedades estruturais, dinâmicas e comportamentais esperadas para a solução do software.

Portanto, a construção do software envolve a elaboração e utilização de vários modelos em diferentes níveis de abstração e perspectivas. Esta “disjunção conceitual” é importante para apoiar os desenvolvedores a tratar os diferentes aspectos relacionados ao problema e ao software. A junção de todas as representações leva a visualização da solução em software. Assim, se por um lado a multiplicidade de mecanismos e instrumentos de representação facilita a descrição do problema e sua solução, por outro lado, a garantia da qualidade dos artefatos produzidos não é uma tarefa trivial.

Diagramas estruturais (como por exemplo, diagramas de classes que permitem representar as classes e seus relacionamentos) apresentam uma visão da organização estática da solução. As representações e relacionamentos não se alteram ao longo do ciclo de vida do objeto. Por outro lado, diagramas comportamentais (tais como diagramas de atividades, estados ou sequência) descrevem aspectos dinâmicos e comportamentais

de um sistema de software, ou seja, descrevem os comportamentos que se alteram ao longo do ciclo de vida do objeto.

Os diferentes diagramas comportamentais têm objetivos e momentos apropriados para sua utilização, como por exemplo, os diagramas de atividades, que se assemelham a fluxogramas. Os fluxogramas são uma das mais antigas ferramentas de modelagem de software existente e foram precursores dos diagramas de atividades da UML 2.

Os diagramas de atividades podem ser utilizados em diferentes contextos e com diferentes propósitos, tais como: modelagem da descrição do caso de uso, modelagem de processo de negócio e modelagem de uma operação complexa que necessita de maiores detalhes, além de servirem para descrever a sequência de ações relacionada a algum comportamento particular de um objeto (BEZERRA, 2006). GUITIÉRREZ *et al.* (2008) sugerem o uso de diagramas de atividades para representar casos de uso de forma a vislumbrar o comportamento global de um caso de uso, especialmente aqueles relacionados a seus aspectos dinâmicos.

Os diagramas de estados permitem descrever o ciclo de vida de um objeto e os eventos que causam a transição de um estado para outro. O uso deste diagrama está relacionado com a modelagem dinâmica de classes, assim como os diagramas de atividades, porém seu foco é descrever a evolução dos estados de um objeto pertencente a uma determinada classe (SILVA, 2007).

Os diagramas de estados e atividades possuem definições e focos diferentes, porém possuem uma relação dual, visto que os eventos que provocam a transição de estados no primeiro podem representar ações ou atividades no segundo.

Assim os diferentes diagramas são construídos para garantir que os mesmos estão corretos e consistentes. O uso de inspeção é uma alternativa viável, visto que os defeitos encontrados nesta fase não são propagados para as próximas fases do projeto. O custo de correção de um defeito na fase de teste é muito maior quando comparado às fases iniciais do projeto (BOEHM, 1981).

O padrão IEEE 610.12 (IEEE, 1990) define a seguinte terminologia para os seguintes conceitos que serão utilizados nesta dissertação:

- Erro: é o defeito cometido por um indivíduo ao tentar entender uma determinada informação, resolver um problema ou utilizar um método ou uma ferramenta;
- Falta: é um erro no artefato de software que pode causar diversas faltas;
- Falha: é o comportamento incorreto do software, ou seja, diferente do esperado pelo usuário, em consequência de pelo menos uma falta;

- Defeito: é o termo genérico utilizado para erro, falta ou falha.

FAGAN (1976) introduziu o conceito de inspeção de software, que permite a revisão dos artefatos e a garantia de sua qualidade. A utilização da inspeção ao longo do processo, além de representar um mecanismo eficiente e de baixo custo para identificar defeitos, proporciona ganhos em relação à produtividade e qualidade final do produto. Por isso, após a definição e elaboração dos diagramas a serem utilizados no projeto de software, os mesmos devem ser inspecionados a fim de garantir sua consistência com os requisitos do software.

A inspeção *ad-hoc* é uma das técnicas de inspeção mais conhecidas, onde o inspetor utiliza apenas o seu conhecimento para realizar esta atividade (PETERSSON, 2002). Outra técnica de inspeção conhecida na literatura técnica são os *checklists*, que consistem em uma série de perguntas que guiam o inspetor na detecção de defeitos. Um exemplo do uso de *checklists* está na técnica de inspeção por fases (KNIGHT e MEYERS, 1991), onde os mesmos são usados para detectar defeitos e avaliar características de qualidade do artefato. Existem ainda as técnicas de leitura, que oferecem uma sequência de instruções que guiam o inspetor a detectar defeitos. Por exemplo, OO-PBR (MAFRA, 2006) representa uma técnica de inspeção em requisitos, baseada em perspectiva (PBR) (SHULL *et al.*, 2000).

Os problemas identificados através da inspeção podem não representar especificamente defeitos. Em geral, inspetores tendem a indicar um conjunto de problemas que estão associados a sua interpretação do problema e, muitas vezes, sem levar em consideração a mesma perspectiva usada para a criação do artefato. Desta forma, a seguinte nomenclatura foi adotada para classificar os problemas apontados pelos inspetores em decorrência da atividade de inspeção:

- Discrepância: representa um problema detectado durante a inspeção, mas ainda não se tem certeza que o mesmo é de fato um defeito ou um falso positivo;
- Falso positivo: representa a discrepância informada como resultado da inspeção, mas que não representa um defeito de fato. Na verdade, a tentativa de sua correção no artefato vai introduzir um ou mais defeitos.

O interesse em identificar técnicas de inspeção para diagramas de estados com base em diagramas de atividades está associado ao processo de desenvolvimento que

usualmente vem sendo utilizado para construir os projetos de software contemporâneos e que prevê a inspeção dos artefatos produzidos. Por exemplo, na literatura técnica, considerando os modelos descritos em UML, podem ser encontradas técnicas de inspeção que apoiam a inspeção de alguns destes modelos ou realizam a inspeção com base neles, como por exemplo, OORTs (TRAVASSOS *et al.* 2002), uma família de técnicas de leitura para inspecionar diagramas UML e *ActCheck* (DE MELLO, 2011) um *checklist* para inspecionar a especificação de requisitos (casos de uso) descritos através de diagramas de atividades. Nenhuma destas técnicas, entretanto, apoia a inspeção de diagrama de estados com base em diagramas de atividades. Em complemento, o resultado de uma *quasi*-revisão sistemática da literatura sobre técnicas de inspeção para diagrama de fluxos de atividades revelou que existe carência de tecnologias nesta área, principalmente no que diz respeito à inspeção de diagrama de estados envolvendo comparação com diagrama de atividades.

A especificação de requisitos é fundamental para o projeto de software e, portanto, técnicas de inspeção podem ser utilizadas com o objetivo de ter os requisitos com a qualidade desejada. Portanto, assim que os requisitos são capturados, os mesmos devem ser descritos para a continuidade do projeto. Uma das maneiras de descrevê-los é através de casos de uso em forma textual utilizando um formato padrão, como citado anteriormente. Outra forma para descrever casos de uso é através de diagramas de atividades. Neste caso, a prática proposta por MASSOLLAR (2011) apoia esta atividade, onde cada diagrama de atividades representa a descrição de um caso de uso.

Estes diagramas de atividades são descritos utilizando três estereótipos: Ação do Ator, Ação do Sistema e Resposta do Sistema. Esses estereótipos estão descritos no metamodelo *UCModel*, que é implementado na ferramenta *UseCaseAgent* (MASSOLLAR, 2011), usada na construção e avaliação sintática destes diagramas de atividades. A Figura 1-1 apresenta um exemplo de caso de uso descrito através do diagrama de atividades construído utilizando a ferramenta.



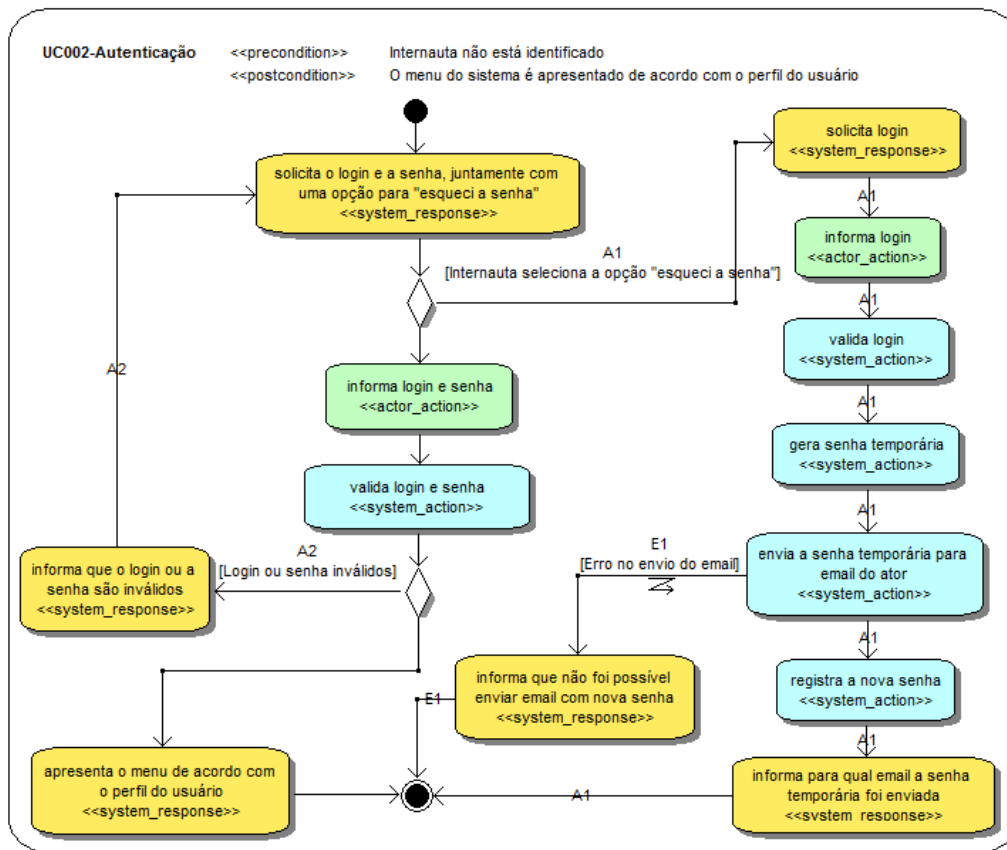


Figura 1-1 Diagrama de atividades que representa a descrição do caso de uso, conforme MASSOLLAR (2011).

A partir deste ponto, os diagramas de atividades podem ser revisados com *ActCheck* e passar para a fase seguinte do projeto, onde os outros modelos UML, incluindo diagramas de estados serão construídos. A partir deste momento, outras técnicas de inspeção para modelos UML são necessárias para garantir que os modelos produzidos estão corretos e consistentes quando comparados com os diagramas de atividade, os quais passam a representar o oráculo do projeto.

## 1.2 Objetivo da Solução Proposta

Esta dissertação tem o objetivo de apresentar *Shiô*, uma técnica de leitura para inspeção de diagramas de estados utilizando diagramas de atividades que descrevem os casos de uso do software (requisitos). A técnica complementa a família de técnicas de leitura OORTs (TRAVASSOS *et al.*2002).

O foco principal de *Shiô* é a identificação, principalmente, de defeitos semânticos (conteúdo) além de apoiar também a identificação de defeitos sintáticos (notação). A técnica orienta o inspetor a seguir uma sequência de passos, com dados de entrada e de saída bem definidos, apresentando uma série de instruções que guiam o inspetor a encontrar os defeitos. Assim, a técnica instrui os inspetores a realizar marcações nos diferentes recursos sintáticos utilizando cores pré-definidas em ambos os diagramas e, em seguida, é feita uma comparação entre os elementos coloridos nos diagramas, de forma a apoiar o inspetor na detecção dos defeitos e sua classificação de acordo com a taxonomia (TRAVASSOS, 2001): omissão, fato incorreto, ambiguidade ou informação estranha.

Os diagramas de atividades que a técnica *Shiô* utiliza devem estar modelados utilizando a proposta de MASSOLLAR (2011), que faz uso de diferentes estereótipos que classificam as ações dos diagramas. Esses estereótipos são explorados pela técnica para a comparação com os recursos sintáticos dos diagramas de estados. Os diagramas de atividades considerados por *Shiô*, e modelados conforme informado anteriormente, são consistentes com a UML 2.3 (OMG, 2010). Desta forma, esta versão da técnica não trata os recursos sintáticos tais como: eventos, nós *buffer*, nós de paralelismo e sincronização de ações, raia, nós objetos, sinais, interrupção, região de expansão e exceção que estão previstos nas versões da UML 2.3 e 2.4.1 (OMG, 2011).

Como os diagramas de atividades representam o oráculo para a inspeção, ou seja, são utilizados como referência para a técnica *Shiô*, os mesmos devem estar inspecionados para estarem consistentes com a especificação de requisitos. Assim uma alternativa é a revisão prévia dos diagramas com *ActCheck*, um checklist configurável para inspecionar os diagramas de atividades modelados de acordo com MASSOLLAR (2011).

### **1.3 Metodologia de Trabalho**

A técnica de leitura *Shiô* foi desenvolvida e avaliada com base em experimentação, seguindo um processo de desenvolvimento de tecnologias de software baseado em evidência (SHULL *et al.*, 2001; MAFRA *et al.*, 2006; SPÍNOLA *et al.*, 2008). A Figura 1-2 apresenta um resumo das atividades realizadas neste trabalho.

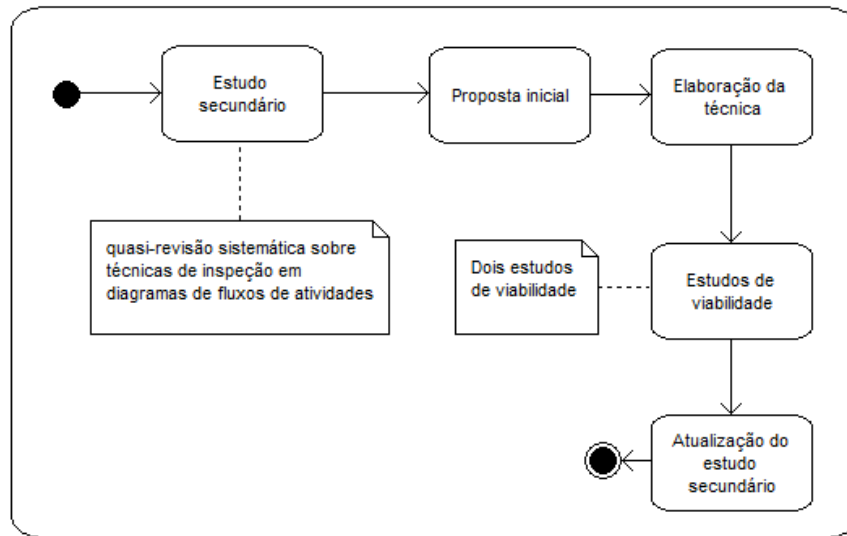


Figura 1-2 Metodologia de pesquisa adotada.

Segundo MAFRA *et al.* (2006), para que novas tecnologias de software sejam definidas com base em evidência, estudos secundários devem ser executados, que neste caso correspondem às revisões sistemáticas. Essa atividade foi realizada nesta dissertação, onde foi executada uma *quasi-revisão* sistemática para identificar técnicas de inspeção para diagramas de fluxos de atividades, onde diagramas de estados e atividades se incluem.

Os resultados obtidos no estudo secundário serviram como base para a elaboração da versão inicial da técnica de leitura. A partir deste ponto, os seguintes estudos primários devem ser executados sequencialmente: estudo de viabilidade, estudo observação, estudo de caso em ciclo de vida e estudo de caso na indústria. O estudo de viabilidade tem o objetivo de determinar a usabilidade e a eficiência da nova tecnologia. Esta atividade foi realizada nesta pesquisa, onde foram executados dois estudos de viabilidade para avaliar a técnica de leitura proposta.

Os resultados dos estudos de viabilidade apontaram que a técnica *Shiô* encontra defeitos não usualmente identificados por inspeções *ad-hoc*, porém sua aplicação depende um tempo maior do que na inspeção *ad-hoc*. Os resultados dos estudos foram usados para evoluir a técnica, visando simplificar sua descrição e reorganizar sua estrutura de aplicação.

Após estes estudos, foi realizada a atualização (reexecução) do estudo secundário, indicando a singularidade de *Shiô* no contexto de técnicas de leitura envolvendo diagramas de estados e atividades.

## 1.4 Organização do Trabalho

Esta dissertação está organizada em seis capítulos, incluindo este, que introduz o trabalho.

O Capítulo 2 trata da fundamentação teórica, apresentando os diagramas de atividades e diagramas de estados, abordando a sua aplicabilidade e os recursos sintáticos de cada diagrama utilizado pela técnica *Shiô*. O capítulo ainda trata de inspeção de software, apresentando o processo de inspeção, taxonomia de defeitos utilizada e cita algumas técnicas de inspeção em diagramas UML.

O Capítulo 3 aborda a *quasi*-revisão sistemática conduzida para identificar técnicas de inspeção relevantes em diagramas de fluxos de atividades na literatura técnica. O protocolo definido é apresentado, os termos utilizados para a busca são expostos, assim como os critérios de inclusão e exclusão. Por fim, o resultado encontrado pela *quasi*-revisão sistemática é analisado.

O capítulo 4 apresenta a técnica de leitura *Shiô* que visa inspecionar diagramas de estados utilizando os diagramas de atividades que especificam casos de uso como referência. As premissas definidas para a aplicação da técnica é apresentada, assim como a relação entre os dois diagramas utilizados na inspeção. A estrutura da técnica *Shiô* e a primeira versão da técnica são apresentadas no fim do capítulo.

O capítulo 5 apresenta os dois estudos conduzidos para avaliar a viabilidade da técnica *Shiô*. Os objetivos, o perfil dos participantes e os planejamentos dos estudos são apresentados, assim como os resultados e as análises dos dados obtidos também são analisadas. As lições aprendidas e as ameaças à validade dos estudos são apresentadas e a segunda versão da técnica *Shiô*, obtida com base nos resultados dos estudos, é apresentada no final do capítulo.

Por fim, o capítulo 6 apresenta as conclusões desta dissertação, destacando as contribuições da pesquisa e as limitações do trabalho, além das propostas de pesquisas futuras.

## 2 Fundamentação Teórica

*Este capítulo trata dos conceitos utilizados nesta dissertação, tais como: diagramas de atividades, diagramas de estados e inspeção de software. Os recursos sintáticos e a aplicabilidade de cada um dos diagramas são apresentados. Adicionalmente, o processo de inspeção, a taxonomia para classificação de defeitos e exemplos de técnicas de inspeção são descritos.*

### 2.1 Introdução

Em um projeto de software após a fase de especificação de requisitos, inicia-se a fase de *design*, consistindo na criação dos modelos. Assim, vários modelos são criados, de acordo com a necessidade de cada projeto. A UML apoia a criação de modelos, fornecendo notação específica e dividindo os diagramas em estruturais e comportamentais. A Figura 2-1 ilustra os diferentes diagramas pertencentes à UML.

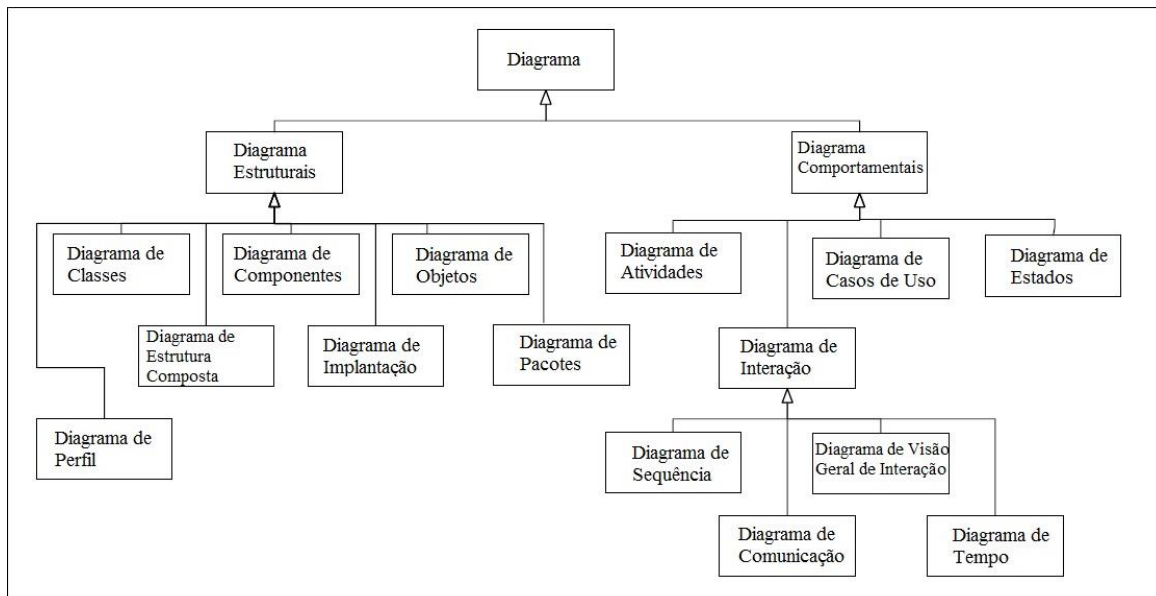


Figura 2-1 Diagramas estruturais e comportamentais da UML (Adaptado da OMG, 2011).

Cada um dos tipos de diagrama apresentados na Figura 2-1 visa representar um determinado aspecto do projeto, tendo em vista que o objetivo dos diagramas é diferente entre si. Diagramas estruturais apresentam uma visão estática da solução, permitindo visualizar, por exemplo, as classes e seus relacionamentos que não se alteram ao longo do ciclo de vida do software. Por outro lado, diagramas comportamentais, por exemplo, diagramas de atividades, estados ou sequência descrevem aspectos dinâmicos e comportamentais de um sistema de software, ou seja, os eventos e comportamentos que ocorrem ao longo do ciclo de vida do software.

Assim, após a escolha e elaboração dos diagramas que serão utilizados no projeto, os mesmos podem ser inspecionados a fim de garantir que estão capturando os requisitos corretos e que estão consistentes entre si. Inspeções de software permitem a revisão dos artefatos e a garantia de sua qualidade, além de representar um mecanismo eficiente e de baixo custo para identificar defeitos (BOEHM, 1981; LAITENBERGER e ATKINSON, 1999). Na literatura técnica existem diferentes técnicas de inspeção que objetivam apoiar a inspeção de diagramas de projetos, como por exemplo, OORTs (TRAVASSOS *et al.*, 2002) que consiste em uma família de técnicas de leitura para inspecionar diagramas UML, e OORTs/ProDeS (MARUCCI *et al.*, 2002), que apresenta um conjunto de técnicas de leitura, baseado em OORTs, que inspecionam artefatos de software pertencentes ao processo de desenvolvimento específico, denominado ProDeS/UML (COLANZI, 1999). Outro exemplo de técnica de inspeção é *ActCheck* (DE MELLO, 2011), que consiste em um checklist para apoiar a inspeção da especificação de requisitos descrita através de diagramas de atividades.

A Figura 2-2 apresenta os artefatos mais comumente utilizados durante a fase de projeto do software, incluindo a especificação de requisitos, casos de usos e diagramas da UML. Nesta figura, as linhas contínuas que ligam as caixas indicam a existência de uma técnica de leitura, pertencente à família OORTs, enquanto que a linha tracejada representa o checklist *ActCheck*. Estas técnicas serão apresentadas na seção 2.4.3.

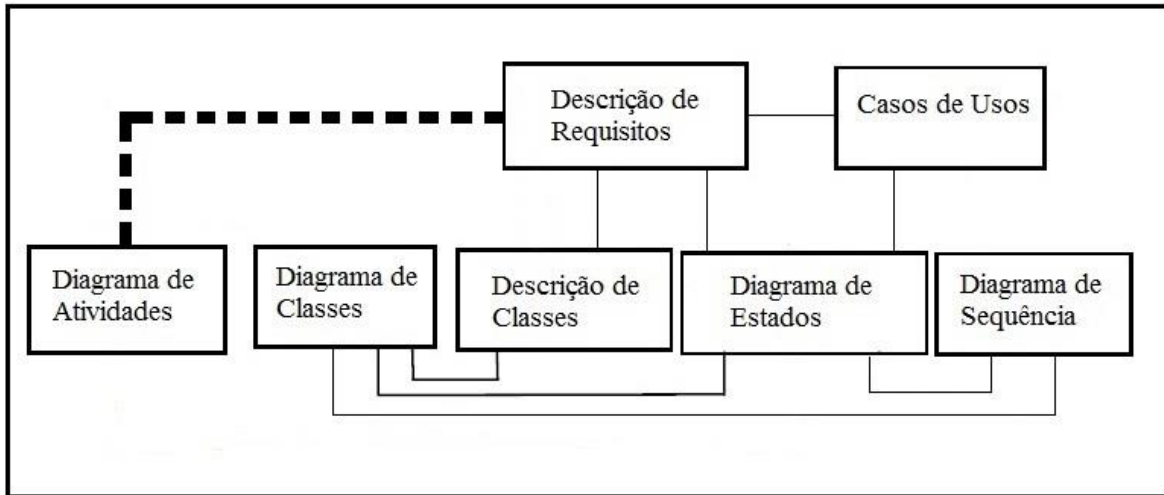


Figura 2-2 Exemplos de Técnicas de Inspeção em Diagramas da UML (Adaptado de TRAVASSOS, 2002).

O objetivo principal deste capítulo é apresentar os conceitos necessários para esta dissertação, visto que a técnica de leitura proposta inspeciona diagramas de estados utilizando como referência diagramas de atividades. Desta forma, este capítulo apresenta a aplicabilidade destes diagramas e os seus recursos sintáticos. Além disso, o conceito de inspeção também é apresentado, assim como o seu processo, a taxonomia utilizada e alguns exemplos de técnicas de inspeção.

## 2.2 Diagrama de Atividades

A UML (*Unified Modeling Language*) sofre constantes evoluções na sua documentação. Assim, a OMG (*Object Management Group*) lança novas versões no decorrer do tempo desde a sua aceitação em 1997 (OMG, 2001). A partir da versão 2 (no desenvolvimento deste trabalho na versão 2.4.1), a UML passou por uma grande reestruturação, como por exemplo, o diagrama de atividades estendeu sua capacidade para oferecer suporte à modelagem de fluxos em diversos domínios computacionais e até mesmo físicos (BOCK, 2003).

Como mostrado na Figura 2-1, a UML divide os seus modelos em duas categorias: os de representação da estrutura estática e os de representação do comportamento dinâmico dos sistemas (BOCK, 2003). Assim, os diagramas de atividades se classificam no segundo grupo, pois descrevem os aspectos dinâmicos e comportamentais de um sistema de software.

### 2.2.1 Possíveis Usos de Diagramas de Atividades

BEZERRA (2006) descreve algumas situações em que diagramas de atividades podem ser utilizados:

- Modelagem de casos de uso: o diagrama de atividades pode representar a descrição de um caso de uso, pois através dos diferentes recursos sintáticos é possível representar, por exemplo, as ações do fluxo principal, do fluxo alternativo e das exceções;
- Modelagem do processo de negócio: os diagramas de atividades podem ser utilizados na modelagem do negócio, visando identificar onde estão os pontos com problemas no fluxo de trabalho;
- Modelagem de uma operação complexa: os diagramas de atividades podem ser utilizados para detalhar alguma operação ou regra de negócio mais complexa.

### 2.2.2 Estruturas

Os diagramas de atividades possuem diversos recursos sintáticos. Nesta seção são apresentados apenas os elementos dos diagramas de atividades que serão utilizados pela técnica de inspeção proposta.

#### Ação

Uma ação representa a unidade atômica de comportamento do diagrama, aquela que não admite decomposição em outras ações, ou seja, é executada de forma ininterrupta. Sua representação é realizada através de um retângulo com os cantos arredondados com um identificador (SILVA, 2007). Uma ação pode ser conectada com outras ações através de setas que representam o fluxo de controle de uma atividade. A Figura 2-3 ilustra as ações “Enviar pagamento” e “Receber pagamento” que estão conectadas.

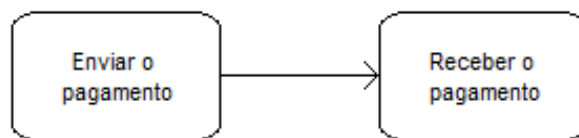


Figura 2-3 Exemplo de duas ações.



## Atividade

Uma atividade representa um elemento não atômico de modelagem de comportamento. Ela pode ser composta por ações ou por outras atividades. A sua representação, assim como na ação, se dá por um retângulo com os cantos arredondados, com um identificador e uma marca de "tridente", indicando a existência de outro diagrama detalhando as informações (SILVA, 2007). A Figura 2-4 ilustra o exemplo do diagrama de atividades "Processar Venda" composta por ações e pela atividade "Separar Produtos", detalhada em outro diagrama de atividades.

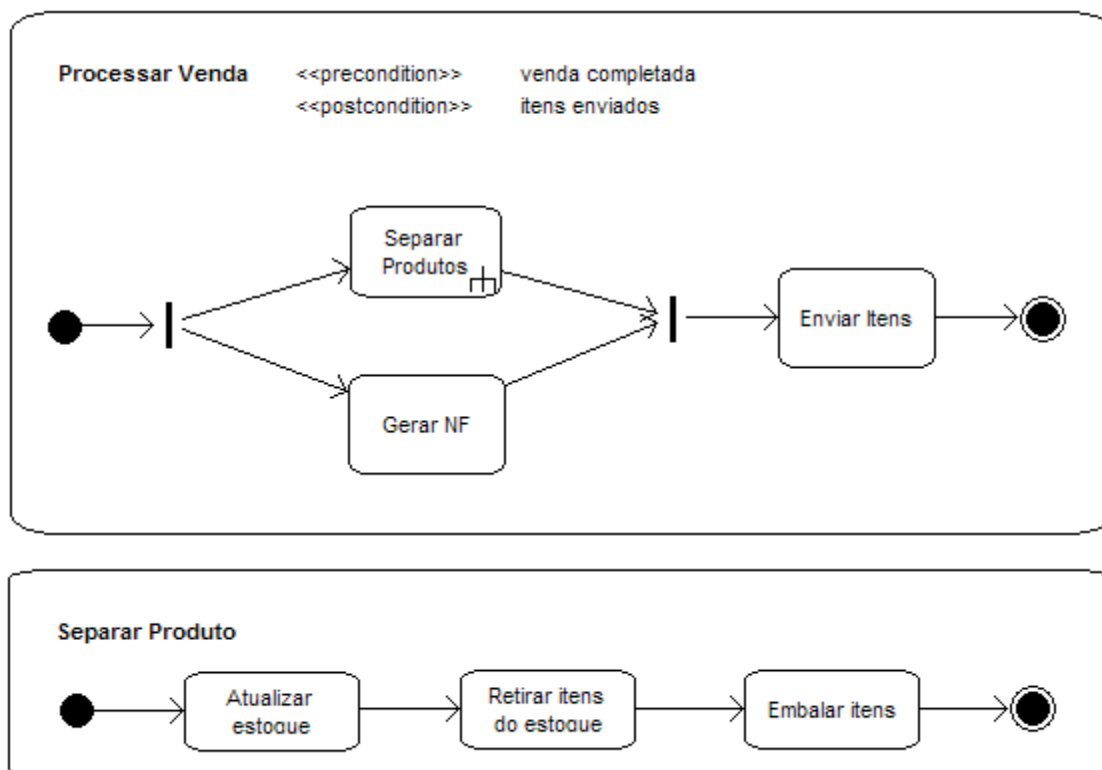


Figura 2-4 Exemplo da atividade Separar Produtos.

As atividades podem possuir pré e pós-condições, que correspondem às condições em que o sistema deve estar antes e após a execução da atividade. A representação é feita através dos estereótipos <<precondition>> e <<postcondition>> respectivamente, logo após o identificador da atividade. A Figura 2-4 ilustra um exemplo de pré e pós-condição do diagrama de atividades "Processar Venda".

## Fluxo de Controle

O diagrama de atividades possui recursos sintáticos para representar diferentes fluxos de controle, tais como (SILVA, 2007):

- **Nó inicial:** representa o início da execução da atividade. Um diagrama pode apresentar mais de um nó inicial. A sua representação é feita através de um círculo totalmente preenchido, ilustrado na Figura 2-5;
- **Nó final da atividade:** representa o final da execução da atividade. Um diagrama pode apresentar mais de um nó final. A sua representação é feita através de um círculo preenchido com borda dupla, ilustrado pela Figura 2-5;

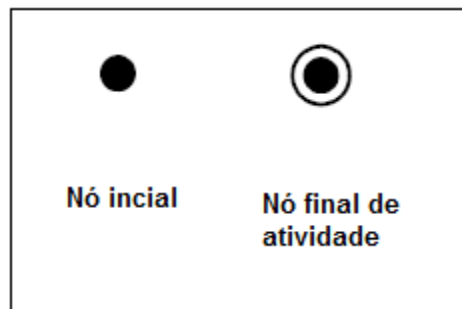


Figura 2-5 Nós inicial e final de atividade.

- **Nó de decisão:** representa a possibilidade de existirem fluxos alternativos, condicionados por expressões. Este nós equivalem ao conceito de *if, if-else* e *switch* das linguagens de programação. A sua representação é feita através de um losango, contendo uma entrada e diversas saídas, onde cada saída possui uma expressão, chamada de condição de guarda, representada entre colchetes. A condição de guarda indica o que deve ocorrer para mudar o fluxo de execução do caso de uso. A Figura 2-6 ilustra o nó de decisão e as condições de guarda "Pedido Aceito" e "Pedido Rejeitado", que estão associadas a diferentes fluxos.

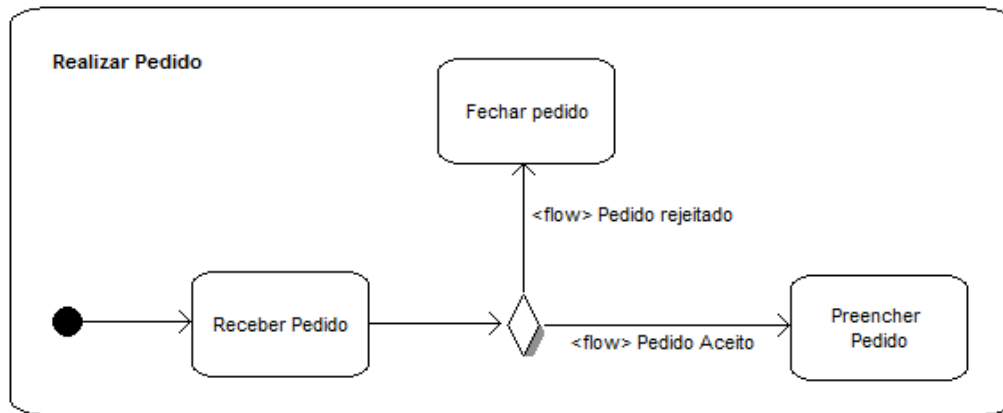


Figura 2-6 Nó de decisão (Adaptado da OMG, 2011).

## 2.3 Diagrama de Estados

Pode ser feita uma analogia entre as representações do diagrama de estados com os objetos do mundo real. Por exemplo, uma jarra está cheia de suco (poderia estar vazia) ou que uma pessoa está cansada (poderia estar descansada). A partir destes exemplos é possível observar que os objetos estão alterando os seus estados quando acontece algum evento, seja ele interno ou externo. Assim, quando um objeto muda de um estado para outro, diz-se que ele realizou uma transição entre estados e que essas transições e todos os estados do objeto constituem o ciclo de vida do objeto. Essa analogia aplicada aos objetos do mundo real pode ser aplicada aos objetos de uma classe, modelada por diagramas de classes (BEZERRA, 2006). Assim, os estados (valores dos atributos) correspondem ao nome dado a cada situação da classe-objeto e as transições se referem às mudanças de um estado para outro (SILVA, 2007).

Desta forma, o diagrama de estados da UML (OMG, 2011) permite descrever o ciclo de vida dos objetos de uma classe, apresentando os eventos que causam a transição de um estado para outro. A notação inicial do diagrama foi proposta por David Harel através de seu trabalho com máquinas de estados finitos, posteriormente estendida por Jim Rumbaugh, Mealy e Moore. (BEZERRA, 2006).

O uso do diagrama de estados normalmente está voltado para a modelagem dinâmica de classes, assim como os diagramas de atividades, porém o foco deste diagrama é descrever a evolução de estados de um objeto da classe (SILVA, 2007).

### 2.3.1 Estruturas

A especificação da UML apresenta diversos recursos sintáticos para os diagramas de estados, porém a mesma abordagem adotada para descrever as características de diagramas de atividades será utilizada nesta seção, onde serão apresentados apenas os elementos que estão presentes nos diagramas de estados e que serão utilizados pela técnica de inspeção proposta.

#### Estado

Um estado corresponde a uma situação em que um objeto se encontra durante o seu ciclo de vida. Assim, um estado pode ser uma situação na qual o objeto permanece até que ocorra um evento que o faça sair dele ou a execução de uma tarefa, sendo que o final da execução acarretaria a saída da situação.

A representação de estado simples é um retângulo de cantos arredondados, com um identificador que corresponde ao nome do estado. A Figura 2-7 ilustra um exemplo de um estado.

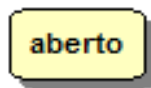


Figura 2-7 Exemplo de estado.

Os estados de um objeto podem variar de acordo com o contexto em que está sendo modelado. Por exemplo, o objeto automóvel no contexto de movimento pode se comportar conforme ilustrado na Figura 2-8. Supondo agora o mesmo objeto automóvel no contexto de uma oficina mecânica (Figura 2-9), os estados neste contexto são totalmente diferentes do diagrama anterior. Assim, um mesmo objeto automóvel pode assumir estados diferentes de acordo com o contexto com o qual o modelo foi construído (SILVA, 2007).

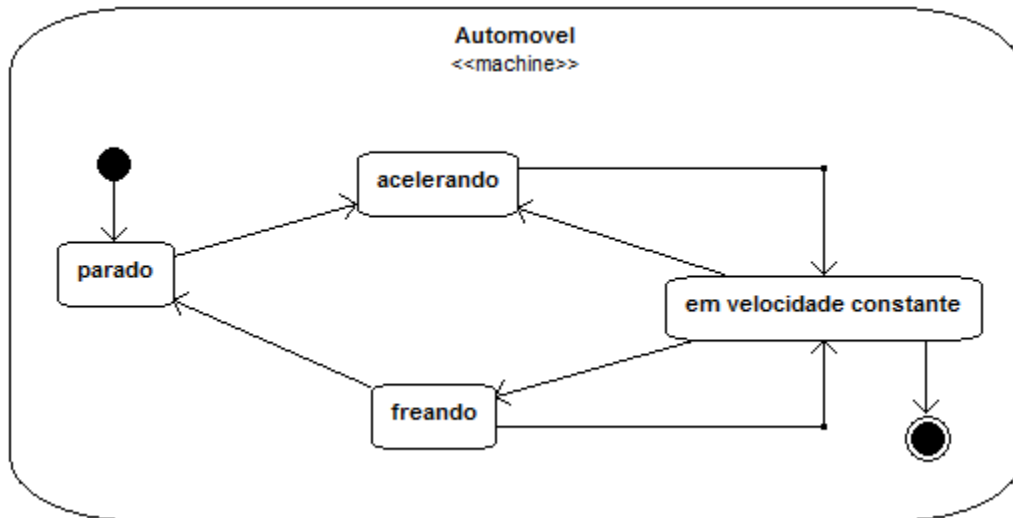


Figura 2-8 Diagrama de estados para Automóvel, tendo como foco o movimento (Adaptado de SILVA, 2007).

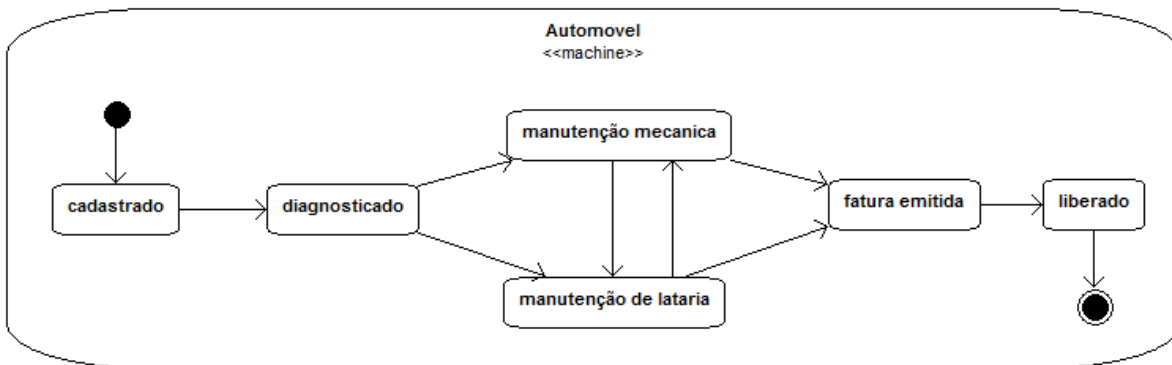


Figura 2-9 Diagrama de estados para Automóvel no contexto de uma oficina mecânica (Adaptado de SILVA, 2007).

Os diagramas de estados, assim como os diagramas de atividades, possuem estado inicial e estado final. No diagrama de estados deve existir apenas um estado inicial e a sua representação é feita através de um círculo fechado. O estado final é representado através de um círculo fechado com uma borda dupla e pode existir mais de um estado final em um diagrama de estados (BEZERRA, 2006). A Figura 2-10 (a) ilustra o estado inicial e a Figura 2-10 (b) o estado final. A representação de estado inicial e final do diagrama de estados é igual a representação do diagrama de atividades.

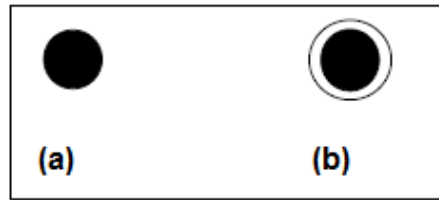


Figura 2-10 (a) Estado inicial (b) Estado final.

### Transição de Estado

Os estados estão associados a outros pelas transições, que são representadas como uma linha direcionada conectando os estados. Quando ocorre uma transição entre estados, diz-se que a transição foi disparada. Vale ressaltar que transição para o mesmo estado é válida. Detalhes de uma transição são descritos através do formato *evento [guarda]*. Assim, tem-se que (BEZERRA, 2006):

- **Eventos:** uma transição possui um evento associado e este provoca a transição do estado. A Figura 2-11 ilustra o diagrama de estados de "Venda" e apresenta as transições com os eventos "Recebendo produtos", "Pagamento [dinheiro]", "Pagamento [cartão de crédito]", "Saldo insuficiente no cartão de crédito" e "Saldo suficiente no cartão de crédito".
- **Condição de guarda:** expressão que resulta em um valor lógico (verdadeiro ou falso). Assim, o evento associado à condição de guarda só ocorre quando a condição for verdadeira. Vale ressaltar que é importante não confundir condição de guarda de uma transição com um evento de mudança, que também é definido através de uma expressão de valor lógico. A diferença está no fato da expressão da condição de guarda estar sempre apresentada entre colchetes. A Figura 2-11 ilustra dois eventos com diferentes condições de guarda: "[dinheiro]" e "[cartão de crédito]". O objeto no estado "Aguardando Pagamento" pode ir para o estado "Autorizado o Pagamento" caso o evento "Pagamento" com a condição de guarda [dinheiro] seja verdadeira. Outra possibilidade é a transição entre o estado "Aguardando Pagamento" para "Aguardando Autorização do Cartão de Crédito", caso o evento de "Pagamento" ocorra com a condição de guarda [cartão de crédito] sendo verdade.

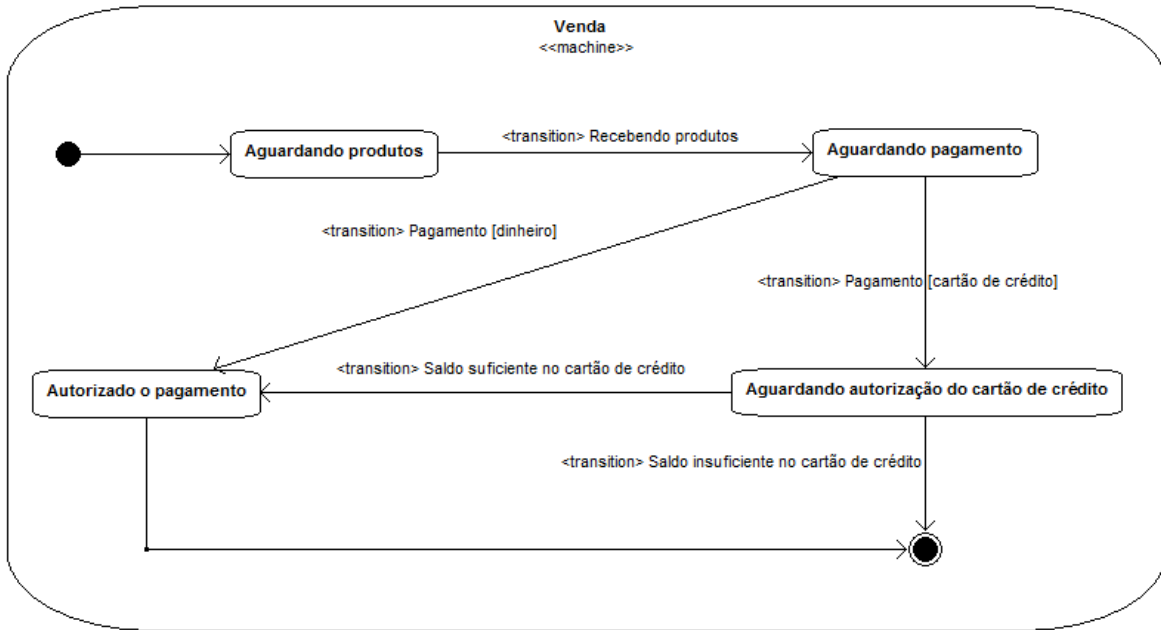


Figura 2-11 Diagrama de estados de Venda.

### Estados Compostos

Segundo SILVA (2007) a utilização dos estados compostos está relacionada à descrição de um diagrama de estados mais complexo e, portanto, com maiores detalhes. Assim tomando o exemplo do automóvel da **Erro! Fonte de referência não encontrada.**, este mesmo diagrama poderia ser modelado como ilustra a Figura 2-12, onde existe o superestado “em movimento” que contém os subestados “acelerando”, “freando” e “em velocidade constante”. A Figura 2-13 representa um modelo alternativo ao da Figura 2-12, onde são acrescentados um estado final e um pseudo-estado inicial no superestado. Entretanto, note que ambos os modelos possuem o mesmo significado.

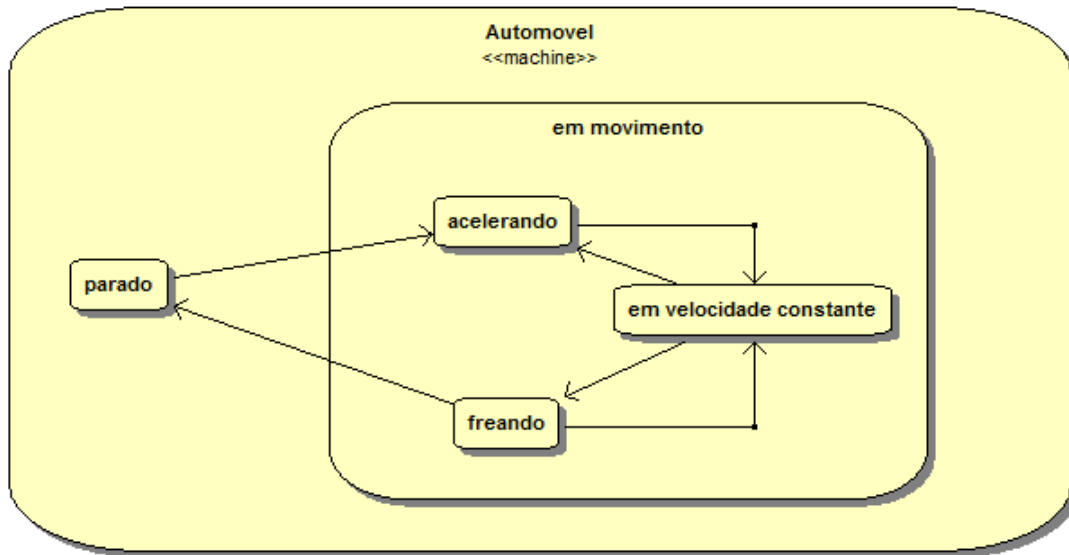


Figura 2-12 Modelagem de estados para Automóvel, com estado composto (Adaptado de SILVA, 2007).

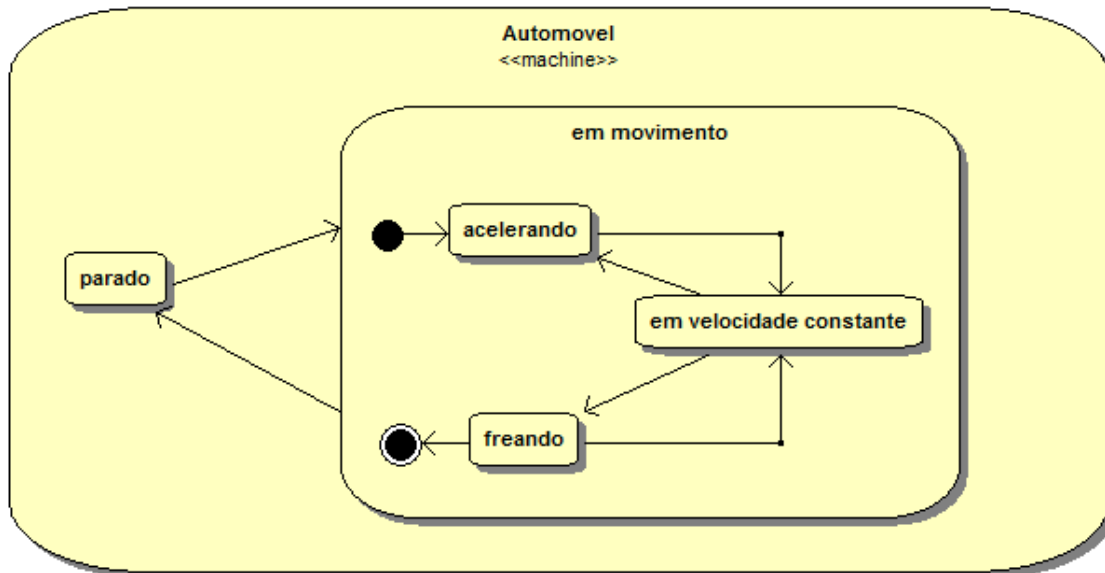


Figura 2-13 Modelagem alternativa para Automóvel, com estado composto (Adaptado de SILVA, 2007).

### Submáquina

O uso de submáquinas está relacionado também à modelagem de sistemas mais complexos, onde a utilização de estados compostos resulta em um diagrama com muitos elementos, prejudicando a legibilidade do diagrama. Assim, o uso da submáquina surge



como solução para este problema, onde os subestados do superestado correspondem a um novo diagrama e o superestado corresponde a submáquina no diagrama de estados (SILVA, 2007).

A Figura 2-14 apresenta a submáquina "em movimento" do objeto Automóvel. Esta modelagem supõe a existência de outro diagrama de estados cujo conteúdo deve ser exatamente o conjunto de subestados de "em movimento" presente na Figura 2-13.

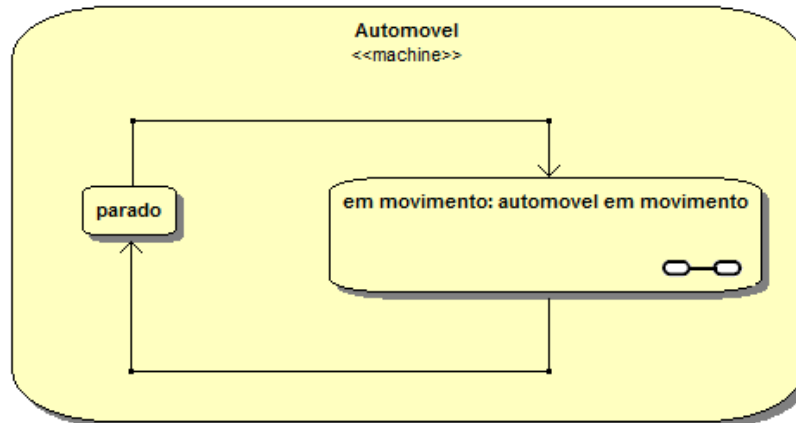


Figura 2-14 Modelagem de Automóvel com estado submáquina (SILVA, 2007).

## 2.4 Inspeção de Software

O conceito de inspeção de software foi introduzido por (FAGAN, 1976) sendo um tipo particular de revisão de software. Seu principal objetivo é encontrar defeitos no artefato que está sendo inspecionado. Inspeções são consideradas atividades de garantia da qualidade, permitindo que defeitos sejam detectados nas fases iniciais do desenvolvimento e, com isso, evitando sua propagação para fases posteriores do projeto, reduzindo o custo com as atividades de teste e, por consequência, reduzindo o custo total de desenvolvimento (PETERSON, 2002).

Inspeção de software é uma atividade recomendada para a melhoria da qualidade e traz diversos benefícios para um projeto, tais como: redução de esforço (CONRADI *et al.*, 1999), redução do tempo (GILB e GRAHAM, 1993), redução de custo (LAITENBERGER e ATKINSON, 1999) e aumento da produtividade (GILB e GRAHAM, 1993).

Outra forma de detectar defeitos em um sistema é através das falhas reveladas pelos testes, porém os testes ocorrem em um momento posterior às inspeções, quando já

se tem grande parte do produto "pronto". Estudos, como de BASILI e SELBY (1987), mostram que os defeitos encontrados por testes e inspeções são de tipos diferentes. Assim, a aplicação de inspeções e testes em um sistema se apresenta como uma abordagem necessária para garantir a entrega do produto desejado com qualidade.

A especificação de requisitos é muito importante em um projeto de software, pois ela descreve todas as funcionalidades do software a ser entregue e fornece uma referência para a validação do produto final. Assim, ter a especificação de requisitos inspecionada e com alta qualidade é um pré-requisito fundamental para um software de alta qualidade (JALOTE, 2005).

Entende-se que inspeções não devem ser realizadas pelo próprio autor do documento, ou seja, pessoas não envolvidas com a criação do documento a ser inspecionado devem realizar a inspeção. Com isso, evita-se o viés do autor em considerar que o artefato produzido está totalmente coerente com a especificação e livre de defeitos.

#### 2.4.1 Processo de Inspeção

Na literatura técnica existem diferentes processos de inspeção com base no processo original proposto por FAGAN (1976). Um exemplo é o processo apresentado por WONG (2006), composto por seis atividades ilustradas na Figura 2-15.

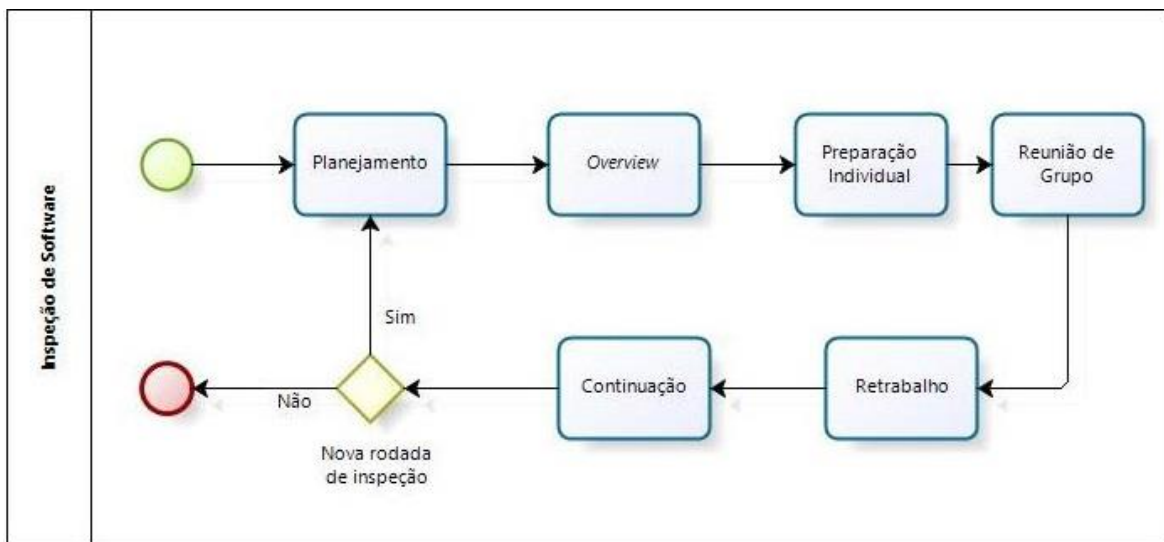


Figura 2-15 Processo de inspeção de software de FAGAN (de acordo com WONG, 2006).

Cada uma das atividades do processo da Figura 2-15 é descrita como:

- **Planejamento:** organiza e prepara o processo de inspeção, envolvendo tarefas tais como: preparar os artefatos a serem inspecionados, escolher a técnica que será utilizada durante a inspeção, agendar a reunião e selecionar os inspetores e definir seus papéis;
- **Overview:** atividade opcional e recomendada quando o artefato a ser inspecionado é complexo, ou seja, difícil de entender. Nesta atividade, o autor do artefato a ser inspecionado oferece um treinamento aos inspetores para que possam se familiarizar com o documento e o sistema como um todo;
- **Preparação Individual:** os participantes da inspeção tentam compreender o artefato individualmente antes da fase de reunião, realizando sua leitura e indicando as discrepâncias. Segundo WONG (2006), estudos evidenciam que conduzir esta atividade é custoso e que a mesma não é necessária quando a inspeção é realizada individualmente;
- **Reunião de grupo:** o objetivo principal desta atividade é detectar defeitos. As listas de discrepâncias são discutidas e classificadas. Assim como a atividade de preparação individual, esta também é custosa para o projeto. WONG (2006) relata diversos estudos que identificaram uma maior eficiência na detecção de defeitos quando esta é realizada em pequenos grupos ou individualmente, visto que reuniões com uma quantidade muito grande de participantes acarretam em uma dispersão da atenção dos participantes;
- **Retrabalho:** o autor do artefato inspecionado recebe a lista de defeitos e os corrige;
- **Continuação:** avalia a qualidade das correções realizadas no artefato inspecionado. Neste momento o moderador decide se uma inspeção será necessária. Métricas tais como: número total de defeitos, tempo dedicado a preparação individual e a reunião de inspeção e o número de inspetores envolvidos, podem ser usadas para apoiar esta decisão.

## 2.4.2 Taxonomia de defeitos

Na literatura técnica existem diversas taxonomias para classificação de defeitos, diferindo entre si em relação à quantidade de categorias e seus detalhamentos. A taxonomia utilizada nesta dissertação segue, por conveniência, a classificação adotada em TRAVASSOS (2001), que apresenta um conjunto de classes genéricas de defeitos não mutuamente exclusivas, ou seja, um defeito pode pertencer a mais de uma categoria.

Os tipos de defeitos variam de acordo com as necessidades de cada projeto, além de variar de acordo com o artefato inspecionado. A Tabela 2-1 descreve cada classe de defeito e apresenta um exemplo do tipo de defeito em questão.

Tabela 2-1 Tipos de defeitos de software (Adaptado de TRAVASSOS, 2001).

<b>Categoria</b>	<b>Descrição</b>	<b>Exemplo</b>
Omissão	Informações necessárias sobre o sistema que foram omitidas do artefato de software	Alguns requisitos importantes do projeto não foram incluídos.
Fato Incorreto	Algumas informações no artefato de software contradizem as informações presentes na especificação de requisitos ou o conhecimento geral do domínio	Um requisito descreve um fato que não é verdadeiro, considerando o contexto para qual o sistema está sendo desenvolvido.
Inconsistência	Algumas informações no artefato de software estão inconsistentes com outras no artefato.	Dois ou mais requisitos são conflitantes.
Ambiguidade	As informações no artefato de software são ambíguas, sendo possível ao desenvolvedor interpretá-las de diferentes maneiras, podendo levar a uma implementação incorreta.	Um requisito tem várias interpretações devido à diferentes termos utilizados para uma mesma característica ou vários significados de um mesmo termo para um contexto particular.
Informação estranha	As informações fornecidas não são necessárias ou nem usadas. São informações adicionais inseridas sem necessidade.	As informações contidas nos requisitos não fazem parte do contexto para qual o sistema está sendo desenvolvido.

## 2.4.3 Exemplos de Técnicas de Inspeção

A técnica de inspeção escolhida no processo de inspeção influencia diretamente no resultado das inspeções. A inspeção pode ser realizada de forma *ad-hoc*, ou seja, quando nenhum tipo de técnica ou heurística é fornecida ao inspetor. Este tipo de inspeção utiliza apenas o conhecimento e experiência do inspetor e é também conhecida como “leitura sem técnica” (PETERSSON, 2002). Assim, para que a inspeção *ad-hoc*

tenha um bom resultado, é necessário que os inspetores mais experientes realizem a atividade de inspeção. Entretanto, a alocação de profissionais com alto grau de experiência eleva o custo do projeto, quando comparado com a utilização de profissionais com menos experiência.

Outra forma de inspecionar os artefatos de software é através do uso de *checklists*, que consistem em uma série de questões que o inspetor deve considerar durante a inspeção, respondendo "sim" ou "não" para cada uma delas (LAITENBERGER *et al.*, 2001). A inspeção utilizando *checklist* é classificada de forma distinta por diferentes autores. Por exemplo, SHULL *et al.* (2000) consideram o *checklist* uma técnica de leitura parcialmente sistemática e não focada. Já PORTER *et al.* (1995) classificam a inspeção utilizando *checklist* e *ad-hoc* como técnicas de leituras não sistemáticas. O uso de *checklists* nas inspeções tende a auxiliar os inspetores menos experientes nesta tarefa, diferente da inspeção *ad-hoc* que é totalmente dependente da experiência do inspetor.

Técnica de leitura é outra abordagem conhecida na literatura técnica para inspecionar artefatos de software. Segundo SHULL *et al.* (2003), as técnicas de leitura consistem em um conjunto de instruções que orientam o inspetor a refletir sobre a informação discrepante, apoiando assim a detecção de defeitos no artefato. Este tipo de inspeção aumenta a eficiência dos inspetores (TRAVASSOS, 2001) e, como com *checklists*, inspetores menos experientes tendem a obter melhor desempenho quando comparado com inspeções *ad-hoc*. Diferente da inspeção com *checklist* e *ad-hoc*, as técnicas de leitura possuem um comportamento sistemático e intuitivo (HE e CARVER, 2006). É possível encontrar na literatura técnica diferentes técnicas de leitura, tais como:

- UBR (*Usage-Based Reading*): a ideia principal é focar o esforço da inspeção nos defeitos mais críticos do artefato inspecionado. Este tipo de técnica utiliza o ponto de vista do usuário para identificar os defeitos que ele julga ter maior impacto negativo no sistema (THELIN *et al.*, 2001);
- DBR (*Defect-Based Reading*): esta família de técnicas visa a inspeção de documentos de requisitos, onde cenários são descritos para apoiar a identificação de tipos específicos de defeitos (PORTER e VOTTA, 1994; PORTER *et al.*, 1995).
- PBR (*Perspective-Based Reading*): representa uma família de técnicas de leitura contendo diferentes perspectivas: projetista, usuário e testador. Assim, para cada

uma das perspectivas, PBR assegura que o inspetor avaliará o documento de requisitos seguindo este ponto de vista. As técnicas orientam durante a inspeção a criação de um documento de apoio representando a abstração de cada perspectiva: diagramas de classes, casos de uso e conjunto de casos de teste, para as perspectivas de projetista, usuário e testador, respectivamente (SHULL *et al.*, 2000).

- MBR (*Metric-Based Reading*): técnica de leitura que visa detectar defeitos em modelos de casos de uso. A técnica assume a premissa de que os valores de algumas métricas e casos de uso podem ser vistos como potenciais indicadores de defeitos. Assim, cada métrica apresenta uma faixa de valores nos quais os valores fora da faixa definida seriam um indicativo de maior probabilidade de ocorrência de defeitos nos modelos de casos de uso (BERNARDEZ *et al.*, 2004).

Diversas técnicas de inspeção foram desenvolvidas utilizando os conceitos de *checklist* e de técnicas de leitura envolvendo os artefatos de projeto. A seguir, são apresentados alguns exemplos de técnicas de inspeção existentes na literatura técnica. Estas técnicas utilizam de alguma forma os diagramas da UML (OMG, 2011) na inspeção. As técnicas apresentadas são duas técnicas de leitura e um checklist, denominadas: OORTs (TRAVASSOS *et al.*, 2002), OORTs/ProDeS (MARUCCI *et al.*, 2002) e *ActCheck* (DE MELLO, 2011).

## **OORTs**

OORTs (*Object-Oriented Reading Techniques*) é uma família de técnicas de leitura, desenvolvida por (TRAVASSOS *et al.*, 1999), que possui o propósito de detectar defeitos nos diagramas OO (Orientados a Objeto) utilizando a notação da UML. OORTs possui sete diferentes técnicas de leitura que apoiam a inspeção dos diagramas de classes, sequência e estados em comparação com a especificação de requisitos e descrição de casos de uso.

OORTs possui dois níveis de inspeção: a inspeção horizontal (verificação) e a inspeção vertical (validação). O primeiro nível de inspeção consiste em comparar documentos na mesma fase do projeto, com o objetivo de identificar se os diagramas descrevem um mesmo sistema, enquanto que a inspeção vertical inspeciona artefatos de diferentes fases do projeto, com o objetivo de verificar se os diagramas são válidos em

relação à especificação dos requisitos e os casos de uso. A Tabela 2-2 apresenta as técnicas pertencentes à OORTs, definindo quais são os artefatos inspecionados e a orientação de cada uma das técnicas.

Tabela 2-2 Técnicas de Leitura OORTs (Adaptado de TRAVASSOS, 2002).

Artefatos inspecionados	Nível
Diagrama de Sequência x Classes	Horizontal
Diagrama de estados x Descrição de Classes	Horizontal
Diagrama de Sequência x Diagrama de Estados	Horizontal
Diagrama de Classes x Descrição de Classes	Horizontal
Descrição de Classes x Descrição de Requisitos	Vertical
Diagrama de Sequência x Casos de Uso	Vertical
Diagrama de Estados x Descrição de Requisitos e Casos de Uso	Vertical

A estrutura das técnicas de leitura é semelhante, onde cada uma das técnicas apresenta o objetivo e uma sequência de passos para orientar o inspetor a realizar a inspeção. As entradas e saídas de cada passo são descritas, além de fornecer uma série de instruções que orientam o preenchimento de um relatório de discrepância, ajustado para cada nível de leitura.

### OORTs/ProDeS

OORTs/ProDeS (MARUCCI *et al.*, 2002) representa um conjunto de técnicas de leitura que visam inspecionar artefatos de software pertencentes ao processo de desenvolvimento específico, denominado ProDeS/UML (COLANZI, 1999). Este processo é baseado no método *Fusion* (COLEMAN *et al.*, 1994), utiliza a notação da UML nos artefatos gerados e define atividades de testes ao longo de todas as fases do desenvolvimento. ProDeS/UML é composto de 4 fases: engenharia de requisitos, análise, projeto e implementação.

Um estudo de viabilidade foi conduzido com o intuito de verificar a aplicação de OORTs sobre os artefatos gerados pelo ProDeS/UML. Assim, foi possível observar que os artefatos da fase de projeto foram todos praticamente cobertos pelas técnicas de OORTs, mas os artefatos presentes nas fases de engenharia de requisitos e de análise careciam de técnicas de leitura.

Os autores propuseram então um conjunto de oito técnicas de leitura para os documentos gerados nestas outras fases do processo de desenvolvimento. As técnicas de leitura foram baseadas em OORTs e, portanto, possuem uma estrutura muito similar à OORTs. A Tabela 2-3 apresenta o conjunto de técnicas de leitura criadas, indicando a fase do processo de desenvolvimento, os documentos inspecionados, o tipo e a orientação de cada uma das técnicas.

Tabela 2-3 Técnicas de Leitura para apoiar o processo de desenvolvimento ProDeS/UML  
(Adaptado de MARUCCI *et al.*, 2002).

Técnica	Fase do Processo	Artefatos Inspeccionados	Tipo	Nível
ER1	Engenharia de Requisitos	Diagrama de Casos de Uso x Especificação dos Casos de Uso x Documento de Requisitos	Validação	Vertical
ER2	Engenharia de Requisitos	Diagrama de Classes do Domínio x Diagrama de Casos de Uso x Especificação dos Casos de Uso	Verificação	Horizontal
A1	Análise	Diagrama de Classes da Análise x Diagrama de Classes do Domínio x Documento de Requisitos	Validação	Vertical
A2	Análise	Modelo de Operações x Especificação dos Casos de Uso	Verificação	Vertical
A3	Análise	Diagrama de Classes da Análise x Modelo de Operações	Verificação	Horizontal
A4	Análise	Modelo de Operações x Modelo de Ciclo de Vida x Documento de Requisitos	Validação	Vertical
P8	Projeto	Diagrama de Classes da Análise Refinado x Diagrama de Visibilidade x Diagrama de Classes da Análise x Documento de Requisitos	Validação	Vertical
P9	Projeto	Descrição de Classes x Modelo de Operações	Verificação	Vertical

### **ActCheck**

Este *checklist* para inspecionar requisitos descritos em diagramas de atividades (DE MELLO, 2011) foi criado com o objetivo de detectar defeitos em diagramas de atividades que utilizam a abordagem de MASSOLLAR (2011). Assume-se que a especificação dos requisitos já foi devidamente inspecionada. A primeira versão de *ActCheck* contém dois *checklists* configuráveis a partir de um questionário de



caracterização da aplicação. Assim, os *checklists* para cada projeto só conterão questões pertinentes, ou seja, são retiradas as questões que não fazem parte do escopo do projeto.

Os *checklists* da técnica são denominados *checklist A* e *checklist B*. O primeiro contém itens de avaliação específicos para verificação da rastreabilidade entre os diagramas de atividades e a descrição textual dos requisitos. O outro, *checklist B*, deve ser aplicado após o *checklist A* e tem como objetivo verificar a consistência interna de um diagrama de atividades e a sua relação com os outros diagramas de atividades referentes a mesma especificação de requisitos.

## 2.5 Conclusão

Inspeções de software são atividades importantes na detecção de defeitos, principalmente por detectar defeitos na fase inicial do projeto, evitando que os defeitos se propaguem para fases posteriores do desenvolvimento. Caso esses defeitos fossem propagados, o custo para corrigi-los na fase de teste seria muito maior quando comparado ao custo para corrigir durante a fase de *design*.

Os dois diagramas da UML descritos neste capítulo, diagrama de atividades e diagrama de estados, possuem recursos sintáticos semelhantes em simbologia, porém com foco distinto, visto que o diagrama de estados tem por finalidade modelar o ciclo de vida dos objetos do sistema enquanto que o diagrama de atividades visa modelar as atividades realizadas pelo sistema, ou seja, o comportamento dinâmico. Este capítulo apresentou a sua aplicabilidade e os recursos sintáticos utilizados na técnica de inspeção proposta nesta dissertação.

O Capítulo 3 trata de uma revisão *quasi*-sistemática da literatura técnica que foi conduzida com o intuito de identificar técnicas de inspeção aplicáveis a diagramas de fluxos de atividades em geral, onde diagramas de atividades e, conseqüentemente, diagramas de estados se incluem.

Existe uma dualidade entre os diagramas de atividades e os diagramas de estados, visto que os eventos que provocam a transição de estados representam ações ou atividades nos diagramas de atividades. Essa relação semântica entre os recursos sintáticos dos diagramas de atividades e o diagramas de estados serão apresentados no Capítulo 4, que trata propriamente da técnica de leitura proposta e apresenta a sua primeira versão.

As inspeções possuem um processo bem definido, com diferentes papéis envolvidos em diferentes atividades. O processo que foi descrito neste capítulo não é

único, mas serve como referência para que organizações criem seus próprios processos, de acordo com as suas necessidades. Assim, para que a organização obtenha todos os benefícios da inspeção, a mesma deve ser aplicada de forma sistemática, seguindo o seu processo definido.

## 3 Técnicas de Inspeção para Diagramas de Fluxos de Atividades

*Neste capítulo é apresentada uma quasi-revisão sistemática da literatura técnica conduzida para identificar técnicas de inspeção em diagramas que representem fluxos de atividades.*

### 3.1 Introdução

Este capítulo apresenta o protocolo e as considerações sobre os resultados de uma *quasi-revisão* sistemática conduzida em 2011 e atualizada em 2013, com o objetivo de identificar trabalhos relevantes sobre técnicas de inspeção para diagramas de fluxos de atividades. O termo fluxos de atividades foi utilizado para definir de maneira ampla os diagramas que representem de diferentes formas os fluxos de atividades, tais como diagramas de atividades, BPMN, SPEM, diagramas de sequência e entre outros. Os diagramas de estados são também contemplados com os resultados deste estudo, devido à dualidade existente entre os diagramas de atividades e os diagramas de estados, pois os eventos que provocam a transição de estados representam ações ou atividades nos diagramas de atividades. Este estudo está em conformidade com a abordagem para definição de novas tecnologias de software baseada em evidência (MAFRA *et al.*, 2006).

Uma revisão sistemática da literatura (estudo secundário) tem como objetivo reunir a maior quantidade de material bibliográfico disponível e relevante sobre um assunto específico e prover evidência a respeito deste assunto (KITCHENHAM, 2004). Assim, seu propósito é identificar, avaliar e interpretar os trabalhos relevantes na literatura técnica sobre um determinado tópico de interesse. Diferente da revisão tradicional da literatura, a revisão sistemática segue uma estratégia de pesquisa formal representada por um protocolo de revisão (TRAVASSOS *et al.*, 2008), permitindo que o mesmo possa ser reexecutado e, conseqüentemente, auditado por outros pesquisadores. O protocolo da revisão sistemática é um documento que especifica a questão de pesquisa central e define a forma em que a mesma será conduzida, explicitando os critérios de inclusão e de exclusão, as informações que devem ser extraídas de cada estudo selecionado, dentre outras informações.

Uma *quasi*-revisão sistemática (TRAVASSOS *et al.*, 2008) se distingue da revisão sistemática tradicional devido a seu objetivo principal ser realizar uma caracterização da área (quando não existe objeto para comparação), porém seguindo um protocolo de investigação com o mesmo rigor apresentado por uma revisão sistemática. Assim, a estrutura utilizada neste estudo foi baseada em (BIOLCHINI, 2005). As seções a seguir apresentam o protocolo de pesquisa completo e os resultados obtidos.

### 3.2 Planejamento da *quasi*-Revisão Sistemática

O objetivo deste estudo secundário foi identificar as técnicas de inspeção existentes e aplicáveis a diagramas de fluxos de atividades utilizados em projetos de software. Em uma busca *ad-hoc* por técnicas de leitura em diagramas de atividades, utilizando a máquina de busca Scopus e a expressão “reading technique” AND “activity diagram”, nenhum resultado relevante foi encontrado. Assim este estudo visa encontrar diferentes técnicas de inspeção, se possível técnicas de leitura, para diferentes tipos de diagramas de fluxos de atividades. Vale ressaltar que técnicas de verificação que visam a consistência entre diagramas com base em pre-especificações (*model checking*) não fazem parte do escopo desta *quasi*-revisão sistemática.

A questão principal de pesquisa descrita no protocolo de revisão é: Quais são as técnicas de inspeção aplicáveis aos modelos de fluxos de atividades em projeto de software?

Uma revisão informal na literatura foi realizada para identificar possíveis artigos de controle, onde foram encontrados apenas dois estudos relevantes, (TANRIÖVER e BILGEN, 2007) e (DE MELLO *et al.*, 2010). Vale ressaltar que o estudo de TANRIÖVER e BILGEN (2007) foi utilizado como artigo de controle no trabalho de (DE MELLO *et al.*, 2010) referente ao desenvolvimento de *ActCheck*. Detalhes sobre estes dois artigos são descritos na seção 3.4.

Para responder a questão principal de pesquisa, a abordagem PICO (PAI *et al.*, 2004) foi utilizada para a organização da *string* de busca, que consiste na divisão da *string* em 4 dimensões: População, Intervenção, Comparação e Resultados (*Outcomes*). Por se tratar de uma *quasi*-revisão sistemática, a parte referente à Comparação é vazia. As palavras-chaves utilizadas em cada uma das dimensões foram obtidas através da revisão informal da literatura. Assim, a definição de cada um dos conceitos da *string* de busca, seguido das palavras-chave utilizadas são:

- População: Projetos em geral de desenvolvimento de software

- Palavras-chave:
  - Software, System, Application, Program, Project
- Intervenção: Técnicas de inspeção para fluxos de atividades
  - Palavras-chave:
    - Reading Technique, Inspection Technique, Review Technique, Verification Technique, Validation Technique, Verifying Technique, Reading Approach, Inspection Approach, Review Approach, Verification Approach, Validation Approach, Verifying Approach, Reading Procedure, Inspection Procedure, Review Procedure, Verification Procedure, Validation Procedure, Verifying Procedure, Reading Activity, Inspection Activity, Review Activity, Verification Activity, Validation Activity, Verifying Activity, Reading Method, Inspection Method, Review Method, Verification Method, Validation Method, Verifying Method, Reading Strategy, Inspection Strategy, Review Strategy, Verification Strategy, Validation Strategy, Verifying Strategy, OORTS, Object Oriented Reading Technique, UBR, Usage-Based Reading, CBR, Checklist-Based Reading, DBR, Defect-Based Reading, PBR, Perspective-Based Reading, MBR, Metric-Based Reading
    - Activity Diagram, Activity Model, BPMN, Business Process Modeling Notation, EPC, Event-driven Process Chain, eEPC, Flowchart, Workflow, YAWL, Yet Another Workflow Language, Sequence Diagram, Collaboration Diagram, SPEM, System Process Engineering Meta-Model, Activity graph
- Comparação: Não há
- Resultados: Lista de técnicas de inspeção para fluxo de atividades identificadas.
  - Palavras-chave:
    - Reading Technique, Inspection Technique, Review Technique, Verification Technique, Validation Technique, Verifying Technique, Reading Approach, Inspection Approach, Review Approach, Verification Approach, Validation Approach, Verifying Approach, Reading Procedure, Inspection Procedure, Review Procedure, Verification Procedure, Validation Procedure, Verifying Procedure, Reading Activity, Inspection Activity, Review Activity, Verification

Activity, Validation Activity, Verifying Activity, Reading Method, Inspection Method, Review Method, Verification Method, Validation Method, Verifying Method, Reading Strategy, Inspection Strategy, Review Strategy, Verification Strategy, Validation Strategy, Verifying Strategy, OORTS, Object Oriented Reading Technique, UBR, Usage-Based Reading, CBR, Checklist-Based Reading, DBR, Defect-Based Reading, PBR, Perspective-Based Reading, MBR, Metric-Based Reading

- Activity Diagram, Activity Model, BPMN, Business Process Modeling Notation, EPC, Event-driven Process Chain, eEPC, Flowchart, Workflow, YAWL, Yet Another Workflow Language, Sequence Diagram, Collaboration Diagram, SPEM, System Process Engineering Meta-Model, Activity graph

Foram utilizadas três máquinas de busca: Scopus, IeeeXplore e a EI Compendex. A escolha por estas máquinas de busca está relacionada ao fato de referenciar grande parte do conhecimento científico da área de engenharia de software. Além das máquinas de busca citadas, uma busca manual nos anais do SBES, SBQS e ESELAW foi realizada, pois as máquinas de busca ainda não indexam artigos destes eventos.

A *string* de busca combinou termos relacionados à inspeção e seus sinônimos juntamente com termos relacionados às técnicas e seus respectivos sinônimos, evitando assim termos isolados como “inspeção” e seus sinônimos. Os termos conhecidos sobre inspeção baseada em perspectiva e seus respectivos acrônimos também foram incluídos na *string* de busca. Vale ressaltar que se existir alguma técnica de inspeção com nome diferente das citadas nas palavras-chaves na perspectiva de inspeção, a mesma não será retornada pela *string* de busca. Uma observação importante está nos termos das palavras-chaves da dimensão de Resultados (*Outcomes*) serem idênticos aos termos utilizados para Intervenção. Esta igualdade se justifica, pois foi definido que os Resultados representam a lista de técnicas de inspeção em diagramas de fluxos de atividades, que são identificadas na dimensão de Intervenção.

A *string* de busca foi composta através dos conectores AND e OR juntamente com as palavras-chave definidas para População, Intervenção e Resultados. Como os termos de Resultados são iguais aos de Intervenção, a *string* de busca foi simplificada contendo

apenas uma vez os termos das palavras chaves das dimensões de Intervenção e Resultados. Assim, a *string* de busca completa utilizada foi:

((Software OR System OR Application OR Program OR Project) AND (Reading Technique OR Inspection Technique OR Review Technique OR Verification Technique OR Validation Technique OR Verifying Technique OR Reading Approach OR Inspection Approach OR Review Approach OR Verification Approach OR Validation Approach OR Verifying Approach OR Reading Procedure OR Inspection Procedure OR Review Procedure OR Verification Procedure OR Validation Procedure OR Verifying Procedure OR Reading Activity OR Inspection Activity OR Review Activity OR Verification Activity OR Validation Activity OR Verifying Activity OR Reading Method OR Inspection Method OR Review Method OR Verification Method OR Validation Method OR Verifying Method OR Reading Strategy OR Inspection Strategy OR Review Strategy OR Verification Strategy OR Validation Strategy OR Verifying Strategy OR OORTS OR Object Oriented Reading Technique OR UBR OR Usage-Based Reading OR CBR OR Checklist-Based Reading OR DBR OR Defect-Based Reading OR PBR OR Perspective-Based Reading OR MBR OR Metric-Based Reading) AND (Activity Diagram OR Activity Model OR BPMN OR Business Process Modeling Notation OR EPC OR Event-driven Process Chain OR eEPC OR Flowchart OR Workflow OR YAWL OR Yet Another Workflow Language OR Sequence Diagram OR Collaboration Diagram OR SPEM OR System Process Engineering Meta-Model OR Activity graph))

### **3.3 Execução da *quasi*-Revisão Sistemática**

As buscas iniciais foram realizadas no dia 20 de abril de 2011 nas três máquinas de busca e foram considerados os artigos que estivessem escritos em português ou em inglês. Diferentes máquinas de busca frequentemente indexam o mesmo artigo, portanto já era esperado que as máquinas retornassem artigos em comum. Assim, dentro de 286 artigos retornados, foram encontradas 98 duplicatas e, após sua eliminação, obteve-se o total de 188 artigos distintos. A Tabela 3-1 indica quantidade de artigos retornados pelas máquinas de busca utilizadas. A Figura 3-1 indica a quantidade de artigos distintos (sem duplicatas) retornados por cada uma das máquinas de busca. Na eliminação de duplicatas, foi adotada uma ordem de prioridade entre as máquinas, onde a Scopus tinha a maior prioridade, seguida da IeeeXplore e, por fim, a EI Compendex. A Scopus foi escolhida como de maior prioridade, pois esta máquina é mais ampla, ou seja, referencia a maior quantidade de eventos. A IeeeXplore e a EI Compendex não possuem

características muito distintas entre si, portanto, a escolha da IeeeXplore como de segunda maior prioridade foi por conveniência para esta dissertação.

Tabela 3-1 Máquinas de busca e quantidade de artigos retornados.

Máquina de Busca	Número de artigos encontrados
Scopus	158
IeeeXplore	44
El Compendex	84
Total	286
Duplicatas	98
Total (sem duplicatas)	188

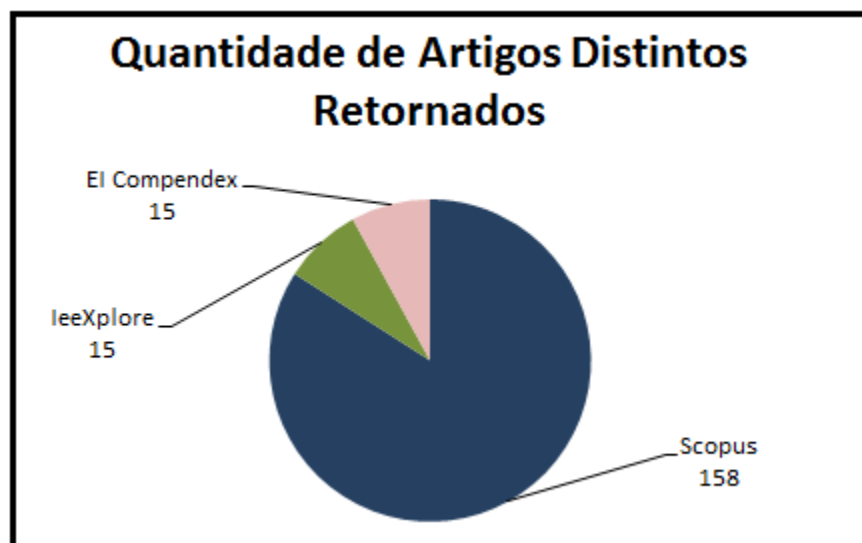


Figura 3-1 Artigos distintos retornados pelas máquinas de busca.

### 3.3.1 Seleção dos artigos

A leitura de todos os títulos e resumos dos artigos foi realizada pelo pesquisador e os mesmos foram classificados em incluídos ou excluídos. Para realizar essa classificação, os seguintes critérios de inclusão e exclusão foram aplicados sobre os títulos e resumos:



- **Critérios de Inclusão:**
  - Os artigos devem tratar sobre inspeção; e
  - Os artigos devem permitir a identificação de pelo menos uma técnica ou abordagem de inspeção em algum diagrama de fluxo de atividade; e
  - Os artigos devem apresentar uma referência bibliográfica que caracterize a abordagem de inspeção utilizada, caso não seja de sua autoria.
  
- **Critérios de Exclusão**
  - Os artigos que não tratem sobre inspeção; ou
  - Artigos que apresentam uma abordagem de inspeção que dependa de transformações dos modelos de *workflow* em outras representações, como redes de *Petri*, algoritmos, grafos ou *model checking*; ou
  - Artigos que somente descrevam regras e/ou critérios de qualidade, sem especificar ou referenciar uma abordagem de inspeção; ou
  - Artigos cuja abordagem de inspeção não apresente oportunidades para identificação de defeitos, como abordagens baseadas em regras de redução e refatoração.

Um segundo pesquisador também participou da execução e realizou a leitura (título, abstract) e classificação de todos os artigos distintos encontrados, utilizando os mesmos critérios de inclusão e exclusão definidos e adotados pelo primeiro pesquisador. A participação do segundo pesquisador representa uma forma de reduzir um possível viés ou equívoco cometido pelo primeiro pesquisador. Assim as duas classificações foram comparadas e no caso de divergência entre os pesquisadores, o artigo foi incluído, para posterior leitura detalhada do artigo.

### **3.3.2 Extração das Informações**

Após a conferência entre as classificações de todos os artigos realizadas pelos dois pesquisadores, seis artigos foram classificados como incluídos. Os artigos incluídos foram inteiramente lidos a fim de identificar não apenas as técnicas, como também outras

informações relevantes para o contexto da pesquisa. Nesta fase de extração, foram extraídas as informações conforme mostra a Tabela 3-2:

Tabela 3-2 Informações extraídas dos resultados.

<b>Informação</b>	<b>Descrição</b>
Título	Título do artigo
Autores	Lista de autores do artigo
Ano de publicação	Ano em que o artigo foi publicado
Fonte da publicação	Nome do <i>journal</i> /conferência/local onde o artigo foi publicado
Fonte	Máquina de busca que retornou o artigo
Tipo de artigo	Onde a abordagem foi aplicada
Resumo	O resumo completo do artigo
Nome da abordagem	Nome da abordagem proposta no artigo
Modelo inspecionado	Tipo de modelo inspecionado
Inspeção individual	Se a inspeção foi realizada foi efetuada individualmente
Orientação	Inspeção foi vertical ou horizontal
Tipo de técnica	Tipo de abordagem proposta (técnica de leitura, <i>checklist</i> , etc)
Foco da abordagem	Se o foco da abordagem é sintático ou semântico
Referências com a descrição	A abordagem é do próprio autor ou faz referência
Tipo de projeto	O tipo de projeto em que a abordagem foi proposta
Tipo de organização	O tipo de organização que propôs a abordagem

Durante a leitura completa e extração de informação dos seis artigos incluídos, apenas três deles realmente condiziam com a questão de pesquisa desta *quasi-revisão* sistemática. Este problema de falso positivo se dá devido aos resumos escritos de maneira inadequada, que é uma ameaça à validade já prevista, pois os resumos dos três artigos excluídos tratavam de inspeções em diagramas de fluxo de atividades, mas somente após a leitura completa dos artigos foi constatado que os mesmos continham abordagens que utilizavam algoritmos ou uma variação de redes de *Petri*, os quais estão descritas nos critérios de exclusão. A Tabela 3-3 mostra os artigos que foram incluídos

para a extração de dados, por satisfazerem todos os critérios de inclusão e de exclusão, e indica quais estão em conformidade após a leitura completa do artigo.

Tabela 3-3 Artigos incluídos após os critérios de inclusão e exclusão.

<b>Título</b>	<b>Autores (Ano)</b>	<b>Conformidade</b>
An Inspection Approach for Conceptual Models in Notations Derived from UML: A Case Study	Tanriöver e Bilgen (2007)	Sim
Obstacles to Comprehension in Usage Based Reading	Cooper <i>et al.</i> (2007)	Sim
Activity Diagram Inspection on Requirements Specification	de Mello <i>et al.</i> (2010)	Sim
Verifying Workflows with Cancellation Regions and OR-joins: An Approach Based on Reset Net and Reachability Analysis	Wynn <i>et al.</i> (2006)	Não
Verification of Composite Services with Temporal Consistency Checking and Temporal Satisfaction Estimation	Ismail <i>et al.</i> (2009)	Não
An Approach to Validating Transactional Properties of WS-BPEL Composition	Wen <i>et al.</i> (2009)	Não

A Tabela 3-4 apresenta as informações extraídas dos artigos que realmente estavam em conformidade com a questão de pesquisa desta *quasi*-revisão sistemática.

Tabela 3-4 Informações extraídas dos artigos selecionados

	<b>Tanriöver e Bilgen (2007)</b>	<b>Cooper <i>et al.</i> (2007)</b>	<b>de Mello <i>et al.</i> (2010)</b>
<b>Fonte da publicação</b>	ISCIS	ASWEC	SBES
<b>Fonte</b>	Scopus	IEEE	IEEE
<b>Tipo de Artigo</b>	Aplicação Industrial	Academia	Academia
<b>Nome da Abordagem</b>	Não Especifica	Não Especifica	Não Especifica
<b>Modelo Inspeccionado</b>	Diagrama de Atividades (notação KAMA)	Diagrama de Sequência e Código	Diagrama de Atividades

<b>Inspeção individual</b>	Sim	Sim	Sim
<b>Orientação</b>	Intra e Inter Diagramas	Inter Diagramas	Intra Diagramas
<b>Tipo da técnica</b>	Técnica de Leitura	SBR/ <i>Ad-hoc</i>	<i>Checklist</i>
<b>Foco da Abordagem</b>	Sintático e Semântico	Sintático e Semântico	Sintático e Semântico
<b>Referências com a descrição</b>	Próprio	Vários autores, especificados na Tabela 1 do artigo	Não Especifica
<b>Tipo de aplicação</b>	Não Especifica	Projeto <i>toy</i> de tocador de áudio	Projeto de Extração de petróleo em águas profundas pelas plataformas através do uso de dutos.
<b>Tipo de organização</b>	Não Especifica	Academia	Academia

### 3.3.3 Avaliação da Qualidade dos Artigos

Após a extração de todas as informações dos três artigos foi realizada uma avaliação sobre a qualidade dos mesmos. Apesar de restarem poucos artigos e isso por vezes não justificar um rigor exagerado em relação à qualidade individual dos artigos, para a fase de análise, a avaliação da qualidade dos artigos está presente devido à garantia da completude do protocolo de revisão definido. Para essa avaliação, as questões a seguir foram utilizadas, onde cada artigo foi pontuado respeitando a nota máxima de cada questão, que está entre parênteses. Assim, quanto mais alta a nota conferida, maior a qualidade atribuída ao artigo, de acordo com a perspectiva deste protocolo:

1. Onde se localiza a descrição da abordagem de inspeção? (1 ponto)

Avalia se a descrição da abordagem se encontra no próprio artigo ou se é referência de outro artigo.

2. O artigo utiliza terminologia adequada? (0.5 ponto)

Avalia se os termos utilizados nos artigos estão condizentes com os termos utilizados na área de Engenharia de Software.

3. O artigo apresenta algum estudo sobre a abordagem de inspeção? (1 ponto)

Avalia se o artigo apresenta algum tipo de estudo (prova de conceitos, experimentos, *surveys*, etc).

4. Os resultados do estudo sobre a abordagem de inspeção são descritos? (1 ponto)

Avalia, caso o artigo apresenta algum tipo de estudo, se os resultados do mesmo estão descritos.

5. Os objetivos estão claramente indicados (razão para o estudo)? (1 ponto)

Avalia se os objetivos da abordagem de inspeção e do estudo estão bem definidos.

6. Há uma descrição do contexto em que a pesquisa foi realizada? (1 ponto)

Avalia se está bem descrito o contexto em que o estudo foi aplicado.

7. Há alguma restrição ou condições sobre o contexto em que a técnica deve ser utilizada? (0,5 ponto)

Avalia se as restrições ou condições sobre o contexto da aplicação da técnica estão explícitas no artigo.

A Tabela 3-5 apresenta os valores de cada artigo selecionado com relação aos itens de avaliação descritos. Ao final é apresentada a soma de cada um dos itens avaliados. DE MELLO *et al.* (2010) possui a maior pontuação, seguido de TANRIÖVER e BILGEN (2007) e COOPER *et al.* (2007).

Tabela 3-5 Conceitos com relação à qualidade dos artigos selecionados.

	Tanriöver e Bilgen (2007)	Cooper <i>et al.</i> (2007)	de Mello <i>et al.</i> (2010)
<b>Onde se localiza a descrição da abordagem de inspeção? (1 ponto)</b>	1	0	1
<b>O artigo utiliza terminologia adequada? (0.5)</b>	0	0.5	0.5

<b>ponto)</b>			
<b>O artigo apresenta algum estudo sobre a abordagem de inspeção? (1 ponto)</b>	0.5	0.5	0.5
<b>Os resultados do estudo sobre a abordagem de inspeção são descritos? (1 ponto)</b>	0	1	1
<b>Os objetivos estão claramente indicados (razão para o estudo)? (1 ponto)</b>	1	1	1
<b>Há uma descrição do contexto em que a pesquisa foi realizada? (1 ponto)</b>	1	1	1
<b>Há alguma restrição ou condições sobre o contexto em que a técnica deve ser utilizada? (0,5 ponto)</b>	0.5	0	0.5
<b>Total de Pontos</b>	<b>4</b>	<b>4</b>	<b>5.5</b>

Os três artigos selecionados apresentavam provas de conceito como um estudo para avaliação da abordagem de inspeção, portanto receberam o valor de 0.5, visto que há tipos de estudos mais rigorosos como, por exemplo, o experimento controlado, que possui uma confiabilidade de resultados mais relevantes que uma prova de conceito. O valor 0 (zero) para alguma questão indica que o artigo não oferece as informações necessárias para responder a questão.

### **3.4 Resultados encontrados**

Os três estudos relevantes (TANRIÖVER e BILGEN, 2007), (COOPER *et al.*, 2007) e (DE MELLO *et al.*, 2010) encontrados foram analisados. Um resumo sobre os artigos é apresentado a seguir, onde os dois primeiros também representam os artigos de controle.

### **3.4.1 Tanriöver e Bilgen (2007)**

A técnica desenvolvida por TANRIÖVER e BILGEN (2007) inspeciona uma variação do diagrama de atividades da UML, pertencente à notação KAMA. A técnica desenvolvida consiste em uma série de recomendações que visam inspecionar o diagrama com a especificação do caso de uso. Estas recomendações se assemelham a técnicas de leitura, porém possuem instruções mais simples do que uma técnica de leitura.

Os autores ainda propõem uma inspeção inter diagramas, composta por: inspeção de conformidade horizontal e inspeção de conformidade vertical. A primeira verifica o diagrama de atividades com os diagramas de ontologia, enquanto a segunda verifica os diagramas de atividades com os seus subdiagramas. Os aspectos semânticos da UML 2.0, tais como bifurcação/sincronização e pré e pós-condições não foram contempladas pela abordagem proposta pelos autores.

### **3.4.2 Cooper et al. (2007)**

A proposta de COOPER *et al.* (2007) inspeciona Diagramas de Sequência e código fonte (Java) utilizando duas técnicas: UBR (*Usage-Based Reading*) e de Análise de Protocolo. UBR é uma das técnicas de leitura, que consiste em instruções que orientam o inspetor na análise dos documentos a serem inspecionados.

Análise de protocolo é utilizada para explorar o aspecto cognitivo dos inspetores. Assim, eles foram solicitados a expressar em voz alta os seus pensamentos enquanto realizaram a inspeção durante o estudo realizado. O estudo de (HUNGERFORD *et al.*, 2004) aponta que alternar a atenção em mais de um artefato durante a inspeção, tende a um resultado mais eficiente com relação à detecção de defeitos.

Uma prova de conceito foi realizada por 10 inspetores (5 profissionais da indústria, 3 graduados e 2 alunos de graduação) para avaliar a habilidade dos inspetores em seguir o cenário UBR. Um cenário de caso de uso de um sistema *toy* de tocador de áudio, de 193 linhas de código foi inspecionado, onde os participantes do estudo tinham que identificar a ordem e o número da linha do comando executado. O tempo em que cada comando foi identificado também é registrado para análise posterior.

O resultado da prova de conceito indicou que há seis questões que dificultam na identificação de quais são as linhas do código executadas. Esta conclusão foi obtida pelos dados coletados das inspeções. Os pontos são listados a seguir:

1. Dificuldade na identificação do início do cenário: nenhuma das inspeções começou no mesmo ponto da solução apontada do modelo;
2. Omissão accidental: os participantes listavam as chamadas dos métodos, mas as mesmas não correspondiam ao conteúdo do método;
3. Polimorfismo: sete participantes não reconheceram o polimorfismo de um método da aplicação, onde a ideia desta característica de orientação a objeto poderia ser percebida pelo diagrama de sequência utilizado;
4. Dificuldade na identificação do método: alguns casos o inspetor não identificou a sobrecarga de métodos ou confundiu as assinaturas dos métodos (listou getNextTrack() ao invés de hasNextTrack());
5. Troca de contexto: um aumento no tempo nas decisões foi percebido quando um método era invocado ou retornado. Assim, os inspetores dispenderam um tempo adicional para localizar em que classe o método em questão foi invocado;
6. Avaliação de condições: houve alguns casos onde os inspetores decidiram de forma incorreta as expressões condicionais.

### **3.4.3 de Mello et al. (2010)**

A técnica de DE MELLO *et al.* (2010) consiste em um *checklist* (*ActCheck*) com 34 questões para inspecionar requisitos especificados em diagramas de atividades. Para cada questão do *checklist* são apresentadas as opções para o inspetor assinalar entre “Sim”, “Não” ou “NA” (não aplicável). A técnica também faz uso de um relatório de discrepância onde o inspetor preenche ao encontrá-las, o tipo de defeito, uma breve descrição e a questão do *checklist* que foi utilizada para identificar a discrepância.

O *checklist* proposto é configurável e utiliza um questionário de caracterização da aplicação, que é preenchido pelo engenheiro de software responsável pelo projeto. Assim, questões do *checklist* que não estão no contexto do projeto são descartadas, restringindo o foco do inspetor para apenas questões relevantes para a inspeção.

Um prova de conceito foi realizada para verificar a aplicabilidade da técnica proposta. Um projeto no contexto de extração de petróleo em águas profundas por plataformas, através do uso de dutos, foi utilizado nesta prova de conceito. A especificação do projeto já havia sido inspecionada de maneira *ad-hoc* por 5 pesquisadores e foram encontrados 81 defeitos, onde apenas 6 estavam relacionados com *workflow* científico (diagrama de atividades) em 107 minutos. Dentre os 6 defeitos, 4



correspondem a defeitos sintáticos, enquanto os outros 2 correspondem a defeitos semânticos. O *checklist* utilizado continha 19 questões e foi configurado previamente de acordo com as características do projeto. Assim, um inspetor realizou a inspeção utilizando este *checklist* e encontrou, ao final, 25 discrepâncias em 117 minutos, porém dentre as discrepâncias detectadas, 5 eram falsos positivos.

Uma comparação entre a inspeção *ad-hoc* e a inspeção utilizando o *checklist* é possível observar que uma inspeção necessitou de menos tempo que a outra, porém com uma quantidade de tempo quase que equivalente o uso do *checklist* encontrou mais defeitos no total, analisando apenas uma parte da documentação (diagrama de atividades). O resultado obtido não pode ser generalizado, pois representa apenas indícios da sua aplicabilidade e que necessitam de mais estudos para uma melhor avaliação.

### 3.5 Atualização da *quasi*-Revisão Sistemática

O protocolo de revisão foi reexecutado em 16 de abril de 2013, com o intuito de verificar se algum trabalho relevante na literatura técnica foi publicado no período entre 2011 e 2013. Assim, o mesmo plano de estudo foi utilizado, porém o Portal Capes não oferece mais acesso à máquina de busca El Compendex. Portanto, a *string* de busca foi executada apenas na Scopus e na IeeeXplore. A Tabela 3-6 apresenta a quantidade de resultados retornados pelas máquinas de busca nesta atualização da *quasi*-revisão sistemática.

Tabela 3-6 Máquinas de buscas e quantidade de artigos retornados na reexecução da *quasi*-revisão sistemática.

Máquina de Busca	Número de artigos encontrados
Scopus	49
IeeeXplore	14
Total	63
Duplicatas	9
Total (sem duplicatas)	54

Os 54 artigos foram analisados e os critérios de inclusão e exclusão definidos na seção 3.3.1 foram aplicados sobre o título e o resumo dos artigos. Um segundo

pesquisador também participou desta etapa e, ao final, foi incluído apenas um artigo: MASSOLLAR *et al.* (2011). Este artigo é um desdobramento do trabalho de DE MELLO (2010), retornado na primeira execução desta *quasi*-revisão sistemática. As informações extraídas do único artigo selecionado são apresentadas na Tabela 3-7.

Tabela 3-7 Informações extraídas de MASSOLLAR *et al.* (2011).

	<b>Massollar <i>et al.</i> (2011)</b>
<b>Fonte da publicação</b>	SCCC
<b>Fonte</b>	IEEE
<b>Tipo de Artigo</b>	Academia
<b>Nome da Abordagem</b>	<i>ActCheck</i>
<b>Modelo Inspeccionado</b>	Diagrama de Atividades
<b>Inspeção individual</b>	Sim
<b>Orientação</b>	Intra Diagramas
<b>Tipo da técnica</b>	<i>Checklist</i>
<b>Foco da Abordagem</b>	Sintático e Semântico
<b>Referências com a descrição</b>	Próprio autor
<b>Tipo de aplicação</b>	Módulo financeiro de um sistema de informação em larga escala
<b>Tipo de organização</b>	Academia

O trabalho de MASSOLLAR *et al.* (2011) consiste em dois estudos experimentais conduzidos para avaliar a ferramenta UCA (*UseCaseAgent*), proposta pelo próprio pesquisador, sobre aspectos diferentes no contexto de um módulo de um projeto real de um sistema de informação larga escala. A ferramenta é utilizada para descrever a especificação de casos de uso em diagramas de atividades. Assim ela permite a criação, a edição e a verificação da especificação dos casos de uso através das restrições contidas no metamodelo *UCModel*.

O primeiro estudo teve como objetivo comparar o tempo para criar diagramas de atividades utilizando a ferramenta UCA e outra para edição de diagramas, ADE (*activity diagram editor*). O estudo ocorreu em duas rodadas, onde os participantes recebiam um conjunto de casos de uso especificados em formato textual definido por um *template*. Na primeira rodada, os participantes criavam um diagrama de atividades para cada caso de uso recebido, utilizando a ferramenta ADE e, na segunda rodada, os diagramas eram

criados utilizando a ferramenta UCA. O resultado indicou que a ferramenta UCA auxilia na criação dos diagramas, pois de maneira geral, menos tempo foi despendido nessa atividade.

O segundo estudo teve como objetivo comparar a quantidade de defeitos encontrados através de inspeções realizadas nos diagramas gerados, pelas duas ferramentas do primeiro estudo. Os participantes não sabiam que os diagramas inspecionados foram gerados por diferentes ferramentas. O *checklist* configurável (*ActCheck*), apresentado na seção 3.4.3, foi evoluído e aplicado nesse estudo. A versão de *ActCheck* utilizada no estudo foi dividida em dois *checklists* (A e B), onde o primeiro contém questões que avaliam cada elemento do diagrama de atividades e da descrição dos requisitos. O checklist B foi aplicado após o primeiro e contém questões que identificam defeitos nos requisitos que podem ser melhor visualizados nos diagramas. O resultado do estudo apresentou que os diagramas criados pela ferramenta UCA contém menos defeitos que os diagramas gerados pela ADE, pois a diferença da quantidade de defeitos reportados nos diagramas gerados pela UCA é significativamente menor do que a reportada pela outra ferramenta. Os resultados dos dois estudos não podem ser generalizados por terem sido obtidos em um contexto específico.

### **3.6 Ameaças à validade**

Durante a execução desta *quasi-revisão* sistemática, algumas ameaças à validade foram identificadas. Uma delas está relacionada ao vocabulário não definido na área, pois apesar da revisão inicial da literatura para a busca de termos utilizados pela comunidade, referentes à inspeção e diagramas de fluxos de atividades, existe ainda a possibilidade de que algum termo não conhecido até então tenha sido utilizado em algum artigo e, portanto, não foi retornado por não ter sido contemplado pela *string* de busca.

Outra ameaça à validade está no possível viés do aspecto humano. Um segundo pesquisador participou da seleção dos artigos para validar os resultados do primeiro pesquisador. Essa foi uma tentativa de evitar ao máximo os equívocos cometidos no decorrer da seleção pelo primeiro pesquisador, porém ainda assim erros humanos são passíveis de ocorrer.

### 3.7 Conclusão

O resultado final desta *quasi*-revisão sistemática identificou apenas quatro artigos relevantes para a questão de pesquisa. O trabalho de TANRIÖVER e BILGEN (2007) apresenta uma técnica de leitura, com foco tanto no aspecto sintático, quanto no semântico, apesar do último aspecto não ter sido muito explorado. Características presentes nos diagramas de atividades da UML 2 também não foram contempladas, tais como bifurcação/sincronização, pré e pós-condições.

O trabalho de COOPER *et al.* (2007) apresentou um estudo realizado que identificou alguns pontos em que os inspetores tiveram maior dificuldade em inspecionar o diagrama de sequência, com o seu código correspondente. Apesar dos resultados obtidos, o estudo realizado foi uma prova de conceito e, portanto, mais estudos com maior rigor formal deveriam ser executados para agregar mais força nos resultados encontrados.

O trabalho de DE MELLO *et al.* (2010) é o de maior relevância para este protocolo, que consiste em um *checklist* para inspecionar diagramas de atividades que descrevem casos de uso, porém apenas uma prova de conceito foi realizada, não apresentando, portanto, evidência sobre sua viabilidade. Este *checklist* (mencionado na seção 2.4.3) foi evoluído e é apresentado em DE MELLO (2011).

O trabalho de MASSOLLAR *et al.* (2011), única pesquisa relevante encontrada na atualização desta *quasi*-revisão sistemática, trata de uma evolução DE MELLO *et al.* (2010), onde uma nova versão do *checklist* é brevemente apresentada, entretanto o foco do artigo foi avaliar a ferramenta *UseCaseAgent* através do *checklist*.

Apesar da existência de técnicas de inspeção em diagramas de fluxos de atividades, os resultados obtidos nas duas rodadas desta *quasi*-revisão sistemática indicam a carência de técnicas de inspeção que garantam a verificação de consistência entre diagramas de fluxos de atividades (atividades e estados) no contexto de um mesmo projeto. A falta de disponibilidade de tecnologias aliada a necessidade de prover este tipo de técnica para os projetos atuais reforçam a oportunidade de criação de uma técnica de inspeção para preencher esta lacuna. Desta forma, uma proposta de técnica para inspecionar diagramas de estados com diagramas de atividades é apresentada no próximo capítulo.

## 4 Técnica de Leitura para inspecionar Diagramas de Estados com referência nos Diagramas de Atividades

*Este capítulo apresenta a primeira versão da técnica de leitura proposta que possui a finalidade de inspecionar diagramas de estados utilizando como referência os diagramas de atividades representando a especificação de casos de uso. As premissas da técnica de leitura e a relação entre os diagramas também são discutidas neste capítulo.*

### 4.1 Introdução

Os resultados da *quasi*-revisão sistemática (Capítulo 3) indicam a existência de algumas técnicas de inspeção aplicadas a diagramas de fluxos de atividades, porém não foi encontrada na literatura técnica alguma técnica de inspeção que explore veementemente o aspecto semântico. As técnicas que foram encontradas pela revisão possuem um foco maior no aspecto sintático.

O trabalho de TANRIÖVER e BILGEN (2007) consiste em uma série de recomendações para inspeção de uma variação do diagrama de atividades, porém alguns recursos semânticos da UML 2 (OMG, 2011) não foram contemplados e, portanto, a verificação semântica é pouco explorada. COOPER *et al.* (2007) propôs a inspeção em diagrama de sequência e de código Java, através de UBR e de Análise de Protocolo, e indicou questões que mostram uma maior dificuldade em indicar qual a linha do código que era executada, dado um diagrama de sequência.

*Actcheck* (DE MELLO, 2011) propõe uma técnica de inspeção que em sua primeira versão é composta por dois *checklists*. Essa técnica de inspeção possui o foco semântico, além do foco sintático, porém com base nos resultados do experimento realizado, para a detecção de defeitos semânticos, o *checklist* pouco auxiliou, pois as instruções são genéricas e dependem muito da experiência do inspetor para que esse tipo de defeito seja identificado.

Assim, apesar das técnicas de inspeção encontradas, identificou-se uma carência de tecnologia nesta área, principalmente no que diz respeito à inspeção envolvendo a comparação com diagrama de estados.

A ideia de propor uma técnica de leitura está relacionada à maior taxa de detecção de defeitos (PORTER *et al.*, 1995; SHULL, 1998) quando comparada com os demais tipos de técnicas de inspeção como, por exemplo, *ad-hoc* e *checklists*. As técnicas de leitura possuem a característica de ter um maior foco semântico na detecção de defeitos, além de também ajudar a capturar defeitos sintáticos. Existem técnicas de leitura aplicáveis a diagramas de projeto, como OORTs (TRAVASSOS *et al.*, 1999). Entretanto, OORTs (apresentado na seção 2.4.3) não contempla os diagramas de atividades nas inspeções. O motivo é que, na época da sua criação, diagramas de atividades ainda não eram frequentemente utilizados em projetos de software. Desta forma, surgiu a ideia de incorporar a OORTs uma técnica de leitura que utilizasse esse diagrama, aumentando assim a cobertura desta família de técnicas de leitura.

As técnicas de OORTs realizam a inspeção comparando dois diagramas. Os diagramas de estados podem ser considerados uma representação dual dos diagramas de atividades, pois os eventos que provocam a transição de estados representam ações ou atividades nestes diagramas. Portanto, para apoiar a leitura dos diagramas de estado foi então escolhido o diagrama de atividades. A Figura 4-1 ilustra o conjunto de técnicas de leitura que compõem OORTs com a inclusão da nova técnica de leitura para diagramas de estados utilizando o diagrama de atividades como referência.

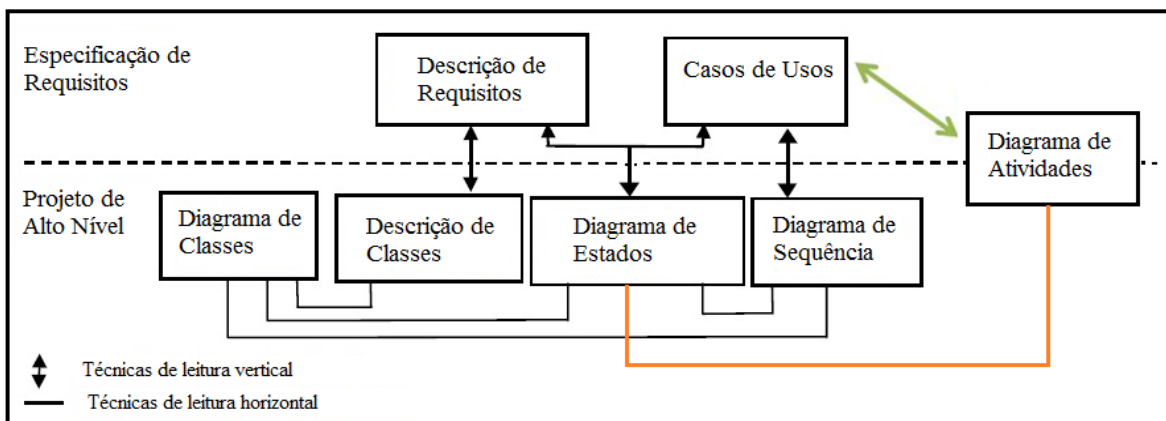


Figura 4-1 Representação gráfica da Técnica de Leitura de Diagramas de Estados com Diagramas de Atividades dentro do contexto de OORTs.

Assume-se que os diagramas de atividades utilizados na inspeção modelam as descrições de casos de uso, seguindo a abordagem proposta por MASSOLLAR (2011), representando assim o comportamento do sistema. Devido a essa característica, os diagramas de atividades na Figura 4-1 estão dispostos entre a especificação de requisitos e o nível de projeto, pois os diagramas são também inerentes à fase de projeto, mas como estes diagramas representam as descrições de casos de uso, esses estão posicionados também na fase de especificação de requisitos (momento de transformação).

Na seção 4.2 são apresentados os detalhes dos diagramas de atividades utilizados como referência na técnica de leitura *Shiô*. Na seção 4.3 são apresentadas as premissas definidas para a aplicação da técnica de leitura. A relação entre os diagramas de estados e de atividades é apresentada na seção 4.4. A primeira versão da técnica de leitura é apresentada na seção 4.5 e, por fim, a conclusão deste capítulo está apresentada na seção 4.6.

## **4.2 Modelos Inspeccionados**

A técnica de leitura proposta nesta dissertação utiliza diagramas de atividades que são comparados com diagramas de estados. Os diagramas de atividades representam descrições de casos de uso e modelados de acordo com a abordagem proposta por MASSOLLAR (2011) (a ser descrita nesta seção).

Um caso de uso representa uma sequência de transações realizada por um sistema para produzir um resultado de valor observável para um determinado ator ou conjunto de atores (JACOBSON, 1992). Os casos de uso resumem as funções do sistema (requisitos funcionais) em termos verificáveis de forma que usuários, executivos e desenvolvedores possam avaliá-los e validá-los de acordo com os seus interesses (COCKBURN, 1997). Essas são algumas das diversas definições encontradas na literatura sobre casos de uso.

Os recursos sintáticos do diagrama de atividades, contemplados pela abordagem, são apresentados na Tabela 4-1.

Tabela 4-1 Recursos sintáticos do diagrama de atividades explorados na abordagem de (MASSOLLAR, 2011).

Nome	Descrição	Símbolo
Nó inicial	Representa o ponto de partida para execução das atividades representadas no diagrama	
Atividade	Um processo ou comportamento que está sendo modelado	
Nó de ação ou Ação	Ação que compõe a sequência de ações do processo ou comportamento que está sendo modelado	
Nó de decisão	Define um ponto a partir do qual existem diferentes alternativas de execução	
Fluxo de controle	Conexão direcionada entre duas ações que denota a sequência de execução do diagrama	
Condição de guarda	Expressão associada a um fluxo de controle que define se este pode ou não executado	[expressão]
Nó final	Representa o ponto de parada da execução das atividades representadas no diagrama	

Desta forma, MASSOLLAR (2011) formalizou a relação entre as estruturas presentes nas descrições de casos de uso com os elementos presentes nos diagramas de atividades, visto que ambos modelam comportamentos de um sistema. Essa similaridade entre os conceitos estão expostos na Tabela 4-2.

Tabela 4-2 Relação dos conceitos entre a descrição de casos de uso e diagramas de atividades (MASSOLLAR, 2011).

Descrição de Casos de Uso	Diagrama de Atividades
Caso de Uso	Atividade
Pré e pós-condições do Caso de Uso	Pré e pós-condições da Atividade
Sequência de passos	Fluxo de controle
Passos do Caso de Uso	Ações
Fluxos Alternativos	Nós de decisão e condições de guarda

O *UCModel* (MASSOLLAR, 2011) é um metamodelo que estende o diagrama de atividades da UML (OMG, 2011) e acrescenta novos elementos para a descrição do



comportamento do caso de uso. O *UCModel* define, para cada ação do diagrama de atividade, três possíveis estereótipos que estão em conformidade com a classificação de Jacobson (1992) para transações. Os estereótipos foram definidos como:

- **Ação do Ator:** representa uma interação do ator com o sistema na qual este faz uma requisição ao sistema informando os dados que foram solicitados pelo sistema. Nos diagramas de atividades as ações desse tipo estão representadas pelo estereótipo <<*actor\_action*>>.
- **Ação do Sistema:** representa uma ação do sistema cujos resultados gerados não são diretamente observados pelo ator. Esta ação está, normalmente, associada à recuperação de informações, alteração do estado interno do sistema e geração de resultados que serão posteriormente apresentados. Nos diagramas de atividades as ações desse tipo estão representadas pelo estereótipo <<*system\_action*>>.
- **Resposta do sistema:** representa uma ação do sistema que tem um resultado observável pelo ator. Essa resposta pode usar uma interface HC, um e-mail, um sinal que é enviado a um sensor ou qualquer outro meio através do qual o sistema possa se comunicar com o mundo externo. Nos diagramas de atividades as ações desse tipo estão representadas pelo estereótipo <<*system\_response*>>.

A *UseCaseAgent* é uma ferramenta proposta por MASSOLLAR (2011) que implementa o metamodelo *UCModel*. A *UseCaseAgent* está integrada à ferramenta BOUML (PAGÉS, 2011) de construção de diagramas UML. Na época do trabalho proposto, a BOUML era uma ferramenta *open-source*, mas atualmente para as novas versões lançadas uma licença deve ser paga para sua utilização. A ferramenta tem o objetivo de apoiar a construção da especificação dos casos de uso, garantindo que essas especificações e restrições estejam em conformidade com o *UCModel*. A apresenta Figura 1-1 um exemplo de um diagrama de atividades construído utilizando a ferramenta *UseCaseAgent*.

### 4.3 Premissas para Aplicação da Técnica

Para a criação e, por consequência, aplicação da técnica foram definidas algumas premissas, que permitiram definir uma perspectiva comum e viável para a criação da técnica de leitura:

a) Os requisitos devem estar previamente inspecionados, para garantir que os diagramas de atividades e os demais diagramas previstos no projeto sejam construídos utilizando informações corretas dos requisitos;

b) Os diagramas de atividades devem proposta por MASSOLLAR (2011). Esta abordagem utiliza a ferramenta *UseCaseAgent*, que faz uso do metamodelo *UCModel* que foram propostos pelo próprio autor da abordagem representar a especificação dos requisitos e devem estar modelados de acordo com a abordagem;

c) Como a técnica de leitura é baseada em OORTs, precisou-se definir qual dos diagramas utilizados (diagrama de atividades ou diagrama de estados) seria considerado o oráculo para a inspeção. Assim, optou-se por escolher o diagrama de atividades, pois para garantir que este diagrama está em conformidade com os requisitos, existe o *checklist* de inspeção *ActCheck* (DE MELLO, 2011), descrito no Capítulo 2;

d) Todos os diagramas da UML utilizados no projeto do sistema devem, obviamente, representar o sistema em questão, pois não faz sentido realizar a comparação de dois diagramas diferentes, durante a inspeção, que não tratem do mesmo sistema.

### 4.4 Relação entre diagramas de estados e diagramas de atividades

A técnica de leitura baseia-se na comparação entre os diagramas de estados com os diagramas de atividades, portanto foi necessário identificar quais elementos dos diagramas de atividades e de estados possuem significado equivalente. Assim, para cada um dos elementos do diagrama de atividades, buscou-se identificar qual era o seu elemento correspondente no diagrama de estados. Um exemplo simples sobre a funcionalidade "Realizar Venda" é utilizado para expor a relação encontrada entre os diagramas. A Figura 4-2 ilustra o diagrama de atividades do exemplo de "Realizar Venda" e o diagrama de estados "Venda" está apresentado na Figura 4-3.

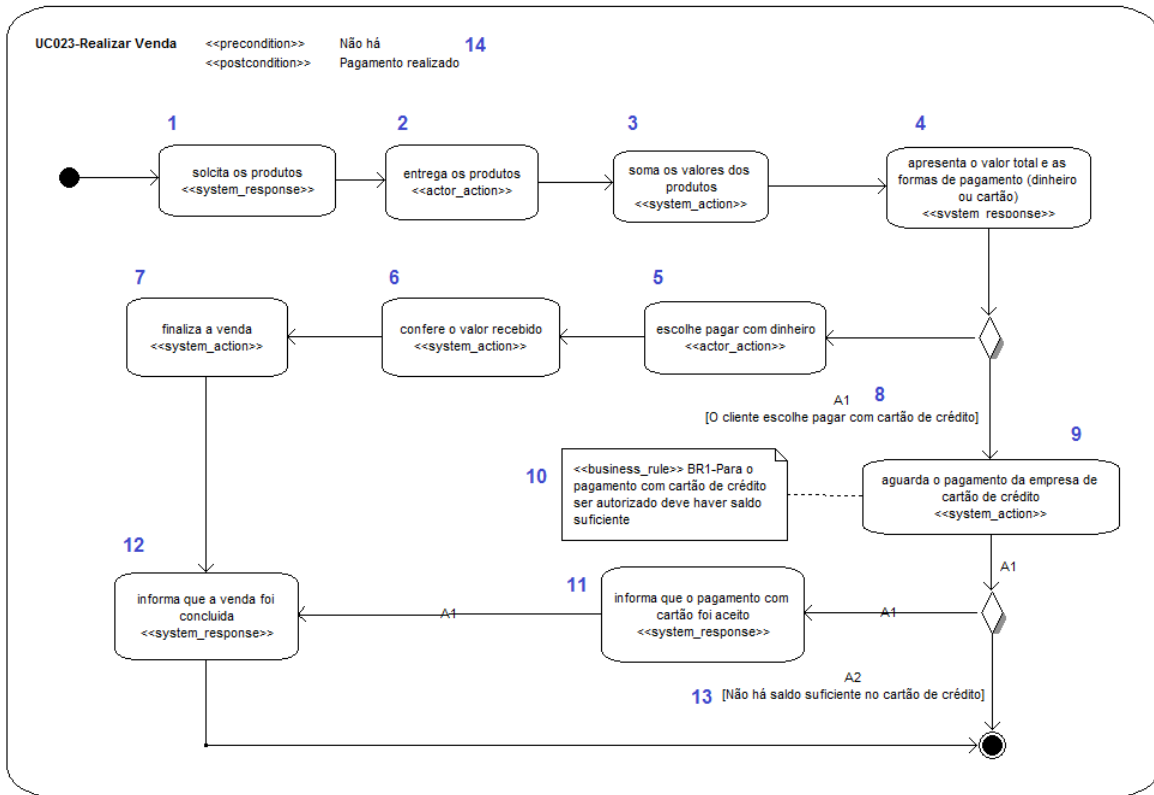


Figura 4-2 Diagrama de atividades de Realizar Venda.

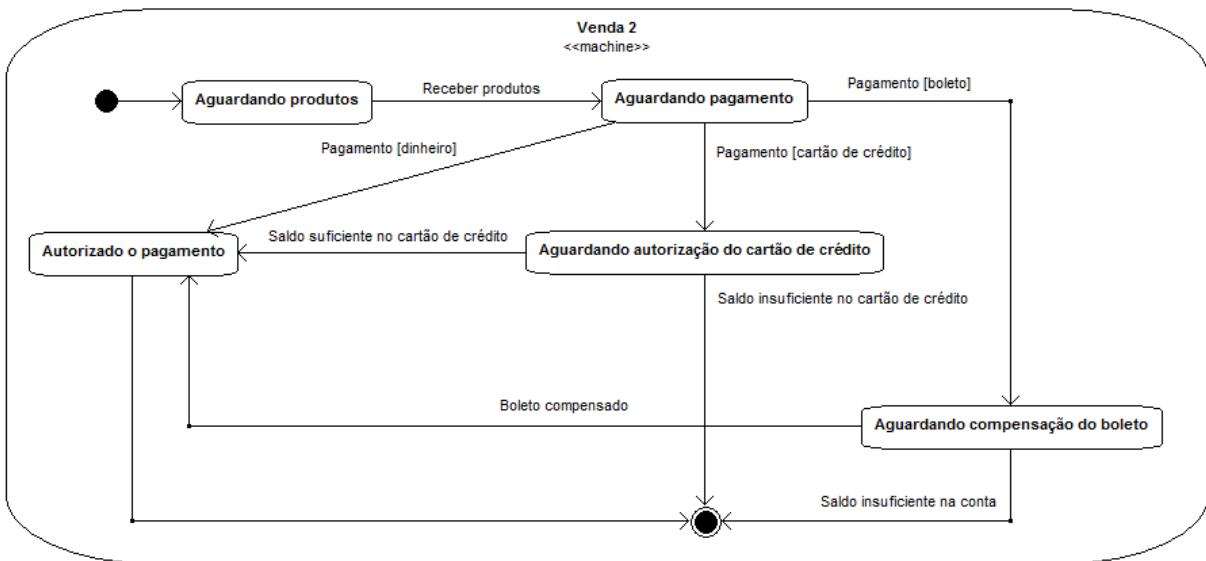


Figura 4-3 Diagrama de estados de Venda.

Segue abaixo a correspondência realizada para os diagramas presentes nas Figura 4-2 e Figura 4-3. O diagrama de atividades teve cada um dos elementos numerados para facilitar mostrar a relação com o diagrama de estados, utilizada como base para a elaboração da proposta da técnica.

### **Ações**

Seguindo a abordagem proposta por MASSOLLAR (2011), as ações do diagrama de atividades podem assumir três diferentes estereótipos, que possuem características que as diferenciam entre si. Os três tipos de ações foram analisadas e encontrou-se a relação onde:

- Ação do Sistema e Resposta do Sistema: correspondem aos estados no diagrama de estados. Como pode ser observado em:
  - Ação (1): corresponde ao estado “Aguardando produtos” no diagrama de estados;
  - Ações (3 e 4): correspondem ao estado “Aguardando pagamento” no diagrama de estados;
  - Ações (6, 7, 11 e 12): correspondem ao estado “Autorizado o pagamento” no diagrama de estados; e
  - Ação (9): corresponde ao estado “Aguardando autorização do cartão de crédito” no diagrama de estados.
  
- Ação do Ator: correspondem às transições no diagramas de estados. Como pode ser observado em:
  - Ação (2): corresponde à transição “Receber produtos” no diagrama de estados; e
  - Ação (5): corresponde à transição “Pagamento no dinheiro” no diagrama de estados.

### **Fluxos de Controle com Pontos de Decisão e Condições de Guarda**

Os fluxos de controle com pontos de decisão e condições de guarda do diagrama de atividades possuem relação com as transições no diagrama de estados, assim como as ações com estereótipos *Ação do Sistema*. Essa relação pode ser encontrada em:

- Condição de Guarda (8): corresponde à transição “Pagamento no cartão de crédito” no diagrama de estados; e

- Condição de Guarda (13): corresponde à transição “Saldo insuficiente no cartão de crédito” no diagrama de estados.

### **Regras de Negócio**

As regras de negócio são representadas no diagrama de atividades através de anotações ligadas às ações que contém a regra, como mostra a Figura 4-2 no elemento identificado como 10. As regras de negócio no diagrama de atividades possuem relação com as transições no diagrama de estados, assim como as ações com estereótipos *Ação do Sistema* e os fluxos de controle com pontos de decisão e condições de guarda. Essa relação pode ser encontrada em:

- Regra de Negócio (10): corresponde à transição “Saldo suficiente no cartão de crédito” no diagrama de estados.

### **Pré e Pós-Condições**

As pré e pós-condições do diagrama de atividades, identificadas com o elemento 14, possuem relação com estados anteriores ou posteriores no diagrama de estados, respectivamente. Como as pré e pós-condições definem a situação em que o sistema se encontra antes e após a execução do diagrama de atividades (funcionalidade), as cláusulas de pré e pós-condições definem, respectivamente, os estados que deveriam e deverão estar antes e após a execução da atividade. Uma classe/objeto (diagrama de estados) pode ser utilizada por diversos diagramas de atividades. Assim as pré e pós-condições podem definir estes estados quando mais de um diagrama de atividades interferir na mudança de estados desta classe/objeto.

O exemplo citado nas Figura 4-2 e **Erro! Fonte de referência não encontrada.** é um exemplo simples, onde a classe/objeto "Venda" tem o ciclo de vida limitado ao diagrama de atividades de "Realizar Venda". Portanto, neste exemplo não é possível estabelecer a relação das pré e pós-condições do diagrama de atividades com o diagrama de estados.

## 4.5 A Técnica Shiô<sup>1</sup>

A técnica de leitura seguiu o mesmo formato de OORTs, contendo uma sequência de passos com entradas, saídas e instruções que guiam o pesquisador a encontrar discrepâncias. A visão geral da técnica é apresentada na Figura 4-4, através de um diagrama de atividades, sendo que a técnica de inspeção é descrita com maiores detalhes ainda nesta seção.

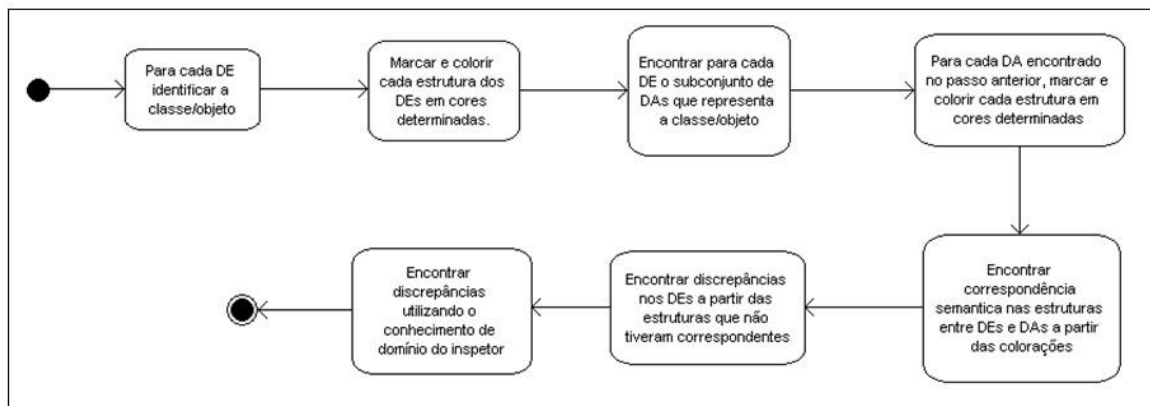


Figura 4-4 Visão geral da técnica proposta.

A técnica proposta é dividida em cinco passos dependentes entre si e, como dito anteriormente, em cada um dos passos foram especificadas as suas entradas e as saídas. A descrição de cada um dos passos da técnica é apresentada na próxima seção.

### 4.5.1 Estrutura da Técnica Shiô

A técnica de leitura foi estruturada em cinco passos que devem ser seguidos sequencialmente pelo inspetor durante a inspeção dos modelos. Cada um dos passos é descrito a seguir, através de um pequeno exemplo, que foi apresentado na seção 4.4, porém o diagrama de estados teve uma pequena modificação para detectar as

---

<sup>1</sup>Shiô significa Sal em japonês. Os japoneses utilizam o sal em diversas situações para proteção, como por exemplo, é costume carregar um pouco de sal em viagens e o mesmo também é jogado no *dohyō* (ringue) de sumô. Após os enterros, os japoneses costumam lavar a mão com água e sal para que os maus espíritos não os acompanhem até a suas casas.

discrepâncias. O diagrama de estado utilizado para ilustrar a técnica é apresentado na Figura 4-5 e o diagrama de atividade é o que foi apresentado anteriormente na Figura 4-2.

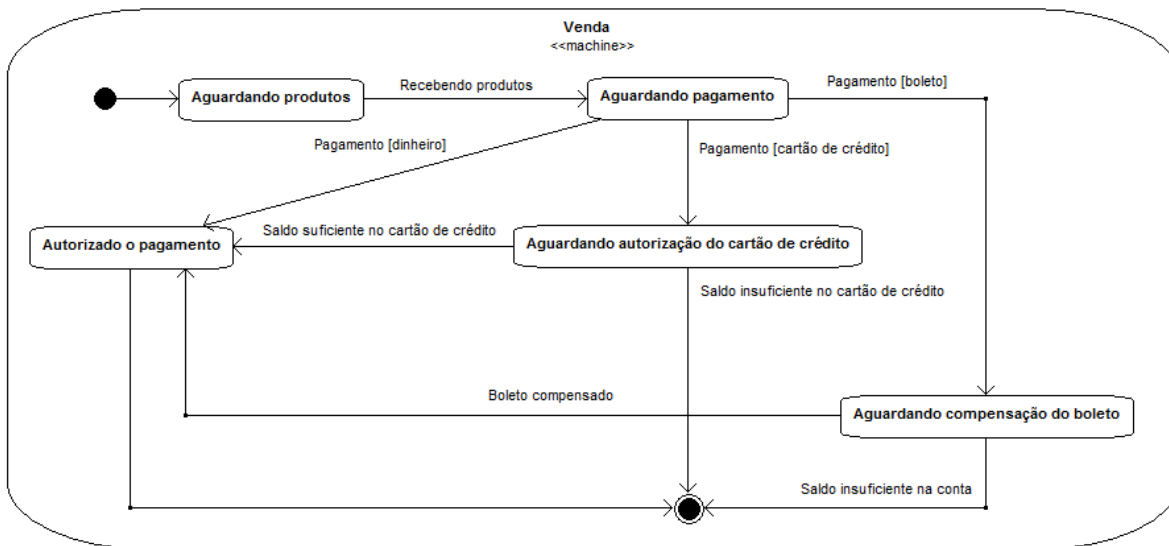


Figura 4-5 Diagrama de estados de Venda utilizado para ilustrar a aplicação da técnica *Shiô*.

### Primeiro Passo

No primeiro passo é solicitado que, para cada diagrama de estados do projeto a ser inspecionado, seja identificada a classe/objeto que o diagrama em questão representa. Instruções para detectar a discrepância, caso não seja possível identificar a classe/objeto são apresentadas.

Após a identificação da classe/objeto, a técnica propõe que o inspetor percorra todo o diagrama e marque os estados e as transições entre estados com cores pré-definidas. Instruções para detectar defeitos também são apresentadas, tanto para o aspecto sintático, quanto para o aspecto semântico, porém o último depende da percepção do inspetor.

Por fim, para cada diagrama de estados analisado, a técnica orienta na identificação de um subconjunto dos diagramas de atividades que manipulem a classe/objeto que é representada pelo diagrama de estados. Para realizar essa identificação são fornecidas instruções para que o inspetor procure por determinados recursos sintáticos (Ex: ações, condições de guarda, ...) nos diagramas de atividades. Uma marcação deve ser feita tanto no diagrama de estados quanto no subconjunto de diagramas de atividades identificado. Essa marcação é o rastro que a classe/objeto está

sendo utilizada nos diagramas de atividades. Portanto, para cada diagrama de estados, deve ser feita uma marcação distinta da realizada nos demais diagramas de estados. Instruções para detecção de defeitos relacionados às nomenclaturas e conceitos ambíguos também são apresentados.

A Figura 4-6 apresenta o diagrama de estados do exemplo, onde os estados foram coloridos em vermelho e as transições em azul. Uma marcação (estrela) foi feita no diagrama de estados e no diagrama de atividades, ilustrado na Figura 4-7. Essa estrela indica que o diagrama Realizar Venda manipula a classe/objeto Venda representada no diagrama de estados.

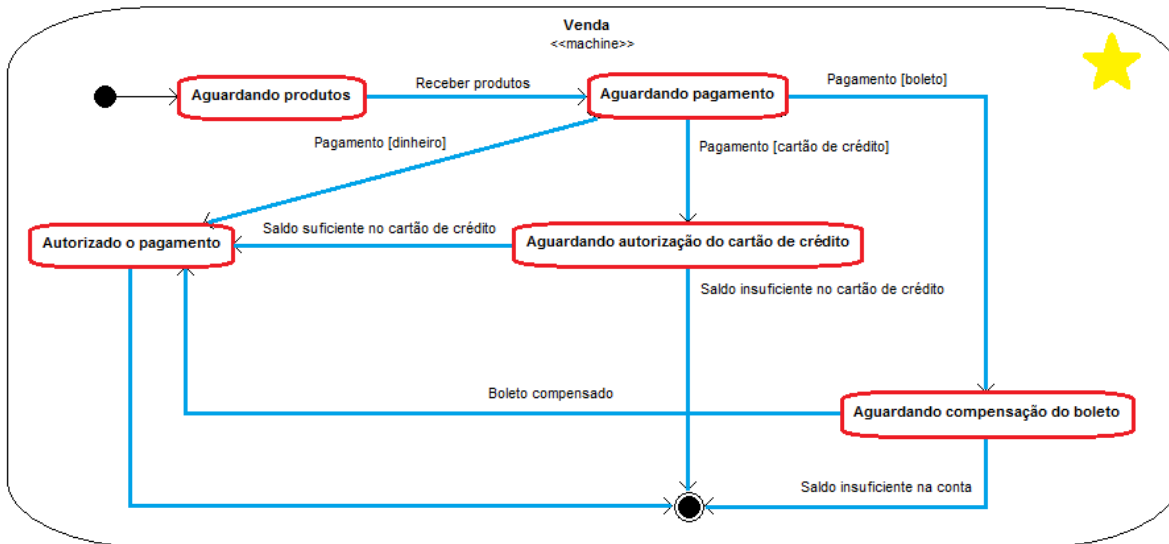


Figura 4-6 Diagrama de estados - Passo 1 da Técnica *Shiô*



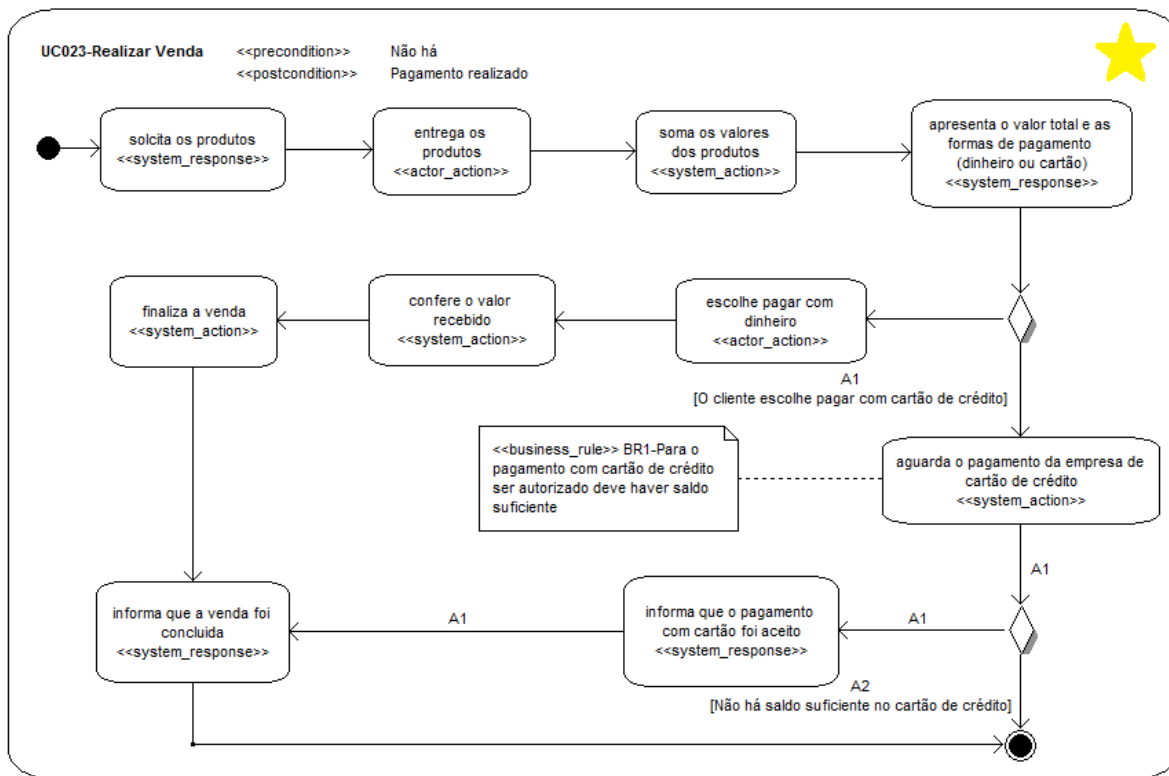


Figura 4-7 Diagrama de atividades - Passo 1 da Técnica Shiô

## Segundo Passo

No segundo passo da técnica são realizadas as marcações e colorações nos subconjuntos dos diagramas de atividades identificados no primeiro passo. Assim, os inspetores são orientados a percorrer todos os diagramas de atividades identificados anteriormente e realizar marcações e identificações com *labels* distintos em cada um dos recursos sintáticos do diagrama de atividades com cores pré-definidas pela técnica. As estruturas marcadas e identificadas são: ações com o estereótipo de *Respostas do Sistema* e *Ação do Sistema*, ações com o estereótipo *Ação do Ator*, condições de guarda nos diferentes fluxos de controle, regras de negócio e pré e pós-condições.

A Figura 4-8 apresenta o diagrama de atividades devidamente marcado e colorido, onde foram utilizadas as seguintes cores nos determinados recursos sintáticos:

- Vermelho: ações com o estereótipo Resposta do Sistema
- Azul: ações com os estereótipos Ação do Sistema e Ação do Ator, regras de negócio e fluxos entre ações que possuem condições de guarda/restrições;
- Verde: pré e pós-condições.

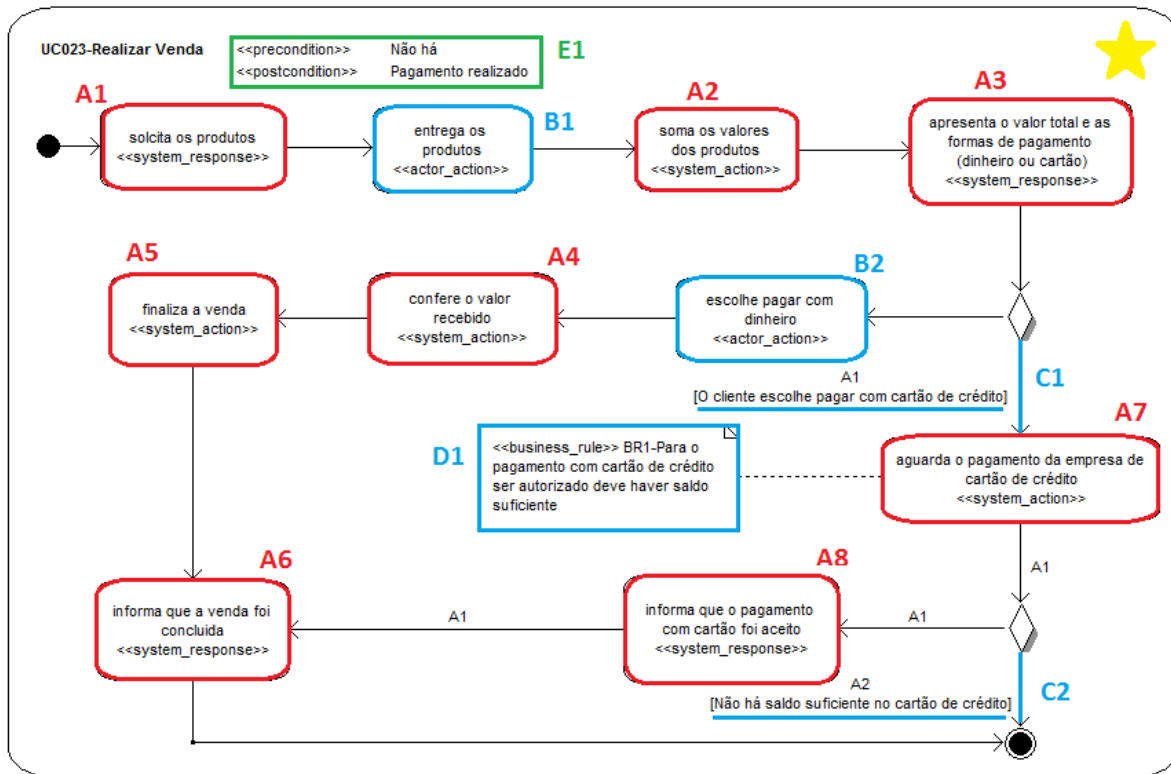


Figura 4-8 Diagrama de atividades - Passo 2 da Técnica Shiô

Instruções com relação ao conceito de subdiagramas de atividades também são apresentadas. Assim, o instrutor decide se há a necessidade de verificar os detalhes contidos nos subdiagramas ou se as informações contidas no próprio diagrama estão de acordo com o nível de abstração necessário.

### Terceiro Passo

O terceiro passo da técnica visa encontrar os elementos das estruturas coloridas e marcadas com cores idênticas entre os diagramas de estados e de atividades. Assim, para cada recurso sintático marcado e colorido no segundo passo, a técnica orienta o inspetor a procurar no diagrama de estados os recursos sintáticos que possuem o mesmo significado do recurso em questão do diagrama de atividades. As marcações coloridas ajudam neste passo, pois as cores indicam onde o inspetor deve analisar. Quando a correspondência é encontrada, o elemento do diagrama de estados é nomeado com o mesmo *label* do elemento do diagrama de atividades. Outra marcação é realizada nos dois diagramas para indicar que foi encontrada a correspondência entre eles.

As Figura 4-9 e Figura 4-10 apresentam o diagrama de atividades e o diagrama de estados marcados, onde é ilustrada a correspondência entre os elementos dos dois diagramas. Assim, os elementos dos diagramas de estados foram nomeados com os mesmo labels indicados no diagrama de atividades que possuem o mesmo significado.

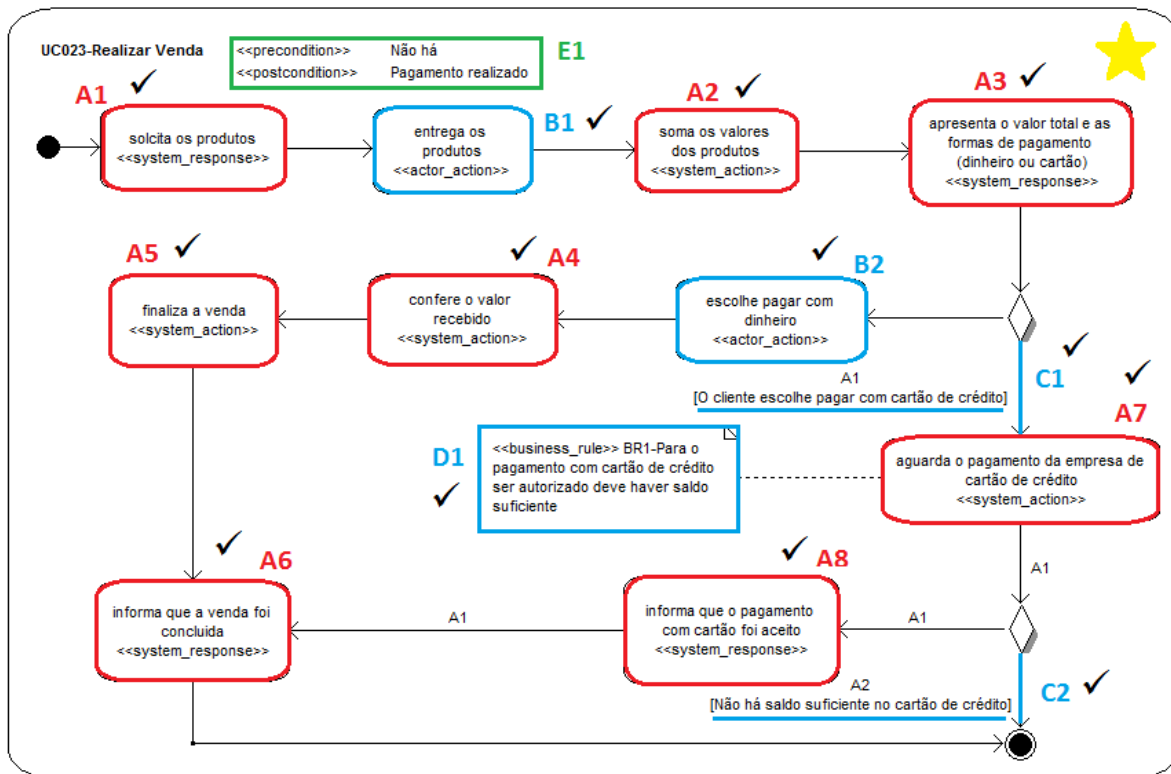


Figura 4-9 Diagrama de atividades - Passo 3 da Técnica Shiô

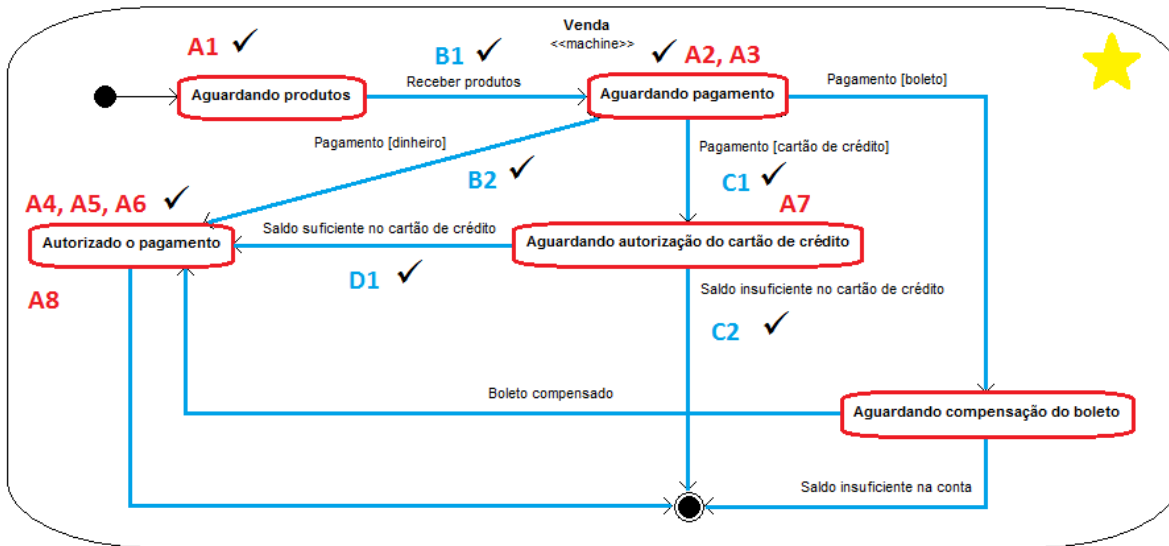


Figura 4-10 Diagrama de estados - Passo 3 da Técnica *Shiô*

Vale ressaltar que nem todos os elementos coloridos nos diagramas de atividades (passo 2) têm correspondência no diagrama de estados, visto que o diagrama de atividades está representando a descrição de uma funcionalidade, que pode ser muito mais ampla que o ciclo de vida de uma classe/objeto.

#### Quarto Passo

O quarto passo da técnica concentra na detecção e no preenchimento dos defeitos no relatório. Assim, a técnica fornece instruções para os inspetores procurarem elementos coloridos nos diagramas que não possuem a marcação de correspondência entre os elementos dos diagramas de atividades e de estados.

Neste passo o inspetor encontra diferentes instruções orientando na detecção de defeitos (omissão, informação estranha e fato incorreto) e no preenchimento do relatório de discrepância, informando os tipos de discrepâncias. As discrepâncias com classificação de inconsistência não são diretamente contempladas pela técnica, pois sendo os diagramas de atividades o oráculo da inspeção, considera-se que a diferença entre os dois diagramas deve ser classificada como fato incorreto.

A Figura 4-11 apresenta os elementos do diagrama de estados que não possuem correspondência com o diagrama de atividades. Estes elementos estão circulos em roxo e correspondem às transições e estados referentes à venda realizada por boleto. Como os diagramas de atividades são considerados os oráculos para a inspeção, os

elementos encontrados como discrepância no diagrama de estados, são consideradas informação estranha em relação ao oráculo. Assim, a técnica orienta o inspetor a relatar essas discrepância encontradas no relatório de discrepância.

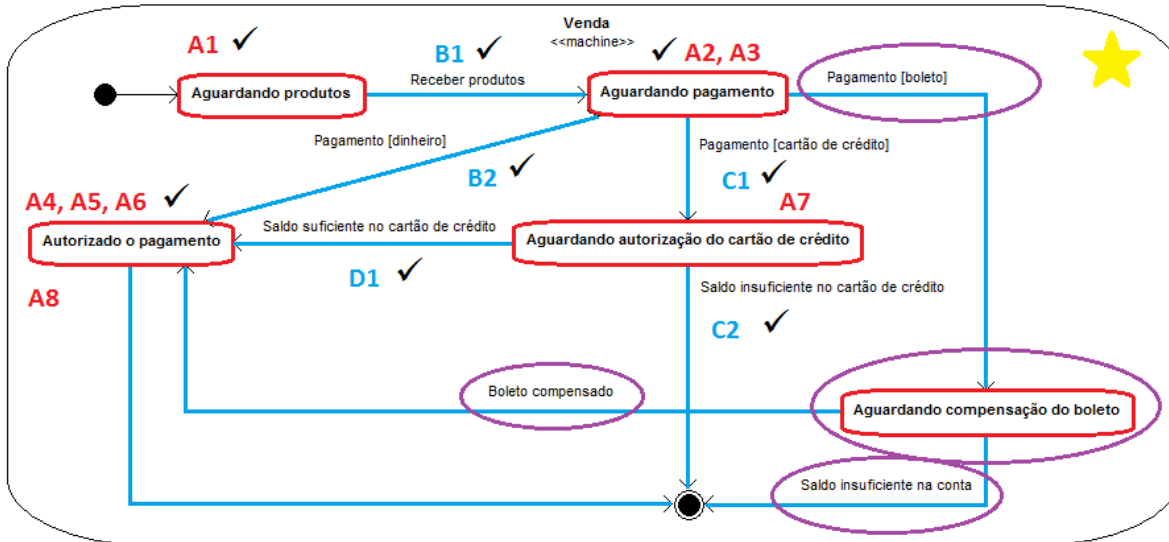


Figura 4-11 Diagrama de atividades - Passo 4 da Técnica Shiô

### Quinto Passo

No quinto e último passo da técnica é solicitado ao inspetor que utilize o seu conhecimento sobre o domínio e também de inspeções anteriores para relatar eventuais defeitos que seriam detectados de forma *ad-hoc* e que não puderam ser identificadas através do uso da técnica. Para orientar o inspetor, alguns pontos dos diagramas são ressaltados para que ele reflita e possa detectar esses defeitos.

O exemplo apresentado nessa seção é bastante simples e, portanto, não contém discrepâncias que possam ser detectadas com esse passo da técnica.

#### 4.5.2 Primeira Versão da Técnica

A primeira versão da técnica foi criada seguindo os passos descritos na seção 4.5.1 e é apresentada a seguir:

### Técnica de Leitura para Casos de Uso (UC), Diagramas de Atividades (DA) e Diagramas de estados (DE)

**Objetivo:** Verificar se os DEs que modelam o sistema estão em conformidade com os DAs que descrevem as atividades do sistema.

**Entradas para o processo:**

1. DAs que descrevem as atividades realizadas pelo sistema.
2. DEs que descrevem os estados internos de objetos do sistema.

Execute os seguintes passos:

**I.** Para **cada DE**, faça:

**ENTRADAS:** DEs e DAs

**SAÍDAS:** DEs marcados com canetas coloridas; conjunto de DAs e relatório de discrepância.

- A. Determine qual classe/objeto está sendo modelada pelo DE.
    - Se você não pode identificar a classe/objeto que está sendo modelada, então alguma informação está sendo omitida ou está ambígua. Indique isto no formulário de discrepância.
  - B. Acompanhe a sequência de estados e as ações de transição do DE. Comece pelo estado inicial (●) e siga as transições até que você encontre algum estado final (●), se houver. Esteja certo que você passou por todas as transições (diferentes caminhos). Pense sobre os estados e como eles podem estar representados em conjunto. Inclua em sua análise, se existir, todas as submáquinas e os estados compostos, considerando, portanto o DE contido em cada submáquina e os estados internos presentes nos estados compostos, que podem estar presentes.
    - Caso não exista a indicação de estado inicial, relate como uma omissão no relatório de discrepância.
    - Marque os estados utilizando uma caneta vermelha.
    - Marque as transições entre estados utilizando uma caneta azul.
    - Esteja certo que você pode entender e descrever o que está acontecendo com a classe/objeto apenas lendo o DE. Se você não pode, então o DE está ambíguo ou incompleto. Indique isto no relatório de discrepância.
  - C. Para cada DE encontre os DAs que representam a classe/objeto modelada pelo DE; buscando pelo próprio nome do DE dentro de todas as estruturas (ações, atividades, transições, condições de guarda,...) dos DAs. Fique atento para representações com nomes diferentes, mas que possuem o mesmo significado (Ex: no DA existe o conceito de carro, porém no DE a classe/objeto representada é automóvel. Carro e automóvel representam um mesmo conceito, porém com nomes distintos). Utilize algum tipo de marcação (números ou símbolos) para marcar a correspondência entre o DE e os DAs que utilizam a classe/objeto modelado pelo DE. Este subconjunto de DAs será utilizado para o restante da inspeção. Lembre-se: um DA pode pertencer a subconjuntos de diferentes DEs.
    - Caso encontre um mesmo conceito utilizando nomes diferentes em partes diferentes do diagrama ou até mesmo em diagramas diferentes, relate no formulário de discrepância como ambiguidade.
- II.** Para cada DA do subconjunto previamente identificado no passo anterior, faça:
- ENTRADA:** Subconjuntos de DAs
- SAÍDAS:** DAs marcados com canetas coloridas; relatório de discrepância.
- A. Acompanhe a sequência de ações e fluxos de controle do DA. Comece pelo estado inicial (●) e siga os fluxos de controle até que você encontre um estado final (●). Esteja certo que você passou por todas as ações e fluxos de controle.
    - Para cada subdiagrama (indicado pelo símbolo  $\uparrow$ ), avalie a granularidade da abstração, analisando a necessidade de utilização do subdiagrama para

entendimento da atividade. Caso sejam necessários maiores detalhes sobre a atividade, utilize o DA indicado pelo subdiagrama para a inspeção. Caso contrário utilize apenas o subdiagrama, representando a ação.

- Marque as ações indicadas pelos estereótipos <<system\_response>> e <<system\_action>> do DA à medida que você as encontra, utilizando uma caneta vermelha. Dê a cada ação um label único [A1, A2, ...].
- Marque as ações indicadas pelo estereótipo <<actor\_action>> do DA à medida que você as encontra, utilizando uma caneta azul. Dê a cada ação um label único [B1, B2, ...].
- Marque os fluxos entre as ações do DA que contenham pontos de decisão, loops, restrições e condições de guarda (Ex: restrição para  $t > 0$ , isto é, a transição para outra ação/atividade só deve ocorrer caso o valor de t seja maior que zero), usando uma caneta azul. Dê a cada fluxo um label único [C1, C2, ...].
- Marque as regras de negócio (Business Rules) que indiquem restrições e condições de funcionamento do sistema (Ex: restrição de tempo, restrição da condição em que o sistema ou parte dele se encontra), usando uma caneta azul. Para realizar tal tarefa, você deve utilizar seu conhecimento do domínio para identificar quais são as regras de negócio correspondentes e viáveis. Dê a cada regra de negócio um label único [D1, D2, ...].
- Marque as pré e pós-condições da atividade, caso elas indiquem o estado em que o sistema deve estar para executar o caso de uso (pré-condições) e o estado em que o sistema deve estar quando o caso de uso encerrar (pós condição), utilizando uma caneta verde. Dê a cada pré e pós-condição um label único [E1, E2, ...].

III. Para cada DE, analise todos os DAs correspondentes e marcados anteriormente, dando atenção para representações com nomes diferentes, mas que possuem o mesmo significado (vale lembrar que estas discrepâncias devem ter sido previamente encontradas e relatadas no formulário de discrepância).

**ENTRADAS:** DEs e DAs marcados com canetas coloridas

**SAÍDAS:** DEs e DAs com marcações que os relacionam

- A. Para cada ação (System Action ou System Response) ou conjunto de ações que façam semanticamente sentido (marcada em vermelho nos DAs), encontre um estado correspondente no DE (marcado em vermelho). Para fazer isto, você precisa pensar sobre a semântica associada a cada ação. Elas têm alguma relação com os possíveis estados que este objeto poderia assumir? Nomeie o estado do DE com os mesmos labels das ações dos DAs, assim o estado do DE terá os mesmos labels que as ações dos DAs. Faça uma marcação (✓) nas marcas em vermelho nos diagramas.
- B. Para cada ação (ActorAction) ou conjunto de ações que façam semanticamente sentido (marcada em azul nos DAs), encontre uma transição correspondente no DE (marcada em azul). Para fazer isto, você precisa pensar sobre a semântica associada a cada ação. Elas têm alguma relação com as possíveis transições que este objeto poderia sofrer? Nomeie a transição do DE com os mesmos labels das ações dos DAs, assim a transição do DE terá os mesmos labels que as ações dos DAs. Faça uma marcação (✓) nas marcas em azul em ambos os diagramas, indicando que existe correspondência semântica entre eles.
- C. Para cada alternativa de fluxo de controle (ponto de decisão, loop, restrição ou condição de guarda), marcado em azul nos DAs encontre uma transição correspondente no DE (marcada em azul). Nomeie a transição do DE com o

mesmo label do fluxo de controle dos DAs, assim a transição do DE terá os mesmos labels que as ações dos DAs. Faça uma marcação (✓) nas marcas em azul em ambos os diagramas, indicando que existe correspondência semântica entre eles.

- D. Para cada regra de negócio encontrada nos DAs (marcada em azul), encontre uma transição correspondente no DE (marcada em azul). Nomeie a transição do DE com o mesmo label da regra de negócio dos DAs, assim a transição do DE terá o mesmo label que a regra de negócio dos DAs. Faça uma marcação (✓) nas marcas em azul em ambos os diagramas, indicando que existe correspondência semântica entre eles.

- IV. Para cada DE com os seus DAs correspondentes, devidamente marcados, analise:

**ENTRADAS:** DEs e DAs marcados com canetas coloridas e com marcações (✓)

**SAÍDA:** relatório de discrepância

- A. Considerando as ações com estereótipos System Action e System Response (marcadas em vermelho):

- Caso exista marcação em vermelho nos DAs sem a marcação (✓) e que represente estados no DE, então existem estados no DE que não estão sendo descritos. Relate no formulário de discrepância como omissão.
- Caso exista marcação em vermelho no DE sem a marcação (✓), relate no formulário de discrepância como informação estranha.
- Utilizando seu conhecimento sobre o domínio, verifique se existem marcações em vermelho nos DAs sem a marcação (✓) e que determinam um estado ou a alteração de estado da classe/objeto, porém a correspondência no DE indica um estado ou alteração de estados diferente da esperada. Relate como fato incorreto no relatório de discrepância.

- B. Considerando as ações com estereótipos ActorAction (marcadas em azul):

- Caso exista marcação em azul nos DAs sem a marcação (✓) e que represente alterações de estados (transições), relate no formulário de discrepância como omissão.
- Caso exista marcação em azul sem a marcação (✓) e que determine uma alteração de estado (transição) da classe/objeto, porém no DE correspondente está mapeado para uma transição com valor diferente do que está representado pelo DA, relate como fato incorreto no relatório de discrepância.

- C. Considerando os pontos de decisão, loops, restrições ou condições de guarda (marcados em azul):

- Caso exista marcação em azul nos DAs sem a marcação (✓) e que represente estados no DE, relate no formulário de discrepância como omissão.
- Caso exista marcação em azul nos DAs sem a marcação (✓) incompatível com as informações entre transições no DE com as condições de guarda e restrições nos DAs (ex: valores diferentes para restrições de tempo), relate no relatório de discrepância como fato incorreto.

- D. Considerando as regras de negócio (marcadas em azul):

- Caso exista marcação em azul nos DAs e sem a marcação (✓), relate no relatório de discrepância como omissão.
- Caso exista marcação em azul nos DAs sem a marcação (✓) incompatível com as informações entre transições no DE representando regras de



- negócio com as regras de negócio nos DAs (ex: valores diferentes para condições em que o sistema se encontra), relate no formulário de discrepância como fato incorreto.
- Caso exista marcação em azul no DE e sem a marcação (✓), relate no relatório de discrepância como informação estranha. As transições no DE podem corresponder as ações do tipo ActorAction, pontos de decisão, loops, restrições, condições de guardas ou regras de negócio do DAs, e como não foi marcada (✓) em nenhum desses construtos, pode existir alguma informação estranha no DE.
- E. Considerando as pré e pós-condições (marcadas em verde), encontre a ordem cronológica dos estados no ciclo de vida da classe/objeto que esteja presente nas cláusulas de pré e pós-condições. Ou seja, as marcações em verde nos DAs deve indicar qual estado a classe/objeto deve estar para iniciar ou terminar a execução da atividade. Assim no DE o estado indicado na pré-condição corresponde a algum estado anterior ao estado apresentado como pós-condição.
- Caso a ordem dos estados esteja trocada ou diferente das apresentadas nas pré e pós-condições, relate no formulário de discrepância como fato incorreto.
- V. Reveja os diagramas utilizados nesta inspeção
- ENTRADAS:** DEs e DAs marcados com canetas coloridas e com marcações (✓)
- SAÍDA:** relatório de discrepância
- A. Utilizando o seu conhecimento sobre o domínio e sua experiência em modelagem, existe alguma classe/objeto, estado ou transição no DE que esteja faltando e que seja importante para o sistema.
- Caso exista, relate no formulário de discrepância como omissão.
- B. Você consegue identificar algum nome, conceito ou qualquer outro componente no diagrama que esteja diferente do seu conhecimento de domínio?
- Caso a resposta seja positiva, relate no formulário de discrepância como fato incorreto.
- C. Pensando no que você entende do domínio, as sequências de transições e os estados, podem efetivamente representar o ciclo de vida da classe/objeto participante do sistema?
- Caso a resposta seja negativa, relate no formulário de discrepância como fato incorreto.

### 4.5.3 Relatório de Discrepância

O modelo de relatório de discrepância utilizado pela técnica de leitura considera que a inspeção foi realizada de forma individual e que, para cada diagrama de estados inspecionado, existe um relatório de discrepância. O cabeçalho do relatório contém campos para preenchimento com dados relativos à inspeção, tais como:

- Data
- Nome do inspetor
- Horário do início da inspeção

- Horário do fim da inspeção
- Tempo total da inspeção contabilizado em minutos
- Diagrama de estados
- Subconjunto dos diagramas de atividades que foram utilizados na inspeção, identificados no final do primeiro passo da técnica.

Cada discrepância detectada deve ser descrita como uma entrada na tabela que está presente no relatório de discrepância. Assim, para cada discrepância detectada, o inspetor deve preencher:

- **Identificador da discrepância:** deve ser atribuído um nº distinto para cada discrepância, começando pelo nº 1;
- **Passo/Item:** este campo deve ser preenchido com o nº do passo (I, II, III, IV ou V), seguido da letra (A, B, C, D ou E) em que foi detectada a discrepância. Ex: IV/A;
- **Categoria de defeito:** a técnica de leitura fornece nas instruções a classificação da discrepância detectada. Nesse campo, o inspetor deve transcrever esta classificação;
- **Label DE/DA:** esse campo deve ser preenchido com o *label* do elemento do diagrama de atividades em que foi encontrada a discrepância. Esse campo é fundamental para a fase de análise das discrepâncias, pois com o relatório de discrepância e com os modelos marcados pelo inspetor, é possível avaliar qual elemento do diagrama estava sendo observado quando o inspetor detectou a discrepância;
- **Diagrama de Atividades:** como o subconjunto de diagramas de atividades que fazem uso da classe/objeto modelada pelo diagrama de estados pode conter mais de um diagrama, esse campo serve para identificar qual o diagrama de atividades utilizado quando o inspetor detectou a discrepância;
- **Descrição:** esse campo fica livre para que o inspetor possa realizar um breve comentário sobre a discrepância encontrada, caso julgue necessário.

O modelo do relatório de discrepância pode ser visualizado no Apêndice C.

## 4.6 Conclusão

Este capítulo apresentou a primeira versão da técnica de leitura para inspeção em diagramas de estados (*Shiô*) utilizando como referência a especificação de casos de uso modelados em diagramas de atividades. As ideias iniciais e as justificativas pela escolha dos diagramas de atividades e de diagramas de estados para serem inspecionados foram apresentadas, assim como o posicionamento da técnica proposta com relação à OORTs (TRAVASSOS *et al.*, 1999) foi descrito e explicado neste capítulo.

As premissas definidas para a aplicação da técnica foram descritas para restringir e, por consequência, definir, o escopo de onde a técnica de inspeção pode ser utilizada. A relação de cada elemento dos diagramas de atividades com elementos dos diagramas de estados foram expostos e essas relações foram fundamentais e utilizadas como base para a criação da técnica de leitura.

Ainda neste capítulo, a primeira versão da técnica de leitura de diagramas de estados com apoio de diagramas de atividades representando casos de uso foi apresentada. A técnica é constituída de cinco passos com uma série de orientações que devem ser seguidas de forma sequencial. Cada um desses passos foi explicado e posteriormente a técnica foi apresentada na íntegra. A técnica realiza a comparação dos dois diagramas (atividades e estados), porém é importante lembrar que os diagramas de estados poderiam ser gerados através dos diagramas de atividades, visto que todos os diagramas de atividades do projeto deveriam considerar todos os estados e transições das classes/objetos. O relatório de discrepância preenchido durante a aplicação da técnica de leitura é exposto, onde cada campo do relatório é descrito.

O Capítulo 5 apresenta dois estudos experimentais realizados em busca de evidência sobre a viabilidade da aplicação de *Shiô* em diagramas de estados.

## 5 Estudos Experimentais

*Neste capítulo são apresentados os resultados de dois estudos experimentais conduzidos para avaliar a viabilidade da técnica de leitura Shiô para inspecionar diagramas de estados utilizando os diagramas de atividades especificando casos de uso. O planejamento dos estudos, os resultados e a análise dos dados obtidos também são apresentados, assim como as lições aprendidas e as ameaças à validade dos estudos.*

### 5.1 Introdução

A técnica de leitura *Shiô* apresentada no Capítulo 4 foi desenvolvida com o objetivo de inspecionar diagramas de estados utilizando como referência diagramas de atividades que descrevem a especificação de casos de uso. A técnica faz parte de OORTs (Object Oriented Reading Techniques) (TRAVASSOS *et al.*, 2002), que representa uma família de técnicas de leitura para inspecionar diagramas UML. A técnica proposta nessa dissertação considera algumas premissas, definidas na seção 4.3, para o seu uso:

- Requisitos e casos de usos devem estar previamente inspecionados;
- Diagramas de atividades estão modelados de acordo com *UseCaseAgent* (MASSOLLAR, 2011);
- Diagramas de atividades devem estar previamente inspecionados, utilizando *ActCheck* (DE MELLO, 2011) ou outra técnica de inspeção, ou seja, assume-se que são válidos e corretos;
- Diagramas de classes, diagramas de sequência e diagramas de estados devem representar o mesmo sistema.

Dois estudos *in-vitro* foram executados, entre maio e junho de 2012, em diferentes contextos para avaliar a viabilidade da aplicação de *Shiô*. O primeiro foi executado na UFRJ (Universidade Federal do Rio de Janeiro) e o segundo em uma universidade particular na cidade do Rio de Janeiro. Os dois estudos visam avaliar diferentes aspectos

relacionados ao desempenho da técnica *Shiô* quando comparada com inspeção *ad-hoc*. Ambos os estudos utilizaram modelos pertencentes ao MFI (Módulo Financeiro) de um projeto de aplicação real, desenvolvido pelo Grupo de Engenharia de Software Experimental da COPPE/UFRJ para a fundação COPPETEC.

Os modelos utilizados representam diagramas de atividades construídos com base na técnica de modelagem proposta por MASSOLLAR (2011). Esses diagramas de atividades descrevem requisitos (casos de usos) que foram previamente inspecionados com *ActCheck* (DE MELLO, 2011). Nos estudos foram utilizados dois conjuntos de modelos pertencentes ao mesmo módulo do sistema: Conta e Movimento.

O conjunto de modelos de Conta é composto por:

- 1 diagrama de estados: Conta
- 1 diagrama de atividades: Administrar conta

O conjunto de modelos de Movimento é composto por:

- 1 diagrama de estados: Movimento
- 5 diagramas de atividades: Cancelar movimentos; Consultar movimentos; Efetuar pagamento; Estornar movimentos e Liberar movimentos.

O conjunto de modelos Conta possui complexidade menor do que o conjunto de modelos de Movimento, tanto pela quantidade de diagramas envolvidos, quanto pela complexidade dos conceitos envolvidos nos modelos. Todos os modelos utilizados nos diagramas podem ser visualizados no Apêndice B desta dissertação.

Na seção 5.2 é apresentado os detalhes do primeiro estudo executado e os detalhes do segundo estudo são apresentados na seção 5.3. Uma análise qualitativa dos estudos é exposta na seção 5.4 e na seção 5.5 e 5.6 são apresentadas, respectivamente, as lições aprendidas e as ameaças à validade identificadas. A segunda versão da técnica é apresentada na seção 5.7 e por fim, a conclusão é apresentada na seção 5.8.

## **5.2 Primeiro Estudo**

Os objetivos específicos do primeiro estudo *in-vitro* e executado na UFRJ com a finalidade de avaliar a viabilidade da técnica *Shiô* são apresentados na Tabela 5-1.

Tabela 5-1 Objetivos específicos do primeiro estudo.

<b>Analisar a</b>	A inspeção utilizando as técnicas <i>Shiô</i> e <i>ad-hoc</i> .
<b>Com propósito de</b>	Caracterizar
<b>Com respeito ao</b>	Desempenho das técnicas na detecção de defeitos (eficácia e eficiência)
<b>Do ponto de vista do</b>	Pesquisador
<b>No contexto de</b>	Alunos de graduação na disciplina de Engenharia de Software Orientado a Objeto da UFRJ e alunos da Pós Graduação na disciplina de Engenharia de Software Experimental do PESC/COPPE/UFRJ, utilizando modelos reais de um projeto de um sistema de informação Web.

O estudo foi realizado com duas turmas distintas: estudantes da graduação e de pós-graduação. Os estudantes das turmas tinham os conhecimentos necessários para participarem do estudo, tais como: diagramas da UML e inspeção. Os conceitos relacionados à inspeção já haviam sido apresentados na disciplina da turma da graduação e os mesmos realizaram inspeções como trabalhos regulares dentro da disciplina. A disciplina da graduação em que ocorreu o estudo é uma das disciplinas dos últimos períodos do curso. Os formulários de caracterização, respondidos por todos os participantes, também indicaram que todos tinham conhecimento sobre inspeção e diagramas UML, adquiridos na disciplina ou em projetos na academia.

### 5.2.1 Planejamento

A finalidade do estudo foi responder duas questões de pesquisa, e para cada questão foram definidas duas hipóteses (nula e alternativa):

1. A inspeção com a técnica *Shiô* encontra mais defeitos (eficácia) que com *ad-hoc*?
  - $H_0$ : Não existe diferença no número de defeitos detectados na inspeção com a técnica *Shiô* e a técnica *ad-hoc*.
  - $H_1$ : O número de defeitos detectados com a técnica *Shiô* é maior que o número de defeitos obtidos com a técnica *ad-hoc*.
2. A quantidade de defeitos/tempo da técnica *Shiô* é maior que a inspeção *ad-hoc*?
  - $H_0$ : Não existe diferença na relação de eficiência da aplicação da técnica *Shiô* quando comparado com *ad-hoc*.

- H<sub>1</sub>: A eficiência da técnica *Shiô* é maior do que *ad-hoc*.

O estudo foi conduzido para obter dados quantitativos e para responder às questões foram extraídas as seguintes métricas: quantidade de defeitos detectados e o tempo dedicado à inspeção. Como variáveis dependentes neste estudo foram identificadas a quantidade de defeito e o tempo dedicado à realização da inspeção. As variáveis independentes definidas foram: a técnica de inspeção, os modelos utilizados, a experiência dos inspetores em inspeções anteriores e o conhecimento prévio dos inspetores sobre o domínio dos modelos inspecionados.

O estudo visava responder algumas questões, com base nas seguintes métricas:

1. A técnica *Shiô* encontra mais defeitos que a inspeção *ad-hoc*?
  - Número de defeitos: contagem do número de defeitos encontrados;
2. A eficiência da técnica *Shiô* é maior que a inspeção *ad-hoc*?
  - Número de defeitos: contagem do número de defeitos encontrados;
  - Tempo despendido: tempo dedicado às inspeções.

No estudo há questões em aberto, ou seja, que não são possíveis de serem respondidas com estes estudos. São elas:

1. Os inspetores estão realmente utilizando a técnica *Shiô* corretamente?
2. A técnica é viável para projetos em contextos diferentes do contexto em que o estudo foi aplicado?

Todos os participantes do estudo receberam um termo de consentimento e um formulário de caracterização, apresentados no Apêndice A. O formulário de caracterização possui algumas perguntas que têm por objetivo identificar o grau de experiência dos participantes em alguns quesitos. Esses formulários respondidos serviram para classificar e dividir os participantes quanto à experiência em escala ordinal com valores: baixa, média e alta.

### **5.2.2 Projeto Experimental**

No primeiro estudo, a turma de graduação forneceu inicialmente 26 participantes, enquanto que a turma de pós-graduação forneceu inicialmente 9 participantes. Todos os participantes responderam os formulários de consentimento e os formulários de caracterização.

Os participantes das duas turmas foram divididos quanto à experiência, baseado nos formulários de caracterização respondidos e em cada uma das turmas foi criado dois grupos com tamanhos equivalentes (Grupo 1 e Grupo 2). Os grupos de cada turma continham participantes com diferentes níveis de experiência, ou seja, participantes com baixa, média e alta experiência. Esse artifício, denominado *blocking*, foi utilizado com o intuito de reduzir os efeitos da experiência dos inspetores, assim cada um dos grupos ficou balanceado com relação à experiência.

O estudo foi agrupado em duas rodadas com dois conjuntos de modelos, onde na primeira rodada um grupo inspecionava um dos conjuntos de modelos e na rodada seguinte, o conjunto de modelos era invertido entre os grupos. A turma da pós-graduação inspecionou em ambas as rodadas de forma *ad-hoc*, enquanto que a turma da graduação inspecionou apenas utilizando a técnica *Shiô* nas duas rodadas. Nas duas turmas foi utilizado o mesmo arranjo, apresentado na Tabela 5-2, variando apenas a técnica de inspeção aplicada em cada uma delas.

Tabela 5-2 Arranjo do primeiro estudo.

	<b>Rodada 1</b>	<b>Rodada 2</b>
<b>Grupo 1</b>	Conta	Movimento
<b>Grupo 2</b>	Movimento	Conta

O objetivo do arranjo, mostrado na Tabela 5-2, foi comparar o desempenho da técnica *Shiô* com a técnica de inspeção *ad-hoc*. Assim, optou-se em definir que a turma da pós-graduação inspecionaria apenas de forma *ad-hoc*, devido ao fato de serem mais experientes que os estudantes da graduação. Este fato pode ser observado pelos próprios formulários de caracterização respondidos por todos os participantes do estudo.

Como citado anteriormente, as questões do formulário de caracterização foram utilizadas para dividir os participantes de cada turma em grupos. Os critérios utilizados para essa divisão foram: experiência com desenvolvimento de software na prática e experiência com inspeção de software na prática. As questões relativas à experiência em modelagem e inspeção de diagramas de atividades e de estados não puderam ser utilizadas, devido ao fato das turmas, principalmente da graduação, apresentarem experiência homogênea, impossibilitando a classificação dos participantes como: baixa, média ou alta experiência.



O resultado desejado do estudo é que a técnica *Shiô* auxilie os inspetores menos experientes (graduação) na inspeção, de forma que o seu desempenho seja melhor que a inspeção *ad-hoc* executada por inspetores mais experientes (pós-graduação). Outro objetivo do arranjo foi de analisar o efeito do aprendizado da técnica de inspeção (*Shiô* ou *ad-hoc*) entre as rodadas. O desempenho dos inspetores normalmente melhora no decorrer das rodadas de inspeções, visto que os inspetores adquirem conhecimento sobre o domínio e a técnica de inspeção em si.

### 5.2.3 Instrumentação

Os instrumentos preparados e utilizados para apoiar o estudo foram:

1. Material de treinamento sobre inspeção utilizando a técnica *Shiô*;
2. Material de treinamento sobre inspeção *ad-hoc*;
3. Técnica *Shiô*;
4. Planilha eletrônica para registrar o tempo dedicado à inspeção (em minutos) e as discrepâncias encontradas;
5. Diagramas de atividades referentes a Movimento;
6. Diagrama de atividades referentes a Conta;
7. Diagrama de estados referentes a Movimento;
8. Diagrama de estados referentes a Conta;
9. Regras de negócio dos diagramas de atividades referentes a Movimento e
10. Regras de negócio dos diagramas de atividades referentes a Conta.

### 5.2.4 Execução

#### Turma da pós-graduação

A turma inteira da pós-graduação recebeu o treinamento sobre inspeção *ad-hoc*, com duração de aproximadamente uma hora. Dois pacotes de inspeção foram criados, diferenciando apenas nos conjuntos dos modelos, onde em um pacote estavam os modelos referentes a Conta e no outro, estavam os modelos referentes a Movimento. Além dos modelos, cada um dos pacotes continha: o treinamento aplicado, os diagramas de atividades, o diagrama de estados, os documentos com as regras de negócio e o relatório de discrepância. Após o treinamento, os participantes receberam por e-mail um pacote com um dos conjuntos de diagramas a serem inspecionados. A Tabela 5-3 apresenta os participantes de cada grupo, o nível de experiência em que foi classificado e

os artefatos inspecionados em cada rodada pelos participantes da turma da pós-graduação.

Tabela 5-3 Distribuição dos participantes e dos artefatos da turma da pós-graduação.

Grupo	Participante	Nível de Experiência	Rodada 1	Rodada 2
Grupo 1	P1	Baixa	Conta	Movimento
	P2	Média		
	P4	Média		
	P6	Baixa		
	P9	Alta		
Grupo 2	P3	Baixa	Movimento	Conta
	P5	Baixa		
	P7	Média		
	P8	Alta		

No treinamento, os inspetores foram instruídos a inspecionar de forma *ad-hoc* os modelos enviados, ou seja, nenhuma técnica ou heurística foi apresentada para que a inspeção fosse realizada. Eles utilizaram apenas o conhecimento de inspeções anteriores e seu conhecimento prévio sobre o domínio. A hora de início e do fim da inspeção e, por consequência, o tempo total dedicado à inspeção deveriam ser marcados (em minutos) e indicados no relatório de discrepância. Os inspetores foram instruídos para que interrupções fossem evitadas, a fim de obter o tempo de inspeção mais preciso possível. Em cada uma das rodadas, os inspetores tiveram um prazo de dois dias para realizar a inspeção, preencher o relatório de discrepância e enviar o resultado da inspeção por e-mail. O segundo pacote de inspeção foi somente enviado após a entrega da primeira inspeção pelo inspetor por e-mail ou pessoalmente. Na segunda rodada também foi estabelecido o mesmo prazo de dois dias para a entrega das inspeções.

Todos os participantes entregaram as inspeções em ambas as rodadas e após obter todos os relatórios de discrepância, os mesmos foram analisados para identificar quais as discrepâncias apontadas eram realmente defeitos e, por consequência, quais eram falsos positivos. Uma lista de defeitos foi criada com todos os defeitos distintos encontrados pelos inspetores da pós-graduação. Essa lista foi considerada como *baseline*

para a comparação com a lista de defeitos gerada pela turma da graduação, visto que além dos estudantes da pós-graduação serem mais experientes do que os estudantes da graduação, não havia uma lista prévia de defeitos conhecidos dos conjuntos de modelos.

### **Turma da graduação**

Os alunos da graduação receberam um treinamento diferente da turma de pós-graduação, pois nesse treinamento os alunos receberam informações e um exemplo didático do funcionamento da técnica de inspeção *Shiô*. Os pacotes utilizados na turma da graduação foram bem similares àqueles utilizados com a turma de pós-graduação, diferindo no treinamento que foi aplicado, em pequenas mudanças no relatório de discrepância e na adição da técnica de inspeção *Shiô*. Assim como na turma de pós-graduação, os inspetores receberam por e-mail um pacote com um dos conjuntos de modelos a serem inspecionados. A Tabela 5-4 apresenta os participantes de cada grupo, o nível e experiência em que foi classificado e os artefatos inspecionados em cada rodada pelos participantes da turma da graduação.

Tabela 5-4 Distribuição dos participantes e dos artefatos da turma da graduação.

<b>Grupo</b>	<b>Participante</b>	<b>Nível de Experiência</b>	<b>Rodada 1</b>	<b>Rodada 2</b>
Grupo 1	G1	Baixa	Conta	Movimento
	G3	Média		
	G7	Média		
	G8	Baixa		
	G10	Baixa		
	G12	Baixa		
	G14	Alta		
	G16	Média		
	G19	Média		
	G20	Baixa		
	G22	Baixa		
	G23	Baixa		
G25	Alta			
Grupo 2	G2	Média	Movimento	Conta

	G4	Baixa		
	G5	Média		
	G6	Alta		
	G9	Baixa		
	G11	Baixa		
	G13	Alta		
	G15	Baixa		
	G17	Baixa		
	G18	Média		
	G21	Baixa		
	G24	Média		
	G26	Média		

No treinamento os inspetores foram instruídos a realizarem a inspeção aplicando a técnica mostrada durante o treinamento, além do conhecimento prévio do domínio. Assim como a turma da pós-graduação, a hora de início, de fim e o tempo (em minutos) dedicado à inspeção deveriam ser marcados e indicados no relatório de discrepância. As interrupções durante a inspeção também deveriam ser evitadas para que a medição do tempo de inspeção fosse a mais precisa possível. Os inspetores desta turma tiveram um prazo de dois dias, igual ao da outra turma, pois com prazos e modelos iguais, uma comparação pode ser realizada com relação ao desempenho da inspeção *ad-hoc* e da inspeção utilizando *Shiô*. Assim como na outra turma, o segundo pacote de inspeção somente era enviado mediante a entrega da inspeção do primeiro pacote.

Os alunos foram instruídos a realizarem as marcações nos modelos e os mesmos deveriam ser entregues juntamente com o relatório de discrepâncias devidamente preenchido. A forma de marcação (colorir) ficou livre para os alunos decidirem, foi sugerido no treinamento o uso da ferramenta *paint*, nativa do sistema operacional *Windows*, mas as marcações poderiam ser feitas com canetas coloridas nos modelos impressos no papel. Houve entregas das duas formas e, ao final da primeira rodada, 23 alunos entregaram as inspeções, sendo que na segunda rodada 3 inspetores não entregaram e, portanto, foram entregues 20 inspeções. Vale ressaltar que a participação no estudo era voluntária e não contabilizava crédito para a avaliação da disciplina, como indicado no formulário de consentimento preenchido antes da execução do estudo. Nessa turma também foi gerada uma lista com todos os defeitos distintos relatados pelos inspetores da graduação.

### 5.2.5 Análise Preliminar dos Resultados

Na entrega da primeira rodada, foi constatado que o inspetor G1, pertencente ao Grupo 1 da graduação, entregou a inspeção dos modelos de Movimento, conjunto de modelos que deveria ser inspecionado apenas na segunda rodada. Um e-mail foi enviado ao inspetor, para entender este fato, visto que o pacote de Contas foi enviado ao inspetor na primeira rodada. Em resposta ao e-mail, o inspetor G1 afirmou que pegou o pacote de inspeção com outro inspetor, assim, para não descartar o participante, o inspetor G1 foi incorporado ao Grupo 2.

O prazo para entrega das inspeções foi estendido em alguns dias, nas duas turmas, assim na turma de pós-graduação, todos entregaram; mas na turma da graduação alguns inspetores não retornaram o resultado da inspeção. Na turma da graduação, na primeira rodada, os inspetores G4, G18 e G26 não responderam. Na segunda rodada, na turma da graduação, os inspetores que não entregaram na primeira rodada não receberam o segundo pacote de inspeção e os inspetores G3, G13 e G25 não entregaram o resultado da inspeção da segunda rodada.

Em ambas as turmas (pós-graduação e graduação) que participaram do estudo houve casos de inspetores que relataram apenas falsos positivos, ou seja, nenhum defeito encontrado de fato. Partindo da premissa que inspeção encontra defeitos, esses inspetores que identificaram apenas falsos positivos, foram considerados *outliers* e, portanto, foram excluídos da análise de dados. A Tabela 5-5 ilustra os *outliers* da turma da pós-graduação, onde é possível perceber que em ambas as rodadas o inspetor P5 foi considerado *outlier*. A Tabela 5-6 ilustra os *outliers* da turma da graduação, onde é possível analisar que aparentemente muitos inspetores não utilizaram de fato a técnica, visto que os modelos continham defeitos e se a técnica fosse aplicada corretamente encontraria pelo menos algum defeito.

Tabela 5-5 *Outliers* da turma da pós-graduação.

Rodada	Participante	Modelo	Nível de Experiência
Rodada 1	P4	Conta	Média
	P5	Movimento	Baixa
Rodada 2	P5	Conta	Baixa
	P8	Conta	Alta

Tabela 5-6 *Outliers* da turma da graduação.

Rodada	Participante	Modelo	Nível de Experiência	
Rodada 1	G7	Conta	Média	
	G10		Baixa	
	G12		Baixa	
	G14		Alta	
	G16		Média	
	G20		Baixa	
	G5	Movimento	Média	
	G6		Alta	
	G11		Baixa	
	G21		Baixa	
	Rodada 2	G2	Conta	Média
		G6		Alta
G9		Baixa		
G11		Baixa		
G21		Baixa		
G8		Movimento	Baixa	
G16			Média	
G19			Média	
G20			Baixa	
G22			Baixa	

A lista de defeitos distintos apontados pela turma da graduação foi comparada com o *baseline*. A lista de defeitos distintos apontados pela turma de pós-graduação. Os inspetores da pós-graduação relataram um total de 8 defeitos no modelo de Contas e 14 defeitos nos modelos de Movimento. As Tabela 5-7 e Tabela 5-8 mostram para cada inspetor a quantidade de discrepâncias apontadas, defeitos encontrados, tempo despendido para a realização das inspeções e dados como eficiência. Neste contexto eficiência foi considerado o número de defeitos encontrados sobre o tempo dedicado à inspeção e duas métricas como eficácia, onde a primeira representa os defeitos encontrados sobre as discrepâncias apontadas e a segunda métrica se refere aos

defeitos encontrados sobre o total de defeitos distintos encontrados de cada modelo por cada turma.

Tabela 5-7 Dados da inspeção *ad-hoc*.

Grupo	Participante	Modelo	Rodada	Disc	Def	Tempo (min)	Eficiência (def/tempo)	Eficácia (def/disc)	Eficácia (def/total def)
Grupo 1	P1	Conta	1	14	7	20	0,35	0,5	0,875
Grupo 1	P2	Conta	1	12	3	52	0,057	0,25	0,375
Grupo 1	P6	Conta	1	3	3	32	0,093	1	0,375
Grupo 1	P9	Conta	1	5	5	40	0,125	1	0,625
Grupo 1	P1	Movimento	2	9	5	43	0,116	0,555	0,357
Grupo 1	P2	Movimento	2	4	4	59	0,067	1	0,285
Grupo 1	P4	Movimento	2	9	5	75	0,066	0,555	0,357
Grupo 1	P6	Movimento	2	8	5	73	0,068	0,625	0,357
Grupo 1	P9	Movimento	2	5	4	70	0,057	0,8	0,285
Grupo 2	P3	Movimento	1	15	2	136	0,014	0,133	0,142
Grupo 2	P7	Movimento	1	8	8	88	0,090	1	0,571
Grupo 2	P8	Movimento	1	4	3	115	0,026	0,75	0,214
Grupo 2	P3	Conta	2	7	6	53	0,113	0,857	0,75
Grupo 2	P7	Conta	2	11	7	69	0,101	0,636	0,875

Tabela 5-8 Dados da inspeção com o uso da técnica *Shiô*.

Grupo	Participante	Modelo	Rodada	Disc	Def	Tempo (min)	Eficiência (def/tempo)	Eficácia (def/disc)	Eficácia (def/total def)
Grupo 1	G19	Conta	1	8	6	63	0,095	0,75	0,545
Grupo 1	G22	Conta	1	13	5	185	0,027	0,384	0,454
Grupo 1	G23	Conta	1	4	1	64	0,015	0,25	0,090
Grupo 1	G8	Conta	1	8	2	102	0,019	0,25	0,181
Grupo 1	G7	Movimento	2	1	1	71	0,014	1	0,062
Grupo 1	G10	Movimento	2	3	2	108	0,018	0,666	0,125
Grupo 1	G12	Movimento	2	2	2	180	0,011	1	0,125
Grupo 1	G14	Movimento	2	4	4	237	0,016	1	0,25
Grupo 1	G23	Movimento	2	4	4	95	0,042	1	0,25
Grupo 2	G1	Movimento	1	10	3	160	0,018	0,3	0,187
Grupo 2	G2	Movimento	1	27	7	334	0,020	0,259	0,437
Grupo 2	G9	Movimento	1	23	2	180	0,011	0,086	0,125
Grupo 2	G15	Movimento	1	4	1	120	0,008	0,25	0,062
Grupo 2	G17	Movimento	1	17	13	86	0,151	0,764	0,812
Grupo 2	G24	Movimento	1	15	11	360	0,030	0,733	0,687
Grupo 2	G1	Conta	2	9	4	160	0,025	0,444	0,363
Grupo 2	G15	Conta	2	13	8	60	0,133	0,615	0,727
Grupo 2	G17	Conta	2	9	7	62	0,112	0,777	0,636
Grupo 2	G24	Conta	2	18	4	73	0,054	0,222	0,363
Grupo 2	G5	Conta	2	27	4	206	0,019	0,148	0,363

A Figura 5-1 mostra a distribuição da quantidade de defeitos do conjunto de modelos de Conta encontrados pelas inspeções com *ad-hoc* e com *Shiô*, utilizando um diagrama de *Venn*. Assim, na Figura 5-1 é possível verificar que a inspeção *ad-hoc* encontrou no total 8 defeitos distintos e que a inspeção utilizando *Shiô* encontrou 11 defeitos, sendo 6 defeitos encontrados em comum por ambas as técnicas. A técnica *Shiô* encontrou 5 defeitos que não foram encontrados pela inspeção *ad-hoc*, mas em contrapartida, a inspeção *ad-hoc* encontrou 2 defeitos que não foram detectados pela técnica. Os defeitos encontrados apenas pela inspeção *ad-hoc* foram analisados e procurou-se identificar os motivos dos mesmos não terem sido relatados na inspeção com o uso de *Shiô*. Observou-se que esses defeitos não foram relatados pelos inspetores que utilizaram a técnica, pois as regras de negócio foram desconsideradas pelos inspetores durante a atividade de inspeção. Outras adversidades durante o estudo ocorreram e são descritas com maiores detalhes ainda nessa seção.



Figura 5-1 Quantidade de defeitos encontrados pela inspeção *ad-hoc* e por *Shiô* no conjunto de modelos de Conta.

A Figura 5-2 mostra a distribuição da quantidade de defeitos do conjunto de modelos de Movimento encontrados pelas inspeções *ad-hoc* e por *Shiô*, utilizando o diagrama de *Venn*, onde é possível verificar que a inspeção *ad-hoc* no total, encontrou 14 defeitos distintos e que a inspeção utilizando *Shiô* encontrou 16 defeitos, sendo que existem 12 defeitos comuns. A técnica *Shiô* encontrou 4 defeitos que não foram encontrados pela inspeção *ad-hoc*, mas em contrapartida, a inspeção *ad-hoc* encontrou 2 defeitos que não foram cobertos pela técnica. Assim como no conjunto de modelos de Contas, os defeitos encontrados apenas pela inspeção *ad-hoc* foram analisados e procurou-se identificar os motivos dos mesmos não terem sido relatados na inspeção com



o uso da técnica. Observou-se o mesmo problema, ou seja, a não consideração das regras de negócio durante a inspeção.



Figura 5-2 Quantidade de defeitos encontrados pela inspeção *ad-hoc* e *Shiô* no conjunto de modelos de Movimento.

As Tabela 5-9 e Tabela 5-10 mostram as médias e os desvios padrão dos defeitos e do tempo despendido nas inspeções, em cada uma das rodadas, nos modelos de Conta e de Movimento, respectivamente. Na Tabela 5-9 é possível notar que o comportamento no conjunto de Conta foi o esperado, pois na segunda rodada, houve um melhor rendimento na média de defeitos. Esse comportamento é normal em atividades de inspeção, pois os inspetores adquirem conhecimento tanto do domínio como também da inspeção em si. Em contrapartida, na Tabela 5-10 um comportamento não esperado é verificado, pois a média de defeitos encontrados na segunda rodada foi muito inferior quando comparada à primeira rodada. O Grupo 1 teve uma média de defeitos muito baixa na segunda rodada (média de 2,6) nos modelos de Movimento, enquanto que o Grupo 2 na primeira rodada, teve o rendimento de 6,16 de defeitos encontrados. Uma justificativa seria o fato dos inspetores não terem aplicado a técnica corretamente ou a não aplicação da técnica como um todo, visto que a diferença das médias do tempo despendido entre as rodadas nos modelos de Movimento é muito grande, fato esse que não ocorre entre as rodadas de inspeção com o uso da técnica no outro conjunto de modelos.

Tabela 5-9 Média e desvio padrão de defeitos e tempo no conjunto de Conta.

Rodada	Média Defeito		Desvio Padrão Defeito		Média Tempo		Desvio Padrão Tempo	
	<i>Ad-hoc</i>	<i>Shiô</i>	<i>Ad-hoc</i>	<i>Shiô</i>	<i>Ad-hoc</i>	<i>Shiô</i>	<i>Ad-hoc</i>	<i>Shiô</i>
R1	3,25	3,5	1,258	2,380	36	103,5	13,466	57,285
R2	4	5,4	NA	1,949	61	112,2	NA	66,829

Tabela 5-10 Média e desvio padrão de defeitos e tempo no conjunto de Movimento.

Rodada	Média Defeito		Desvio Padrão Defeito		Média Tempo		Desvio Padrão Tempo	
	<i>Ad-hoc</i>	<i>Shiô</i>	<i>Ad-hoc</i>	<i>Shiô</i>	<i>Ad-hoc</i>	<i>Shiô</i>	<i>Ad-hoc</i>	<i>Shiô</i>
R1	4,333	6,166	3,214	4,999	113	206,666	24,062	113,741
R2	4,6	2,6	0,547	1,341	64	138,2	13,266	68,561

Durante a análise das discrepâncias apontadas pelos participantes da graduação, encontrou-se uma série de adversidades, tais como:

- Grande número de cópias: foi possível notar um grande número de cópias entre os inspetores, apesar de terem sido informados, durante o treinamento, que as inspeções deveriam ocorrer de forma individual. Ficou visível que houve cópias entre os relatórios de discrepâncias. Um exemplo deste fato ocorreu no Grupo 1, na segunda rodada, onde os inspetores G7, G10 e G12 possuem um comportamento muito semelhante, pois o inspetor G7 relatou apenas 1 defeito, que é idêntico ao relatado pelos inspetores G10 e G12, que relataram além deste defeito, um falso positivo. Os inspetores G10 e G12 relataram exatamente as mesmas discrepâncias com descrições muito similares;
- Grande quantidade de falsos positivos: outra característica observada nos relatórios de discrepâncias está na grande quantidade de falsos positivos relatados por muitos inspetores. A causa desse fato pode estar ligada à não compreensão das instruções contidas na técnica, principalmente com a frase: *“Para fazer isto, você precisa pensar sobre a semântica associada a cada ação...”*;
- Regras de negócio desconsideradas: durante a análise das discrepâncias, foi notado que alguns defeitos relacionados às regras de negócio, apontados pela

inspeção *ad-hoc* não foram cobertos pela inspeção com *Shiô*. As regras de negócio são muito importantes, pois como o próprio nome diz, são regras que definem o comportamento do sistema, determinando as regras, restrições e exceções de atividades dentro de um diagrama. Como as regras aparentemente não foram consideradas, defeitos relacionados às transições de estados no diagrama de estados não puderam ser detectados pelos inspetores que utilizaram *Shiô*. Uma possível causa das regras não terem sido consideradas foi a decisão dos pesquisadores de retirar a descrição das regras de negócio dos diagramas de atividades e colocá-las em um documento de texto separado, devido à alta complexidade dos modelos, com relação ao seu tamanho (quantidade de elementos em um diagrama de atividades). Essa decisão foi tomada com o intuito de despoujar os diagramas e assim facilitar sua leitura e inspeção. Desta forma, para cada diagrama de atividades havia um documento texto com a descrição das regras de negócio referentes ao diagrama em questão. Durante o treinamento foi informado que as regras de negócio estariam separadas dos diagramas, mas que deveriam ser levadas em consideração durante as inspeções. Para certificar que essa foi realmente a razão da desconsideração das regras de negócio, um *e-mail* foi enviado aos 20 inspetores que realizaram as duas rodadas de inspeções utilizando a técnica, para averiguar se os documentos de texto com as regras de negócio haviam sido considerados durante a atividade de inspeção. Apenas 11 inspetores responderam, onde: 2 responderam que consideraram as regras, 6 haviam desconsiderado e 3 haviam considerado parcialmente. Esse fato prejudicou o desempenho da técnica quando comparado com a inspeção *ad-hoc*;

- Complementos das ações desconsideradas: durante a análise de discrepâncias, foi possível observar nos diagramas entregues que alguns inspetores não realizaram marcações nos complementos das ações. Esses complementos contêm informações necessárias para o entendimento dos diagramas e a indicação da existência da maioria das regras de negócios está nesses complementos;
- Grupos desbalanceados: após a remoção dos *outliers*, restaram 15 inspetores considerados válidos neste estudo. A Tabela 5-11 mostra por rodada e por conjunto de modelos os seus inspetores válidos, onde os inspetores marcados em

**itálico** e **sublinhado** representam respectivamente os inspetores do Grupo 1 e do Grupo 2 que estão presentes nas duas rodadas como não *outlier*. Na Tabela 5-11 é possível verificar que os participantes não *outliers* do Grupo 1 nas duas rodadas são praticamente distintos, pois existe apenas 1 participante em comum em ambas as rodadas (G23). Em contrapartida os participantes não *outliers* do Grupo 2 são mais constantes, pois o grupo basicamente se manteve (G1, G15, G17 e G24) nas rodadas.

Tabela 5-11 Inspetores não *outliers* por rodada e por conjunto de modelos.

	<b>Conta</b>	<b>Movimento</b>
<b>Rodada 1</b>	(Grupo 1) G19 G22 <b>G23</b> G08	(Grupo 2) <b><u>G1</u></b> G2 G9 <b><u>G15</u></b> <b><u>G17</u></b> <b><u>G24</u></b>
<b>Rodada 2</b>	(Grupo 2) <b><u>G1</u></b> G05 <b><u>G15</u></b> <b><u>G17</u></b> <b><u>G24</u></b>	(Grupo 1) G07 G10 G12 G14 <b>G23</b>

### 5.3 Segundo Estudo

O segundo estudo *in-vitro* executado para avaliar a viabilidade da técnica de inspeção *Shiô* possui muitas similaridades com o primeiro, onde o planejamento e a instrumentação são idênticos ao definido no primeiro estudo. Para o segundo estudo foi definido os seguintes objetivos específicos, apresentados na Tabela 5-12:

Tabela 5-12 Objetivos específicos do segundo estudo.

<b>Analisar a</b>	A inspeção utilizando as técnicas <i>Shiô</i> e <i>ad-hoc</i> .
<b>Com propósito de</b>	Caracterizar
<b>Com respeito ao</b>	Desempenho das técnicas na detecção de defeitos (eficácia e eficiência)
<b>Do ponto de vista do</b>	Pesquisador
<b>No contexto de</b>	Estudantes da disciplina de Engenharia de Software do curso de graduação de Engenharia da Computação de uma universidade particular na cidade do Rio de Janeiro, utilizando modelos reais de um projeto de um sistema de informação Web.

Diferente do primeiro, este segundo estudo foi realizado apenas com uma turma de estudantes do 2º e 3º períodos da graduação de uma universidade particular e os mesmos possuíam conhecimento necessário para participarem do estudo, tais como: diagramas da UML e inspeção. Esses assuntos já haviam sido ministrados anteriormente na própria disciplina em que o estudo ocorreu. Os formulários de consentimento e caracterização, o mesmo utilizado no primeiro estudo, indicaram que todos tinham conhecimento sobre inspeção e diagramas UML.

### 5.3.1 Planejamento e Projeto Experimental

O planejamento utilizado no primeiro estudo foi o mesmo no segundo estudo. Assim, todas as questões, hipóteses, variáveis e métricas foram reutilizadas. Todos os participantes receberam o formulário de consentimento e de caracterização, assim como no primeiro estudo. Os formulários de caracterização foram utilizados para identificar o grau de experiência dos participantes e com base neles, os participantes foram classificados quanto à experiência (baixa, média e alta).

O segundo estudo forneceu inicialmente 16 participantes e todos entregaram os formulários de consentimento e de caracterização respondidos, mas o arranjo utilizado foi diferente do primeiro estudo. Os participantes também foram divididos em dois grupos (Grupo 1 e Grupo 2), baseado nos formulários de caracterização respondidos e a mesma abordagem de divisão dos participantes foi utilizada. Assim, os grupos ficaram com experiências equivalentes, contendo inspetores com diferentes níveis de experiência.

Este estudo também foi agrupado em duas rodadas com os mesmos dois conjuntos de modelos utilizados no primeiro estudo. A inversão dos modelos entre grupos nas rodadas foi igual ao realizado no estudo anterior, porém na primeira rodada, os

participantes dos dois grupos inspecionaram de forma *ad-hoc* e, na segunda, os participantes dos dois grupos inspecionaram utilizando a técnica *Shiô*. O arranjo utilizado neste estudo é apresentado na Tabela 5-13.

Tabela 5-13 Arranjo do segundo estudo.

	Rodada 1 ( <i>Ad-hoc</i> )	Rodada 2 ( <i>Shiô</i> )
<b>Grupo 1</b>	Conta	Movimento
<b>Grupo 2</b>	Movimento	Conta

O objetivo do arranjo mostrado na Tabela 5-13 foi de comparar o desempenho da técnica *Shiô* aplicada na segunda rodada, com o desempenho da inspeção *ad-hoc* aplicada na primeira rodada. Uma comparação entre o desempenho dos inspetores que aplicaram a técnica *Shiô* no segundo estudo, com os inspetores (graduação) que aplicaram a técnica *Shiô* no primeiro estudo também é possível de ser realizada, pois os participantes dos estudos possuíam perfis similares.

Os critérios utilizados para a divisão dos participantes em dois grupos foram os mesmos utilizados no primeiro estudo, ou seja, foram analisadas as questões: experiência com desenvolvimento de software na prática e experiência com inspeção de software na prática.

O resultado desejado para esse estudo é que a técnica *Shiô*, aplicada na segunda rodada, tenha um desempenho superior à *ad-hoc*, aplicada na primeira rodada, pois os inspetores na segunda rodada terão adquirido conhecimento da inspeção *ad-hoc* e do próprio domínio dos modelos, além do auxílio da própria técnica. Outro resultado esperado é que os inspetores da graduação do primeiro estudo tenham um desempenho melhor que os inspetores do segundo estudo (segunda rodada), pois os inspetores do primeiro estudo são um pouco mais experientes por estarem próximos da conclusão do curso, apesar dos perfis serem similar (curso de graduação). Segundo relatos no formulário de caracterização, alguns dos inspetores da graduação do primeiro estudo possuem experiência por serem estagiários ou trabalharem na indústria.

### 5.3.2 Instrumentação e Execução

O conjunto de instrumentos preparados e utilizados no primeiro estudo foi aplicado como apoio para o segundo estudo, visto que o conjunto de diagramas a serem inspecionados, a técnica e o treinamento são idênticos aos utilizados no estudo anterior.

Na primeira rodada, todos os participantes receberam o treinamento sobre inspeção *ad-hoc*, que foi o mesmo treinamento aplicado na turma de pós-graduação da UFRJ, com duração de aproximadamente uma hora. Os inspetores, após o treinamento, receberam por e-mail um pacote idêntico ao enviado à turma de pós-graduação da UFRJ. A Tabela 5-14 apresenta os participantes de cada grupo, o nível de experiência em que cada um foi classificado e os artefatos inspecionados em cada rodada.

Tabela 5-14 Distribuição dos participantes e dos artefatos do segundo estudo.

Grupo	Participantes	Nível de Experiência	Rodada 1 ( <i>Ad-hoc</i> )	Rodada 2 ( <i>Shiô</i> )
Grupo 1	X1	Baixa	Conta	Movimento
	X2	Baixa		
	X5	Alta		
	X7	Baixa		
	X10	Média		
	X11	Baixa		
	X12	Baixa		
	X13	Média		
Grupo 2	X3	Baixa	Movimento	Conta
	X4	Alta		
	X6	Baixa		
	X8	Baixa		
	X9	Baixa		
	X14	Média		
	X15	Média		
	X16	Baixa		

No treinamento os inspetores foram instruídos a inspecionar de forma *ad-hoc* os modelos, ou seja, nenhuma técnica ou heurística foi apresentada para que a inspeção

fosse realizada. Eles utilizaram apenas o conhecimento em inspeções anteriores e do domínio. Assim como no primeiro estudo, os participantes foram instruídos a marcar e indicar o tempo dedicado à atividade de inspeção no relatório de discrepâncias. O prazo de entrega para essa inspeção, não pode ser igual ao utilizado no primeiro estudo, devido à carga-horária da disciplina em que o estudo ocorreu. O prazo foi então estipulado em uma semana para que respeitasse à carga-horária da instituição. A inspeção realizada foi entregue pelo sistema *Moodle*, sistema utilizado pela instituição para entrega de trabalhos e atividades. No dia do término do prazo estipulado, apenas 11 participantes entregaram o resultado da inspeção *ad-hoc* e nesse mesmo dia ocorreu o segundo treinamento sobre a técnica para os participantes.

Na segunda rodada, os participantes receberam o treinamento sobre a técnica, que foi o mesmo treinamento aplicado à turma da graduação da UFRJ. Nesse treinamento os alunos receberam informações e um exemplo didático do funcionamento de *Shiô*. O segundo pacote de inspeção, idêntico ao enviado à turma da graduação da UFRJ, foi enviado por *e-mail* somente aos inspetores que enviaram a primeira inspeção. Os inspetores foram então instruídos a realizarem a inspeção utilizando a técnica proposta, além é claro do conhecimento prévio sobre o domínio. O tempo dedicado à inspeção na segunda rodada também deveria ser marcado e indicado no relatório de discrepâncias, assim como no primeiro estudo.

Todas as instruções do primeiro estudo, sobre as marcações nos modelos e preenchimento do relatório de discrepâncias, também foram apresentadas neste segundo treinamento. O relatório de discrepâncias preenchido e os modelos devidamente marcados após a inspeção deveriam ser entregues também pelo sistema *Moodle*. O prazo estipulado aos inspetores foi igual ao que foi aplicado na primeira rodada deste estudo, uma semana. Poucos inspetores retornaram o resultado da inspeção na segunda rodada, assim alguns dias extras foram oferecidos para que os que não entregaram dentro do prazo pudessem ainda participar do estudo. Após o prazo extra oferecido, apenas 4 participantes entregaram o segundo pacote inspecionado. Assim como no primeiro estudo, a participação neste estudo foi totalmente voluntária.

### **5.3.3 Análise Preliminar dos Resultados**

Como foi dito anteriormente, 11 inspetores entregaram a inspeção *ad-hoc*, são eles: X01, X02, X04, X06, X07, X08, X10, X11, X14, X15 e X16. O conceito de *outlier* foi o mesmo utilizado no primeiro estudo, e dentre os participantes que entregaram a inspeção



na primeira rodada, houve 3 inspetores classificados como *outliers*: X02, X06 e X07. Além do problema de poucos inspetores na primeira rodada, houve ainda a troca de modelos entre os inspetores, ou seja, apesar de ter sido frisado no treinamento que cada um deveria inspecionar o conjunto de modelos que recebesse, os alunos trocaram os modelos entre si. Soma-se ainda o problema de cópias dos relatórios de discrepância entre os participantes. Devido à troca de modelos, os grupos ficaram desbalanceados e assim, 10 participantes entregaram a inspeção do modelo de Movimento e apenas 1 participante entregou a inspeção do modelo de Contas.

Na segunda rodada, como dito anteriormente, apenas 4 participantes entregaram o pacote inspecionado, foram eles: X08, X10, X11 e X15. O participante X11 não entregou os modelos marcados, apenas o relatório de discrepância, portanto foi considerado um *outlier*, pois sem os modelos não foi possível avaliar se as discrepâncias apontadas eram defeitos ou falsos positivos. Os 4 participantes que inspecionaram nesta rodada entregaram o conjunto de modelos Conta e portanto nenhum inspetor entregou o conjunto de modelos de Movimento.

Devido às adversidades encontradas nesse estudo, decidiu-se abandonar quaisquer resultados obtidos por esse estudo, visto que não há como avaliar a técnica onde não houve participação mínima dos participantes.

## 5.4 Análise Qualitativa dos Estudos

Uma análise quantitativa dos dados dos estudos não foi possível de ser realizada, pois nos grupos da pós-graduação restaram apenas poucos participantes, e a turma da graduação, apesar de no início do estudo ter muitos participantes, ao final restaram poucos inspetores. Assim, uma análise quantitativa não seria viável, visto que as amostras com poucos *data points* sempre seguem um comportamento normal, deixando a curva da normal achatada. Portanto uma análise qualitativa foi realizada para se entender os resultados obtidos pelos estudos.

As características específicas de cada inspetor foram observadas na tentativa de entender em que pontos a técnica ajudou e em que pontos ela dificultou, para que mudanças possam ser realizadas com a finalidade de obter uma segunda versão mais aprimorada e robusta. A Tabela 5-15 mostra os inspetores da graduação que relataram defeitos encontrados exclusivamente com o uso da técnica *Shiô*, ou seja, defeitos que a inspeção *ad-hoc* não encontrou.

Tabela 5-15 Inspetores da graduação que encontraram defeitos não encontrados pela inspeção *ad-hoc*

Inspetor	Grupo
G1	Grupo 2
G2	Grupo 2
G5	Grupo 2
G8	Grupo 1
G15	Grupo 2
G17	Grupo 2
G22	Grupo 1
G24	Grupo 2

Os inspetores da Tabela 5-15 possuem algumas características que os distingue dos demais inspetores da graduação, tais características puderam ser obtidas através dos formulários de caracterização respondidos antes das atividades de inspeção. Para realizar essa comparação entre os inspetores foi analisada a moda das características, e após essa análise encontrou-se as seguintes características, que os diferencia dos demais inspetores:

- Experiência em inspeção de casos de uso: o grupo todo da graduação possui como moda “nenhuma experiência”, enquanto que os inspetores da Tabela 5-15 possuem como moda “experiência na prática em um projeto em sala de aula”. A seguir é apresentada a distribuição (inspetores que indicaram o valor da característica/inspetores da graduação que relataram defeitos encontrados exclusivamente com o uso da técnica) dos valores dessa característica:
  - Nenhuma experiência: 2/8
  - Experiência em aula ou livro: 2/8
  - Experiência na prática de um projeto em sala de aula: 3/8
  - Não respondeu: 1/8
- Experiência em Inspeção em diagramas de atividades: o grupo todo da graduação possui como moda “nenhuma experiência”, enquanto que os inspetores da Tabela 5-15 possuem como moda “experiência na prática em um projeto em sala de aula”. A seguir é apresentada a distribuição (inspetores que indicaram o valor da

característica/inspetores da graduação que relataram defeitos encontrados exclusivamente com o uso da técnica) dos valores dessa característica:

- Experiência em aula ou livro: 3/8
  - Experiência na prática de um projeto em sala de aula: 5/8
- Experiência em inspeção em diagramas de estados: o grupo todo da graduação possui como moda “nenhuma experiência”, enquanto que o grupo de inspetores da Tabela 5-15 possuem como moda “experiência na prática em um projeto em sala de aula”. A seguir é apresentada a distribuição (inspetores que indicaram o valor da característica/inspetores da graduação que relataram defeitos encontrados exclusivamente com o uso da técnica) dos valores dessa característica:
    - Experiência em aula ou livro: 2/8
    - Experiência na prática de um projeto em sala de aula: 6/8
- Experiência em inspeção em outros diagramas da UML: o grupo todo da graduação possui como moda “nenhuma experiência”, enquanto que o grupo de inspetores da Tabela 5-15 possuem como moda “experiência na prática em um projeto em sala de aula”. A seguir é apresentada a distribuição (inspetores que indicaram o valor da característica/inspetores da graduação que relataram defeitos encontrados exclusivamente com o uso da técnica) dos valores dessa característica:
    - Nenhuma experiência: 2/8
    - Experiência em aula ou livro: 2/8
    - Experiência na prática de um projeto em sala de aula: 3/8
    - Não respondeu: 1/8
- Experiência em gerência de sistemas: o grupo de inspetores da Tabela 5-15 são mais inexperientes quando comparados com o grupo todo da graduação, visto que os primeiros possuem como moda “nenhuma experiência”, enquanto que o grupo completo possui como moda “experiência na prática em um projeto em sala de aula”;

- Experiência em desenvolvimento de sistemas: a diferença da experiência entre o grupo de inspetores da Tabela 5-15 com o grupo todo da graduação não é muito grande, pois a moda da experiência dos primeiros é de “prática em um projeto na indústria”, enquanto que no grupo completo a moda é de “prática em um projeto em sala de aula”. A seguir, é apresentada a distribuição (inspetores que indicaram o valor da característica/inspetores da graduação que relataram defeitos encontrados exclusivamente com o uso da técnica) dos valores dessa característica:
  - Experiência em aula ou livro: 1/8
  - Experiência na prática de um projeto em sala de aula: 3/8
  - Experiência na prática de um projeto na indústria: 3/8
  - Experiência na prática de vários projetos na indústria: 1/8

A classificação do nível de experiência dos inspetores, baseada nos formulários de caracterização, pode ter sido equivocada, pois nos dois conjuntos de modelos, o inspetor G17 foi o que apresentou melhor rendimento nas métricas de eficácia (defeitos/discrepância e defeitos/total de defeitos). Esse inspetor foi classificado como baixa experiência e também faz parte do grupo de inspetores que encontrou defeitos que não foram detectados pela inspeção *ad-hoc*. Essa classificação ficou contraditória com a análise feita sobre as características desses inspetores, pois a análise mostra que esses inspetores são mais experientes em inspeções. O equívoco também pode estar localizado na percepção dos inspetores sobre as suas experiências, visto que as respostas estavam em escala ordinal.

Um questionário de avaliação, apresentado no Apêndice D, foi distribuído para os inspetores da graduação da UFRJ para avaliar alguns pontos do estudo: sobre a técnica, qualidade do treinamento e modelos inspecionados. Sugestões de melhorias, pontos fracos e fortes foram sugeridos aos inspetores para que respondessem, porém a minoria o fez. Dentre os 20 participantes que aplicaram a técnica, considerando os *outliers*, apenas 7 responderam o questionário de avaliação. Nas respostas enviadas foi encontrada uma cópia entre os participantes, onde o próprio participante afirma ter copiado o questionário de avaliação do outro participante. Portanto, dos dois questionários respondidos de forma idêntica, apenas um foi considerado, totalizando no primeiro estudo um total de 6 questionários válidos.

O mesmo questionário de avaliação foi enviado aos participantes do segundo estudo e apesar dos resultados terem sido descartados, o foco dos questionários era entender os motivos que atrapalharam a aplicação da técnica. Portanto, o questionário foi enviado aos 4 participantes, mas apenas 1 inspetor respondeu. Somando-se aos questionários respondidos pelos participantes do primeiro estudo, obteve-se um total de 7 questionários respondidos, dentro de uma amostragem de 24 participantes dos estudos.

Segundo GARDNER e ALTMAN (1989), a Figura 5-3 calcula o grau de confiabilidade estatística, dado que o  $\epsilon_0$  é o nível de confiabilidade,  $n$  é o tamanho da amostra e  $N$  é o tamanho da população. Assim substituindo os valores de  $n$  por 7 e de  $N$  por 24, obtém-se que o grau de confiabilidade dos questionários de avaliação é de 31,81%, um valor muito baixo e por consequência os dados obtidos não fornecem um bom nível de confiança.

$$n = \frac{N - \frac{1}{\epsilon_0}}{N + \frac{1}{\epsilon_0}}$$

Figura 5-3 Grau de confiabilidade estatística (Gardner e Altman, 1989).

Apesar do baixo nível de confiança obtido pelas poucas respostas do questionário de avaliação da técnica, tentou-se extrair informações que justificassem o desempenho não satisfatório da técnica, quando comparada com a inspeção *ad-hoc*. Assim, com base nas respostas obtidas de cada uma das questões do questionário, as respostas foram tabuladas e gráficos foram gerados para obter uma melhor visualização das respostas. As Figuras 5-4 a **Erro! Fonte de referência não encontrada.** apresentam os gráficos da distribuição das respostas de cada questão do questionário de avaliação.

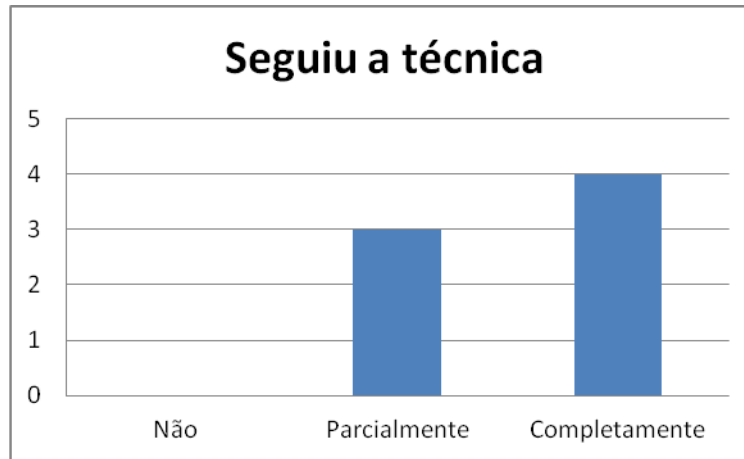


Figura 5-4 Respostas do questionário de avaliação da técnica referente à adesão da técnica.

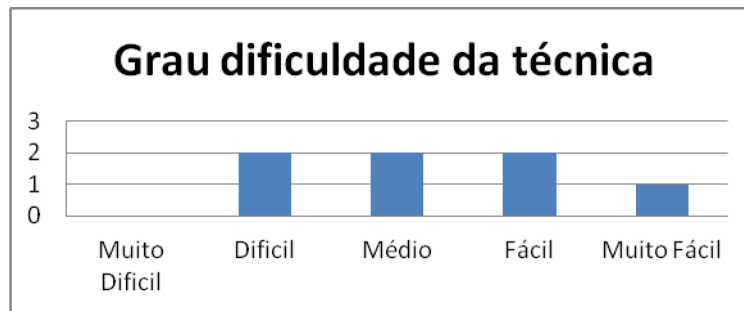


Figura 5-5 Respostas do questionário de avaliação da técnica referente ao grau de dificuldade da técnica.

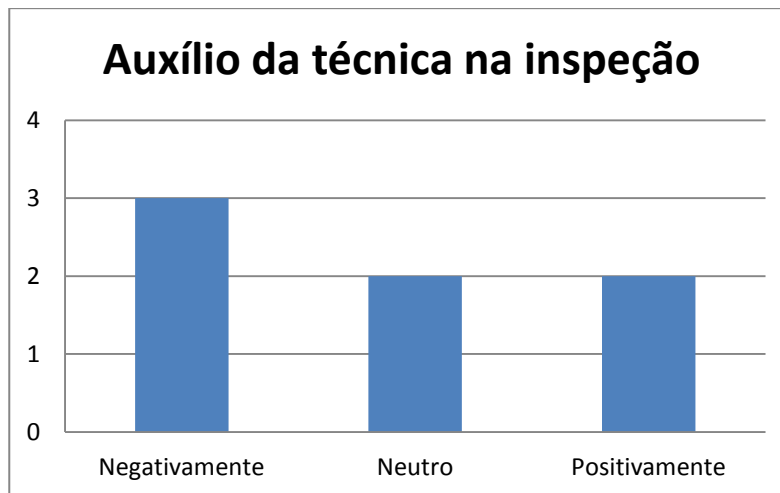


Figura 5-6 Respostas do questionário de avaliação da técnica referente ao auxílio da técnica durante a inspeção.

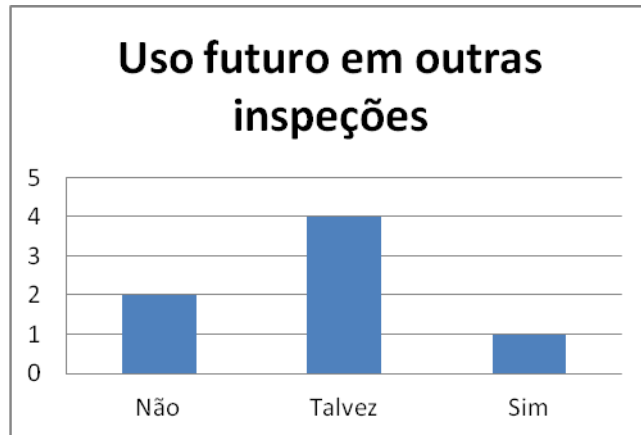


Figura 5-7 Respostas do questionário de avaliação da técnica referente ao uso em oportunidades futuras de inspeção.

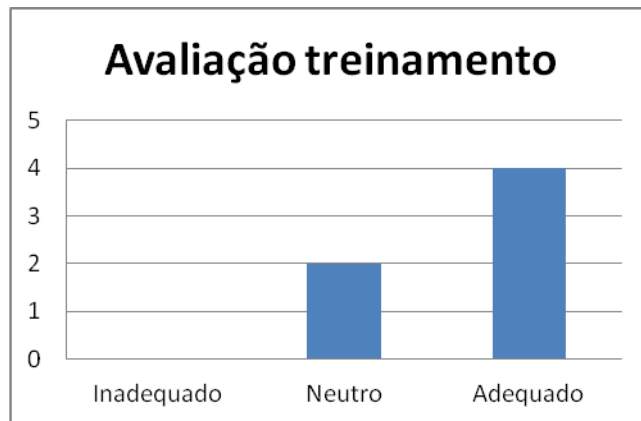


Figura 5-8 Respostas do questionário de avaliação da técnica referente à qualidade do treinamento aplicado.

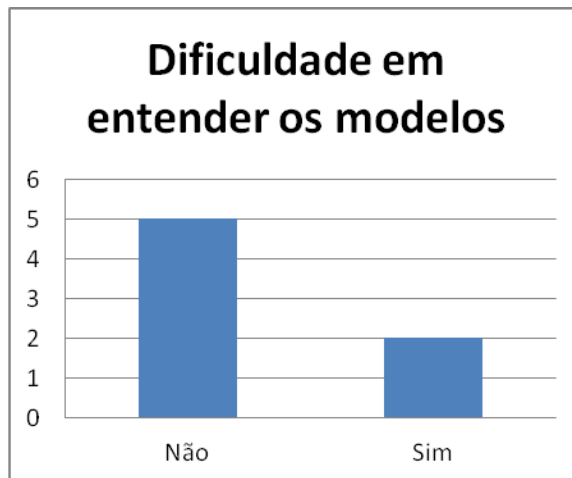


Figura 5-9 Respostas do questionário de avaliação da técnica referente à dificuldade em entender os modelos inspecionados.

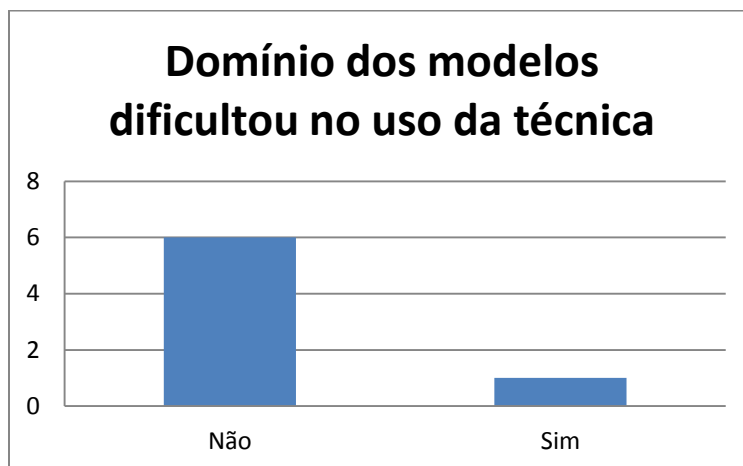


Figura 5-10 Respostas do questionário de avaliação da técnica referente à dificuldade em entender o domínio dos modelos inspecionados.



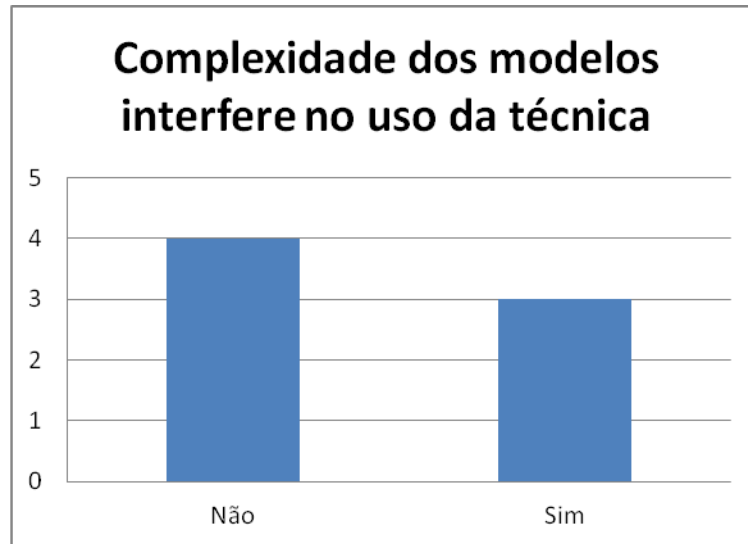


Figura 5-11 Respostas do questionário de avaliação da técnica referente à complexidade dos modelos inspecionados.

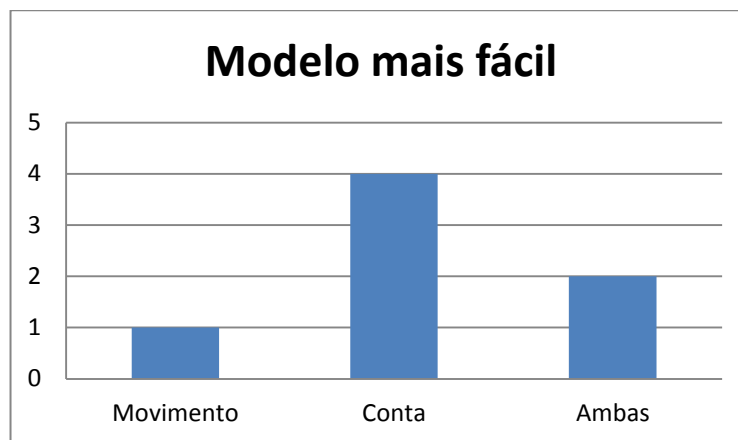


Figura 5-12 Respostas do questionário de avaliação da técnica referente ao modelo mais fácil de inspecionar.

De acordo com as Figura 5-4 a Figura 5-12, é possível observar que apesar da maioria ter seguido as instruções da técnica, os mesmos tiveram alguma dificuldade em aplicá-la. Este fato pode ter impactado ao considerarem que a técnica não auxilia muito na atividade de inspeção e por esta razão não usariam ou ficariam em dúvida em utilizá-la em uma futura oportunidade de inspeção. Apesar disso, os inspetores consideraram que o treinamento oferecido foi adequado e que não tiveram grandes dificuldades em entender os modelos. O domínio financeiro, que poderia ser um fator de confusão no uso da técnica, aparentemente não foi, segundo as respostas dos inspetores. Outro fato

curioso foi a indicação dos inspetores que a complexidade dos modelos, devido ao tamanho (quantidade de elementos), não interferiu na aplicação da técnica, mas consideraram o conjunto que possui menos diagramas (Conta) como o mais fácil de aplicar a técnica.

## 5.5 Lições Aprendidas com os Estudos

Os resultados dos dois estudos propiciaram muitas lições aprendidas, principalmente nas adversidades encontradas durante sua execução. No primeiro estudo, apesar da turma da graduação fornecer inicialmente um considerável número de participantes, a participação no estudo foi de fato baixa. Alguns dos motivos observados são:

- Falta de maturidade dos participantes: apesar da participação ser voluntária nos estudos, uma parte dos participantes era alunos da graduação, ou seja, em média eles são mais novos que os estudantes da pós-graduação e provavelmente não entenderam a seriedade e a necessidade do estudo;
- Prazo estipulado para a inspeção: o prazo para o primeiro estudo foi estipulado em dois dias. Alguns participantes reclamaram sobre o prazo estabelecido, por estarem envolvidos em outras disciplinas e atividades do curso.

Com os resultados obtidos pelos participantes que de fato participaram do primeiro estudo, foi possível perceber que algumas instruções da técnica deveriam ser mais claras, de forma que chame a atenção do inspetor para alguns pontos não percebidos pelos participantes, principalmente sobre as regras de negócio. Outro resultado observado foi a necessidade da reestruturação das instruções da técnica, pois por considerarem confusas, pode ter influenciado na baixa adesão ao estudo.

No segundo estudo pode ser observado que a falta de maturidade dos participantes foi o fator que impossibilitou sua continuidade. Como dito anteriormente, o segundo estudo foi executado em uma disciplina que é ministrada nos primeiros períodos do curso. Assim, os participantes eram mais jovens que os participantes da graduação do primeiro estudo e provavelmente não entenderam a sua importância.

Com base nestas experiências, a técnica *Shiô* foi evoluída para a sua segunda versão, que é apresentada na íntegra na seção 5.7. A sua estrutura permaneceu similar à OORTs, com uma sequência de passos, com dados de entradas e saídas bem definidos, porém foram feitas as seguintes modificações:

- Os passos II, III e IV foram rearranjados, de forma que os passos II.B e IV estão agora subdivididos com o intuito de deixar a leitura mais objetiva e clara;
- As regras de negócio (desconsideradas pela maioria dos inspetores) foram reescritas, alertando os inspetores para serem analisadas;
- Os complementos das ações (notas que ficam anexadas às ações) nos diagramas foram desconsiderados por alguns inspetores e, portanto, foram reescritas a fim de que sejam consideradas durante a aplicação da técnica;
- Defeitos relacionados com alguns pontos dos diagramas, tais como estados iniciais e finais foram frisados, pois apesar de serem relatados pela técnica *Shiô*, poucos inspetores relataram esse tipo de defeito, que é um defeito sintático que pode ser facilmente detectado.

Para os próximos estudos experimentais, a escolha do perfil dos participantes é de fundamental importância para a execução dos estudos. Pela experiência nos dois estudos executados, seria interessante não repetir os estudos utilizando estudantes de graduação, mesmo que sejam dos últimos períodos. Os estudantes da pós-graduação demonstraram mais interesse e, portanto, realizaram o estudo com maior seriedade. O prazo de 2 dias também não foi o mais adequado, pois contratempos com maiores prioridades podem surgir nesse curto prazo, impedindo os inspetores de participarem dos estudos.

## 5.6 Ameaças à validade

As principais ameaças relacionadas à validade dos dois estudos realizados são listadas a seguir:

- **Aderência à técnica:** os inspetores não foram observados enquanto estavam inspecionando os artefatos. Dessa forma, não temos a garantia que os mesmos seguiram a técnica da maneira adequada;
- **Eventos externos:** o estudo foi baseado e planejado com base na agenda das pessoas envolvidas. Entretanto, qualquer evento provocado por fatores externos, tais como, doenças, situações de emergência, entre outros, são considerados ameaças à validade do estudo;
- **Medição do tempo nas inspeções:** foi solicitado aos participantes que marcassem a hora do início e do término da inspeção, calculando assim, a

soma em minutos despendidos nas inspeções. No treinamento ainda foi informado que evitassem o máximo de interrupções, para que o tempo marcado fosse o mais preciso com relação às atividades propostas. Entretanto, não há garantia quanto à exatidão do tempo reportado pelos participantes;

- **Problemas em identificar cores:** a técnica faz uso de marcações e colorações nos modelos, porém há pessoas com dificuldades na percepção de cores. Este fato não estava previsto e, portanto, está fora do escopo deste trabalho;
- **Ausência de uma ferramenta de apoio:** a técnica possui dois passos que consistem basicamente na coloração e marcação dos diagramas. Estes passos requerem certo tempo na aplicação da técnica, assim devido à esses passos serem extremamente mecânicos, esse fato pode ter prejudicado no desempenho dos estudos.
- **Generalização dos estudos:** os resultados obtidos não podem ser generalizados a outros contextos, pois a amostra da população é muito pequena e o domínio de aplicação restrito. Os perfis dos inspetores que participaram são bem específicos, assim como o número de modelos é pequeno.

## 5.7 Segunda Versão da Técnica

A técnica *Shiô* foi evoluída através dos estudos experimentais realizados e das lições aprendidas. A nova versão da técnica é apresentada a seguir:

### **Técnica de Leitura para Casos de Uso descritos por Diagramas de Atividades (DA) e Diagramas de Estado (DE)**

**Objetivo:** Verificar se os DEs que modelam o sistema estão em conformidade com os DAs que descrevem os casos de uso e atividades do sistema.

#### **Entradas para o processo:**

1. DAs que descrevem as atividades realizadas pelo sistema.
2. DEs que descrevem os estados internos de objetos do sistema.

Execute os seguintes passos:

I. Para **cada DE**, faça:

**ENTRADAS:** DEs e DAs

**SAÍDAS:** DEs marcados com canetas coloridas; conjunto de DAs e relatório de discrepância.

- A. Determine qual classe/objeto está sendo modelada pelo DE.
- Se você não pode identificar a classe/objeto que está sendo modelada, então alguma informação está sendo *omitida* ou está *ambígua*. Indique isto no relatório de discrepância.
- B. Acompanhe a sequência de estados e transições de estados do DE. Comece pelo estado inicial (●) e siga as transições até que você encontre algum estado final (●), se houver. Esteja certo que você passou por todas as transições (diferentes caminhos). Inclua em sua análise, se existir, todas as submáquinas e os estados compostos, considerando, portanto o DE contido em cada submáquina e os estados internos presentes nos estados compostos.
- Caso não exista a indicação de estado inicial ou final, relate como uma *omissão* no relatório de discrepância.
  - Marque utilizando uma caneta vermelha os estados
  - Marque utilizando uma caneta azul as transições entre estados
  - Esteja certo que você pode entender e descrever o que está acontecendo com a classe/objeto apenas lendo o DE. Se você não pode, então o DE está *ambíguo* ou *incompleto*. Indique isto no relatório de discrepância.
  - Caso não exista a indicação de estado final, relate como uma omissão no relatório de discrepância.
- C. Para cada DE encontre os DAs que utilizam a classe/objeto modelada pelo DE; buscando pelo próprio nome do DE dentro de todas as estruturas (ações, atividades, condições de guarda, regras de negócio,...) dos DAs. Fique atento para representações com nomes diferentes, mas que possuem o mesmo significado (Ex: no DA existe o conceito de carro, porém no DE a classe/objeto representada é automóvel. Carro e automóvel representam um mesmo conceito, porém com nomes distintos). Utilize algum tipo de marcação (números ou símbolos) para marcar a correspondência entre o DE e os DAs que utilizam a classe/objeto modelado pelo DE. Este subconjunto de DAs será utilizado para o restante da inspeção. Lembre-se: um DA pode pertencer a subconjuntos de diferentes DEs.
- Caso encontre um mesmo conceito utilizando nomes diferentes em partes diferentes do diagrama ou até mesmo em diagramas diferentes, relate no relatório de discrepância como *ambiguidade*.

II. Para **cada DA** do **subconjunto previamente identificado** no passo anterior, faça:

**ENTRADA:** Subconjuntos de DAs

**SAÍDAS:** DAs e Documento texto com a descrição das regras de negócio marcados com canetas coloridas; relatório de discrepância.

- A. Selecione os documentos texto com as regras de negócio referentes ao subconjunto de DAs previamente separados no passo anterior (I.C). Essas regras de negócio serão utilizadas nos próximos passos da inspeção.
- B. Acompanhe a sequência de ações e fluxos de controle do DA. Comece pelo estado inicial (●) e siga os fluxos de controle até que você encontre algum estado

final (●). Esteja certo que você passou por todas as ações e fluxos de controle (diferentes caminhos).

- Para cada subdiagrama (indicado pelo símbolo  $\text{rh}$ ), avalie a granularidade da abstração, analisando a necessidade de utilização do subdiagrama para entendimento da atividade. Caso sejam necessários maiores detalhes sobre a atividade, utilize o DA indicado pelo subdiagrama para a inspeção. Caso contrário utilize apenas o subdiagrama, representando a ação.
- Marque utilizando uma caneta vermelha:
  - i. Ações com os estereótipos *System Response* e *System Action*. Leve em consideração as informações presentes nas notas de complemento anexadas às ações no DA. Atente-se para as diferentes opções de combinações de status/situações que as classes/objeto podem assumir. Vale lembrar que um Diagrama de Atividade pode utilizar vários DEs. Dê a cada ação um *label* único [A1, A2, ...].
- Marque utilizando uma caneta azul:
  - i. Ações com o estereótipo *ActorAction*. Dê a cada ação um *label* único [B1, B2, ...].
  - ii. Fluxos entre as ações que contenham pontos de decisão, *loops*, restrições e condições de guarda (Ex: restrição para  $t > 0$ , isto é, a transição para outra ação/atividade só deve ocorrer caso o valor de  $t$  seja maior que zero). Dê a cada fluxo um *label* único [C1, C2, ...].
  - iii. **Apenas** as regras de negócio (*Business Rules*) que indiquem restrições e condições de funcionamento do sistema (Ex: restrição de tempo, restrição da condição em que o sistema ou parte dele se encontra). As regras de negócio podem limitar as combinações de status/situação identificadas e marcadas nas ações marcadas em vermelho. Para realizar tal tarefa, você deve utilizar seu conhecimento do domínio para identificar quais são as regras de negócio que definem condições para que a situação especificada na regra ocorra. Marque também essas descrições das regras de negócio (documento texto separado no passo II.A). Dê a cada regra de negócio um *label* único [D1, D2, ...]. Indique o mesmo *label* na marcação nos documentos texto, assim a descrição da regra de negócio no documento texto e a regra de negócio no diagrama terão *labels* idênticos.
- Marque utilizando uma caneta verde:
  - i. **Apenas** pré e pós-condições da atividade que indicam o estado em que o sistema deve estar para executar o caso de uso (pré-condições) e o estado em que o sistema deve estar quando o caso de uso encerrar (pós-condição). Dê a cada pré e pós-condição um *label* único [E1, E2, ...].

III. Para **cada DE, analise todos os DAs correspondentes** e marcados anteriormente, dando atenção para representações com nomes diferentes, mas que possuem o mesmo significado (vale lembrar que estas discrepâncias devem ter sido previamente encontradas e relatadas no relatório de discrepância).

**ENTRADAS:** DEs, DAs e Documento texto com a descrição das regras de negócio marcados com canetas coloridas.

**SAÍDAS:** DEs, DAs e Documento texto com a descrição das regras de negócio com marcações que os relacionam.

- A. Para cada ação ou conjunto de ações (*SystemAction* ou *SystemResponse*) que façam sentido juntas (marcada em vermelho nos DAs), encontre um estado correspondente no DE (marcado em vermelho). Para fazer isto, você precisa pensar sobre o significado associado a cada ação ou conjunto de ações. Elas têm alguma relação com os possíveis estados que este objeto poderia assumir? Os diferentes status/situações identificados no passo anterior (II.B) provavelmente representam estados no DE. Analise esta possibilidade e em caso positivo, identifique no DE estes estados. Nomeie o estado do DE com os mesmos *labels* das ações dos DAs, assim o estado do DE terá os mesmos *labels* que as ações dos DAs. Faça um sinal (✓) nas marcas em vermelho em ambos os diagramas, indicando que existe correspondência entre eles.
- B. Para cada ação ou conjunto de ações (*ActorAction*) que façam sentido juntas (marcada em azul nos DAs), encontre uma transição correspondente no DE (marcada em azul). Para fazer isto, você precisa pensar sobre o significado associado a cada ação. Essas ações têm alguma relação com as possíveis transições que este objeto poderia sofrer? Nomeie a transição do DE com os mesmos *labels* das ações dos DAs, assim a transição do DE terá os mesmos *labels* que as ações dos DAs. Faça um sinal (✓) nas marcas em azul em ambos os diagramas, indicando que existe correspondência entre eles.
- C. Para cada alternativa de fluxo de controle (ponto de decisão, *loop*, restrição ou condição de guarda), marcado em azul nos DAs encontre uma transição correspondente no DE (marcada em azul). Nomeie a transição do DE com o mesmo *label* do fluxo de controle dos DAs, assim a transição do DE terá os mesmos *labels* que as ações dos DAs. Faça um sinal (✓) nas marcas em azul em ambos os diagramas, indicando que existe correspondência entre eles.
- D. Para cada regra de negócio encontrada nos DAs (marcada em azul), leia sua descrição no documento texto referente ao diagrama analisado e encontre uma transição correspondente no DE (marcada em azul). Nomeie a transição do DE com o mesmo *label* da regra de negócio dos DAs, assim a transição do DE terá o mesmo *label* que a regra de negócio dos DAs. Faça um sinal (✓) nas marcas em azul em ambos os diagramas e no documento texto, indicando que existe correspondência entre eles.

IV. Para **cada DE** com os seus **DAs correspondentes**, devidamente marcados, analise:

**ENTRADAS:** DEs, DAs e Documento texto com a descrição das regras de negócio marcados com canetas coloridas e com sinal (✓)

**SAÍDA:** relatório de discrepância

- A. Considerando as ações com estereótipos *SystemAction* e *SystemResponse* (marcadas em vermelho). Relate no relatório de discrepância como:
  - *Omissão:* apenas marcação em vermelho nos DAs sem o sinal (✓) que **represente** estados no DE.
  - *Informação estranha:* apenas marcação em vermelho no DE sem o sinal (✓).
  - *Fato incorreto:* utilizando seu conhecimento sobre o domínio, verifique se existem ações em vermelho nos DAs sem o sinal (✓) e que determinam um estado ou a alteração de estado da classe/objeto, porém a

correspondência no DE indica um estado ou alteração de estados diferente da esperada.

- B. Considerando as ações com estereótipos com estereótipo *ActorAction* (marcadas em azul). Relate no relatório de discrepância como:
- *Omissão*: apenas marcação em azul nos DAs sem o sinal (✓) que **represente** transição de estados.
  - *Fato Incorreto*: apenas marcação em azul sem o sinal (✓) que **determine** uma transição de estado da classe/objeto, porém no DE correspondente está mapeado para uma transição com valor diferente do que está representado pelo DA.
- C. Considerando os pontos de decisão, *loops*, restrições ou condições de guarda (marcados em azul). Relate no relatório de discrepância como:
- *Omissão*: apenas marcação em azul nos DAs sem o sinal (✓) que **represente** transição de estado no DE.
  - *Fato Incorreto*: apenas marcação em azul nos DAs sem o sinal (✓) que possua informações de condição de guarda e restrições incompatíveis com as informações apresentadas nas transições de estados (ex: valores diferentes para restrições de tempo).
- D. Considerando as regras de negócio (marcadas em azul). Relate no relatório de discrepância como:
- *Omissão*: apenas marcação em azul nos DAs sem o sinal (✓).
  - *Fato Incorreto*: apenas marcação em azul nos DAs sem o sinal (✓) que possua informações de regras de negócio incompatíveis com as informações apresentadas nas transições de estados (ex: valores diferentes de saldo para que uma transação bancária ocorra).
  - *Informação Estranha*: apenas marcação em azul no DE sem o sinal (✓). As transições no DE pode corresponder à: ações *ActorAction*, pontos de decisão, *loops*, restrições, condições de guardas ou regras de negócio no DAs, e como não foi sinalizada (✓) em nenhum desses construtos, pode existir alguma informação estranha no DE.
- E. Considerando as pré e pós-condições (marcadas em verde), encontre a ordem cronológica dos estados no ciclo de vida da classe/objeto que esteja presente nas cláusulas de pré e pós-condições. Ou seja, as marcações em verde nos DAs devem indicar qual estado a classe/objeto deve estar para iniciar ou terminar a execução da atividade. Assim no DE o estado indicado na pré-condição corresponde a algum estado anterior ao estado apresentado como pós-condição.
- Caso a ordem dos estados esteja trocada ou diferente das apresentadas nas pré e pós-condições, relate no relatório de discrepância como *fato incorreto*.
- V. Reveja os diagramas utilizados nesta inspeção
- ENTRADAS:** DEs, DAs e Documento texto com a descrição das regras de negócio marcados com canetas coloridas e com sinal (✓)
- SAÍDA:** relatório de discrepância
- A. Utilizando o seu conhecimento sobre o domínio e sua experiência em modelagem, existe alguma classe/objeto, estado ou transição no DE que esteja faltando e que seja importante para o sistema.
- Caso positivo relate no relatório de discrepância como *omissão*.



- B. Você consegue identificar algum nome, conceito ou qualquer outro componente no diagrama que esteja diferente do seu conhecimento de domínio?
  - Caso positivo relate no relatório de discrepância como *fato incorreto*.
- C. Pensando no que você entende do domínio, as sequências de transições e os estados, podem efetivamente representar o ciclo de vida da classe/objeto participante do sistema?
  - Caso negativo relate no relatório de discrepância como *fato incorreto*.

## 5.8 Conclusão

A técnica *Shiô* encontra defeitos, mas a sua aplicação pode consumir mais tempo se comparado com a inspeção *ad-hoc*. Entretanto, a técnica encontra defeitos que a inspeção *ad-hoc* não foi capaz de detectar.

No primeiro estudo, os inspetores da graduação que encontraram defeitos exclusivamente devido à aplicação da técnica eram mais experientes dos demais inspetores da graduação do estudo. Este fato pode ter sido um fator que influenciou no desempenho das inspeções, porém existe a hipótese de que a técnica *Shiô* aliada à maior experiência ajude os inspetores a analisarem pontos que poderiam passar despercebidos, caso realizassem uma inspeção *ad-hoc*. Assim, ficou constatado que inspetores mais experientes, juntamente com o uso da técnica, resultaram em uma maior eficiência ao detectar defeitos.

Nos estudos pode ser observado que existiu a possibilidade do agrupamento dos participantes quanto à experiência ter ocorrido de maneira equivocada e influenciada pela percepção de experiência do próprio inspetor ao responder o formulário de caracterização. Somado a essa questão, no decorrer dos estudos foram observadas diversas algumas adversidades nos participantes, tais como: cópias das inspeções, trocas de modelos entre participantes, cópias dos formulários de avaliação da técnica entre outros. Apesar destas anomalias, os dados ainda foram tratados visando a aproveitar o máximo possível as possibilidades de observação.

Outra constatação foi que o arranjo e estrutura da técnica acarretaram dificuldades durante a realização da inspeção, o que resultou em um grande tempo despendido durante a inspeção. A partir dos resultados surgiu uma oportunidade de melhoria, principalmente na reestruturação da técnica, que foi evoluída para uma segunda versão.

## 6 Conclusão e Trabalhos Futuros

*Neste capítulo são apresentadas as considerações finais desta dissertação, destacando as contribuições da pesquisa, as limitações encontradas e as perspectivas de trabalhos futuros.*

### 6.1 Considerações Finais

Esta dissertação apresentou o desenvolvimento de uma técnica de leitura para inspeção de diagramas de estados com base em diagramas de atividades que especificam requisitos (casos de uso). A proposta desta nova tecnologia foi fundamentada nos conceitos recomendados pelo processo de desenvolvimento de tecnologias baseado em evidência, onde foram realizados um estudo secundário (*quasi-revisão sistemática*) e dois estudos primários (estudos de viabilidade).

Os resultados da *quasi-revisão sistemática* indicaram a carência de técnicas de inspeção em diagramas de fluxos de atividades, principalmente de técnicas que realizem comparação entre diagramas. Assim, a técnica de inspeção *Shiô* foi elaborada visando comparar diagramas de estados com diagramas de atividades que descrevem casos de uso. A técnica foi inspirada e construída para se integrar a OORTs e, portanto, contém uma sequência de passos com entradas e saídas bem definidas, seguido de uma série de instruções que guiam o inspetor para detectar as discrepâncias.

Os resultados dos dois estudos primários conduzidos para avaliar a viabilidade de *Shiô* indicaram que a técnica encontra defeitos, incluindo defeitos não detectados comumente por inspeções com técnica *ad-hoc*, porém a sua aplicação pode requerer um tempo maior quando comparado com inspeções com *ad-hoc*. Outro resultado obtido pelos estudos foi a identificação de defeitos encontrados exclusivamente por inspeções *ad-hoc* e durante a análise dos dados foram constatados diversos problemas nas inspeções que utilizaram *Shiô* em sua primeira versão. Por isso, a técnica *Shiô* foi evoluída para a segunda versão, visando chamar atenção dos inspetores para as partes da técnica que ficaram omissas de acordo com os resultados dos estudos. Na segunda versão ocorreu também o rearranjo de alguns trechos da técnica, com o objetivo de tornar a leitura das instruções mais clara e objetiva para os inspetores.

## 6.2 Contribuições da Pesquisa

As principais contribuições desta dissertação são:

- A técnica *Shiô*, que consiste em uma técnica de leitura para inspeção em diagramas de estados utilizando como referência os diagramas de atividades especificando a descrição de casos de uso.
- Um protocolo de pesquisa para uma *quasi*-revisão sistemática relacionado a identificação de técnicas de inspeção em diagramas de fluxos de atividades, cujos resultados demonstram que atualmente existe uma carência desse tipo de tecnologia;
- Os planos de dois estudos de viabilidade *in-vitro* que indicaram que a técnica *Shiô* encontra defeitos em numero e diversidade maior do que encontrados por inspeções com *ad-hoc*, porém a sua aplicação pode despendar mais tempo;

## 6.3 Limitações

Os resultados da *quasi*-revisão sistemática apresentaram poucos trabalhos relacionados e os mesmos não puderam ser utilizados para a elaboração da técnica, visto que nenhum dos trabalhos retornados apresentava a comparação de diagramas de atividades com outro diagrama.

A técnica foi elaborada utilizando diagramas de atividades gerados pela ferramenta *UseCaseAgent*, que não oferece suporte a todos os recursos sintáticos disponíveis na UML 2.4.1. Assim, a técnica *Shiô* só pode ser aplicada em projetos que utilizem esta ferramenta, pois além dos recursos sintáticos estabelecidos, a técnica de inspeção explora os estereótipos definidos na ferramenta.

Os estudos de viabilidade conduzidos para avaliar a aplicabilidade da técnica *Shiô* utilizaram modelos de apenas um projeto e com domínio específico. O projeto utilizado é de larga escala, dividido em diversos módulos, porém foram utilizados alguns diagramas de apenas um módulo deste projeto. Além do fato, de poucos participantes estarem envolvidos nos estudos.

A segunda versão da técnica, evoluída com base nos resultados dos estudos primários ainda não passou por avaliação experimental. A expectativa é que a nova organização e simplificação da técnica permita obter ganhos em relação à primeira versão. Entretanto, por questões de tempo, não foi possível realizar esta avaliação.

## 6.4 Trabalhos Futuros

Conforme dito anteriormente, a segunda versão de *Shiô* ainda não pode ser avaliada por nenhum estudo experimental. Assim, avaliações desta segunda versão da técnica constituem um dos trabalhos futuros. Entendemos ser importante a repetição dos estudos e também a realização de novos estudos *in-vitro* e *in-vivo* com outros tipos de projetos e considerando diferentes domínios. A escolha dos participantes dos estudos deve ser criteriosa, para evitar que os mesmos problemas relatados nos estudos originais ocorram novamente.

Estudos qualitativos também seriam interessantes, com o intuito de observar o comportamento dos inspetores ao longo da aplicação da técnica *Shiô*. O resultado deste tipo de estudo seria uma fonte rica para observar a usabilidade da técnica e permitir sua evolução, caso necessário.

Outra oportunidade de pesquisa se apresenta na extensão da técnica de leitura para contemplar os demais recursos sintáticos previstos nos diagramas de atividades da UML 2.4.1, visto que os recursos sintáticos atualmente cobertos pela técnica estão limitados devido ao uso da ferramenta *UseCaseAgent* e do *checklist Actcheck*.

Destaca-se também a oportunidade de pensar em uma ferramenta que automatize parte da técnica, principalmente os passos I e II, pois se referem basicamente à marcação (colorir) dos recursos sintáticos dos diagramas com cores pré-definidas. Estes passos requerem certo tempo da aplicação da técnica, mas apesar da automação a leitura dos diagramas é fundamental para a inspeção. Essa leitura e compreensão dos diagramas fornecem as informações necessárias para a aplicação de outros passos da técnica. Os demais passos requerem o raciocínio do inspetor para encontrar a relação entre o diagrama de estados e os diagramas de atividades e detectar as discrepâncias. A ferramenta ORION (REIS, 2005) oferece apoio para a coloração dos diagramas utilizados pela família de técnicas de leitura de OORTs, assim a nova ferramenta a ser criada pode complementar ou evoluir ORION.

## REFERÊNCIAS BIBLIOGRÁFICAS

- BASIL, V. R., SELBY, R. W. (1987). "Comparing the Effectiveness of Software Testing Strategies", IEEE Transactions on Software Engineering 13 (12): 1278-1296.
- BERNARDEZ, B., GENERO, M., DURAN, A., TORO, M. (2004). "A controlled experiment for evaluating a metric based reading technique for requirements inspection", Software Metrics. Proceedings of 10th International Symposium on 14-16, Sept. pp. 257-268.
- BEZERRA, E. (2006). "Princípios de Análise e Projeto de Sistemas com UML", 2ed, Editora Campus.
- BIOLCHINI, J., MIAN, P. G., NATALI, A. C., TRAVASSOS, G. H. (2005). "Systematic Review in Software Engineering: Relevance and Utility", Relatório Técnico PESC - COPPE/UFRJ.Brasil. <http://www.cos.ufrj.br/uploadfiles/es67905.pdf>.
- BOCK, C. (2003). "UML 2 Activity and Action Models- Part 2- Actions", Journal of Object Technology, Vol. 2, No 4.
- BOEHM, B. W. (1981). "Software Engineering Economics", Prentice Hall.
- COLANZI, T. E. (1999). "Uma Abordagem Integrada de Desenvolvimento e Teste de Software Baseada na UML", 143p. Dissertação de M.Sc., Instituto de Ciências Matemáticas e Computação, USP, São Carlos, Brasil.
- COLEMAN, D., ARNOLD, P., BADOFF, S., DOLLIN, C., GILCHRIST, H., HAYES, F., JEREMAES, P. (1994). "Object-Oriented Development: The Fusion Method", New Jersey: Prentice Hall International, Englewood Cliffs.
- CONRADI, R., MARJARA, A., HANTHO, O., FROTVEIT, T., SKATOVIC, B. (1999). "A study of inspections and testing at Ericsson", Proceedings of the International Conference on Product Focused Software Process Improvement (PROFES'99), Oulu, Finland.
- COOPER, D. J. A., VON KONSKY, B. R., ROBEY, M. C., MCMEEKIN, D. A. (2007). "Obstacles to Comprehension in Usage Based Reading", 18th Australian Software Engineering Conference (ASWEC'07), Abril, Melbourne, Vic, pp. 233-244.

- DE MELLO, R. M., PEREIRA, W. M., TRAVASSOS, G. H. (2010). "Activity Diagram Inspection on Requirements Specification", XIV Simpósio Brasileiro de Engenharia de Software, Salvador, Bahia, Brasil, pp. 168-177.
- DE MELLO, R. M. (2011). "Técnica para Inspeção de Diagramas de Atividades", Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- FAGAN, M. E. (1986). "Advances in Software Inspections", IEEE Transactions on Software Engineering, Vol. SE-12, no.7.
- GARDNER, M. J., ALTMAN, D. G. (1989). "Statistics with Confidence: confidence intervals and statistical guidelines", . London: BMJ Publishing Group.
- GILB, T., GRAHAM, D. (1993). "Software Inspection", Addison-Wesley.
- GUTIÉRREZ, J. J., NEBUT, C., ESCALONA, M. J., MEJÍAS, M., RAMOS, I. M. (2008). "Visualization of use cases through automatically generated activity diagrams", Proceedings of the 11th International Conference on Model Driven Engineering Languages and Systems (MODELS'08), pp. 83-96.
- HE, L., CARVER, J. (2006). "PBR vs. Checklist: A Replication in the N Fold Inspection Context", ISESE'06, Setembro 2006, Rio de Janeiro, Brasil.
- HUNGERFORD, B. C., HEVNER, A. R., COLLINS, R. W. (2004). "Reviewing software diagrams: A cognitive study", IEEE Transactions on Software Engineering, Fevereiro, 30(2):82-96.
- IEEE (1990). "IEEE Standard glossary of software engineering terminology, Standard 610.12.", IEEE Press, <http://standards.ieee.org/findstds/standard/610.12-1990.html>
- ISMAIL, A., YAN, J., SHEN, J. (2009). "Verification of Composite Services with Temporal Consistency Checking and Temporal Satisfaction Estimation", In Proceedings of the 10th International Conference on Web Information Systems Engineering (WISE '09), Poznan, Poland, Outubro, Springer-Verlag, Berlin, Heidelberg, pp. 343-350.
- JALOTE, P. (2005). "An Integrated Approach to Software Engineering", 3 ed., Indian Institute of Technology Kanpur, Springer.
- KITCHENHAM, B. (2004). "Procedures for Performing Systematic Reviews", Joint Technical Report TR/SE-0401, Software Engineering Group, Department of Computer Science, Keele University, UK and Empirical Software Engineering National ICT Australia Ltd.

- KNIGHT, J. C., MEYERS, E.A. (1991). "Phased Inspections and their Implementation", Software Engineering Notes, Vol.16, No.3, pp. 29-35.
- LAITENBERGER, O., ATKINSON, C. (1999). "Generalized Perspective Based Inspection to handle Object Oriented Development Artifacts", Proceedings of ICSE 99, Los Angeles, CA, USA, pp.494-503.
- LAITENBERGER, O., EL ANAM, K., HARBICH, T. G. (2001). "An Internally Replicated Quasi-Experimental Comparison of Checklist and Perspective- Based Reading of Code Documents", IEEE Transactions on Software Engineering, vol. 27, No. 5.
- MAFRA, S. M. (2006). "Definição de uma Técnica de Leitura Baseada em Perspectiva (OO-PBR) Apoiada por Estudos Experimentais", Dissertação de M.Sc., COPPE/URFJ, Rio de Janeiro, RJ, Brasil
- MAFRA, S. N., BARCELOS, R. F., TRAVASSOS, G. H. (2006). "Aplicando uma Metodologia Baseada em Evidência na Definição de Novas Tecnologias de Software", In: XX SBES, Florianópolis, SC, Brasil.
- MARUCCI, R. A., FABBRI, S., MALDONADO, J. C., TRAVASSOS, G. H. (2002). "OORTs/ProDeS: Definição de Técnicas de Leitura para um Processo de Software Orientado a Objetos", In: Simpósio Brasileiro de Qualidade de Software, Gramado, Rio Grande do Sul.
- MASSOLLAR, J. L. (2011). "Uma Abordagem para Especificação de Requisitos Dirigida por Modelos Integrada ao Controle da Qualidade de Aplicações Web", Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- MASSOLLAR, J. L., DE MELLO, R. M., TRAVASSOS, G. H. (2011). "Investigating the Feasibility of a Specification and Quality Assessment Approach Suitable for Web Functional Requirements", Computer Science Society (SCCC) 30th International Conference of the Chilean, Curico, Chile, Novembro, pp. 108-117.
- OMG (2001). "Object Modeling with UML", Object Management Group, [http://www.omg.org/news/meetings/workshops/presentations/eai\\_2001/tutorial\\_monday/tockey\\_tutorial/1-Intro.pdf](http://www.omg.org/news/meetings/workshops/presentations/eai_2001/tutorial_monday/tockey_tutorial/1-Intro.pdf)
- OMG, (2010), "Unified Modeling Language Superstructure - version 2.3", Object Management Group, <http://www.omg.org/spec/UML/2.3/>

- OMG (2011). "Unified Modeling Language Superstructure - version 2.4.1", Object Management Group, <http://www.omg.org/spec/UML/2.4.1/>
- PAI, M. MCCULLOCH, M. GORMAN, J.D., PAI, N., ENANORIA W., KENNEDY, G., THARYAN, P., COLFORD., J. M. (2004). "Systematic Reviews and meta-analyses: An illustrated, step-by-step guide", The National Medical Journal of India, vol. 17, n.2.
- PETERSSON, H. (2002). "Supporting Software Inspections through Fault Content Estimation and Effectiveness Analysis", Technical report 147, Department of Communication Systems, Lund Institute of Technology.Lund University.
- PORTER, A. A., VOTTA, L. G. (1994). "An Experiment to Assess Different Defect Detection Methods for Software Requirements Inspections".Proceedings of 16th International Conference on Software Engineering, pp. 103-112.
- PORTER, A. A., VOTTA, L. G., BASILI, V. R. (1995). "Comparing detection methods for software requirements inspections: a replicated experiment", IEEE Transactions on Software Engineering, v.21, n. 6, pp. 563-575.
- REIS, L.N.M. (2005). "Apoio Automatizado para Aplicação de Técnicas de Leitura Orientada a Objetos (OORTs)", Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- SILVA, R. P. (2007). "UML 2 em Modelagem Orientada a Objetos", 1ª ed., Florianópolis, SC, Brasil, Visual Books.
- SHULL, F. (1988). "Developing Techniques for Using Software Documents: A Series of Empirical Studies", PhD Thesis, Department of Computer Science, University of Maryland, USA.
- SHULL, F., RUS, I., BASILI, V. R. (2000). "How Perspective-Based Reading Can Improve Requirements Inspections", Computer, Volume:33, Issue:7, Julho, pp. 73 – 79.
- SHULL, F., CARVER, J., TRAVASSOS, G. H. (2001). "An Empirical Methodology for Introducing Software Processes", Proceedings of the Joint 8th European Software Engineering Conference (ESEC) and 9th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-9), pp. 288-296.
- SHULL, F., CARVER, J., TRAVASSOS, G. H., MALDONADO, J. C., CONRADI,R., BASILI, V. (2003). "Replicated Studies: Building a Body of Knowledge about



- Software Reading Techniques”, Lecture Notes on Empirical Software Engineering, World Scientific Publishing Co., Inc., USA, NJ, pp. 39-84.
- SPÍNOLA, R. O., DIAS NETO, A. C., TRAVASSOS, G. H. (2008). “Abordagem para Desenvolver Tecnologia de Software com Apoio de Estudos Secundários e Primários”, In: Experimental Software Engineering Latin American Workshop (ESELAW), Salvador, Brasil.
- TANRIÖVER, Ö., BILGEN, S. (2007). “An Inspection Approach for Conceptual Models in Notations Derived from UML: A Case Study”, ISCIS 2007: 22nd International International Symposium on Computer and Information Sciences, Setembro, Ankara, Istanbul, pp. 1-6.
- THELIN, T., RUNESON, P., REGNELL, B. (2001). “Usage-Based Reading – An Experiment to guide reviewers with use cases”, Information and Software Technology, Elsevier, Dezembro, v 43, n 15, pp. 925-938.
- TRAVASSOS, G. H., SHULL, F., FREDERICKS, M., BASILI, V. R. (1999). “Detecting Defects in Object-Oriented Designs: Using Reading Techniques to Increase Software Quality”, Proc. 14th International Conference on Object Oriented Programming Systems, Languages & Applications (OOPSLA’99), New York, NY, USA, pp. 47-56.
- TRAVASSOS, G. H. apud ROCHA, A. R. C., MALDONADO, J. C., WEBER, K. C. (2001). “Qualidade de Software- Teoria e Prática”. São Paulo, 1ed., Prentice Hall.
- TRAVASSOS, G. H., SHULL, F., CARVER, J., BASILI, V. R. (2002). “Reading Techniques for OO Design Inspections”, Technical Report CS-TR-4353, University of Maryland Computer Science Department, 2002. Disponível em: <http://www.cs.umd.edu/Library/TRs>. Último acesso: março de 2013.
- TRAVASSOS, G. H., SANTOS, P. S. M. D., MIAN, P. G., DIAS NETO, A. C., BIOLCHINI, J. C. D. A. (2008). "An Environment to Support Large Scale Experimentation in Software Engineering", Proceedings of the 13th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS’08), pp. 193-202.
- WEN, S., LI, Q., YUE, L., LIU, A. (2009). “An Approach to Validating Transactional Properties of WS-BPEL Composition”, Fifth International Conference on Semantics, Knowledge and Grid, Zhuhai, China, pp. 216-226.

WONG, Y. K. (2006). "Modern Software Review- Techniques and Technologies", IRM Press.

WYNN, M. T., VAN DER AALST, W. M. P., TER HOFSTEDE, A. H. M., EDMOND, D. (2006). "Verifying workflows with cancellation regions and OR-joins: An approach based on reset nets and reachability analysis", 4th International Conference on Business Process Management, Vienna, Austria, Setembro, pp. 389-394.

# APÊNDICE A – Formulários Utilizados nos Estudos

*Este apêndice apresenta o formulário de consentimento e de caracterização utilizados nos dois estudos relatados nesta dissertação.*

## A.1. Formulário de Consentimento

### Estudos sobre Técnicas de Inspeção para Diagramas UML

Eu declaro ter mais de 18 anos de idade e que concordo em participar em estudos conduzidos pelo Prof. Guilherme Horta Travassos, como parte das atividades do curso de Engenharia de Software Orientado Objetos no Curso de Engenharia de Computação e Informação da UFRJ. Estes estudos visam melhor compreender a aplicação de técnicas de inspeção de software em diagramas UML de um projeto, tentando entender sob que condições estas técnicas podem ser melhor utilizadas.

#### **PROCEDIMENTO**

Uma técnica de inspeção aplicável a modelos UML será aplicada a diagramas específicos. Eu entendo que serei ensinado como utilizar a técnica de leitura e serei solicitado a aplicá-la em um exercício no decorrer do curso. Neste exercício alguns métodos experimentais serão aplicados por mim visando permitir pensar sobre seu uso e avaliá-la. Eu entendo que, uma vez o curso tenha terminado, o trabalho que desenvolvi poderá ser estudado visando entender a eficiência da técnica que me foi ensinada.

Eu entendo que este exercício preenche parte dos requisitos do curso e será avaliado como tal. O pesquisador conduzirá o estudo consistindo da coleta, análise e relato dos dados do exercício. Eu entendo que não tenho obrigação alguma em contribuir com informação sobre meu desempenho nos exercícios, e que posso solicitar a retirada de meus resultados do experimento a qualquer momento e sem qualquer penalidade ou prejuízo. Eu entendo que não existirá nenhum crédito ou benefício extra por participar deste estudo, e que não haverá qualquer impacto negativo em minha avaliação por não participar do estudo. Eu entendo também que quando os dados forem coletados e analisados, meu nome será removido dos dados e que este não será utilizado em nenhum momento durante a análise ou quando os resultados forem apresentados.

#### **CONFIDENCIALIDADE**

Toda informação coletada neste estudo é confidencial, e meu nome não será identificado em momento algum. Da mesma forma, me comprometo a não comunicar os meus resultados enquanto não terminar o estudo, bem como manter sigilo das técnicas e documentos apresentados e que fazem parte do experimento.

## **BENEFÍCIOS, LIBERDADE DE DESISTÊNCIA.**

Eu entendo que os benefícios que receberei deste estudo são limitados ao aprendizado do material que é distribuído e ensinado visando atender os requisitos do curso, independente de participar ou não deste estudo, mas que os pesquisadores esperam aprender mais sobre quão eficiente é a aplicação de técnicas de inspeção e os benefícios trazidos por este estudo para o contexto da Engenharia de Software.

Eu entendo que sou livre para realizar perguntas a qualquer momento ou solicitar que qualquer informação relacionada a minha pessoa não seja incluída no estudo. Eu entendo que minha participação no estudo não afetará minha nota final de qualquer forma, e que participo de livre e espontânea vontade com o único intuito de contribuir para o avanço e desenvolvimento de técnicas e processos para a Engenharia de Software.

## **PROFESSOR RESPONSÁVEL**

Prof. Guilherme Horta Travassos  
Programa de Engenharia de Sistemas e Computação  
COPPE/UFRJ

Nome (em letra de forma): \_\_\_\_\_

Assinatura: \_\_\_\_\_ Data: \_\_\_\_\_

## A.2. Formulário de Caracterização

### Caracterização do Inspetor

Nome \_\_\_\_\_

#### Formação Geral

Por favor, estime sua habilidade em utilizar material de trabalho em Inglês:

- Eu falo, leio e escrevo fluentemente.
- Considero o Inglês como sendo uma linguagem onde (Por favor, complete)  
Minhas habilidades de leitura e compreensão de textos:
  - poderiam ser melhores
  - são moderadas
  - são altas
  - são muito altas

Minha capacidade de trabalhar/seguir instruções escritas em Inglês:

- poderiam ser melhores
- são moderadas
- são altas
- são muito altas

Qual é sua experiência anterior com desenvolvimento de software na prática? (marque aqueles itens que melhor se aplicam)

- nunca desenvolvi software.
- tenho desenvolvido software para uso próprio.
- tenho desenvolvido software como parte de uma equipe, relacionado a um curso.
  - tenho desenvolvido software como parte de uma equipe, na indústria.

Qual é sua experiência anterior com inspeção na prática? (marque aqueles itens que melhor se aplicam)

- nunca inspecionei nenhum documento de software.
- inspecionei diagramas UML em projetos no passado.
- tenho inspecionado diagramas UML como parte de uma equipe, relacionado a um curso.
  - tenho inspecionado diagramas UML como parte de uma equipe, na indústria.

Por favor, explique sua resposta. Inclua o número de semestres ou número de anos de experiência relevante em desenvolvimento (E.g. “Eu trabalhei por 10 anos como programador na indústria”) (Caso tenha experiência com inspeções, relate o seu tempo de experiência e tipos de documentos inspecionados)

## Experiência em Inspeção de Software

Por favor, indique o grau de sua experiência nesta seção seguindo a escala de 5 pontos abaixo:

- 1 = nenhum
- 2 = estudei em aula ou em livro
- 3 = pratiquei em 1 projeto em sala de aula
- 4 = usei em 1 projeto na indústria
- 5 = usei em vários projetos na indústria

### Experiência com Diagramas UML

- Experiência com Unified Modeling Language (UML) 1 2 3 4 5
- Experiência com Diagramas de Atividades (UML) 1 2 3 4 5
- Experiência com Diagramas de Estados (UML) 1 2 3 4 5

### Experiência em Projeto

- Experiência modelando diagramas 1 2 3 4 5
- Experiência escrevendo casos de uso 1 2 3 4 5
- Experiência inspecionando casos de uso 1 2 3 4 5
- Experiência inspecionando diagramas de atividades 1 2 3 4 5
- Experiência inspecionando diagramas de estados 1 2 3 4 5
- Experiência inspecionando outros diagramas UML 1 2 3 4 5
- Experiência em projeto de sistemas 1 2 3 4 5
- Experiência usando modelos de projeto Orientado a Objetos 1 2 3 4 5

### Outras Experiências

- Experiência com gerenciamento de projeto de software? 1 2 3 4 5
- Experiência com desenvolvimento de software? 1 2 3 4 5
- Experiência em desenvolver projetos a partir de requisitos e casos de uso 1 2 3 4 5
- Experiência criando projetos Orientado a Objetos 1 2 3 4 5
- Experiência com testes de integração de software? 1 2 3 4 5

## Experiência em Contextos Diferentes

Nós usaremos esta seção para compreender quão familiar você está com vários sistemas que poderão ser utilizados como exemplos ou para exercícios durante o curso.

Por favor, indique o grau de experiência nesta seção seguindo a escala de 3 pontos abaixo:

- 1 = Eu não tenho familiaridade com a área. Eu nunca fiz isto.
- 3 = Eu utilizo isto algumas vezes, mas não sou um especialista.
- 5 = Eu sou muito familiar com esta área. Eu me sentiria confortável fazendo isto.

Quanto você sabe sobre...

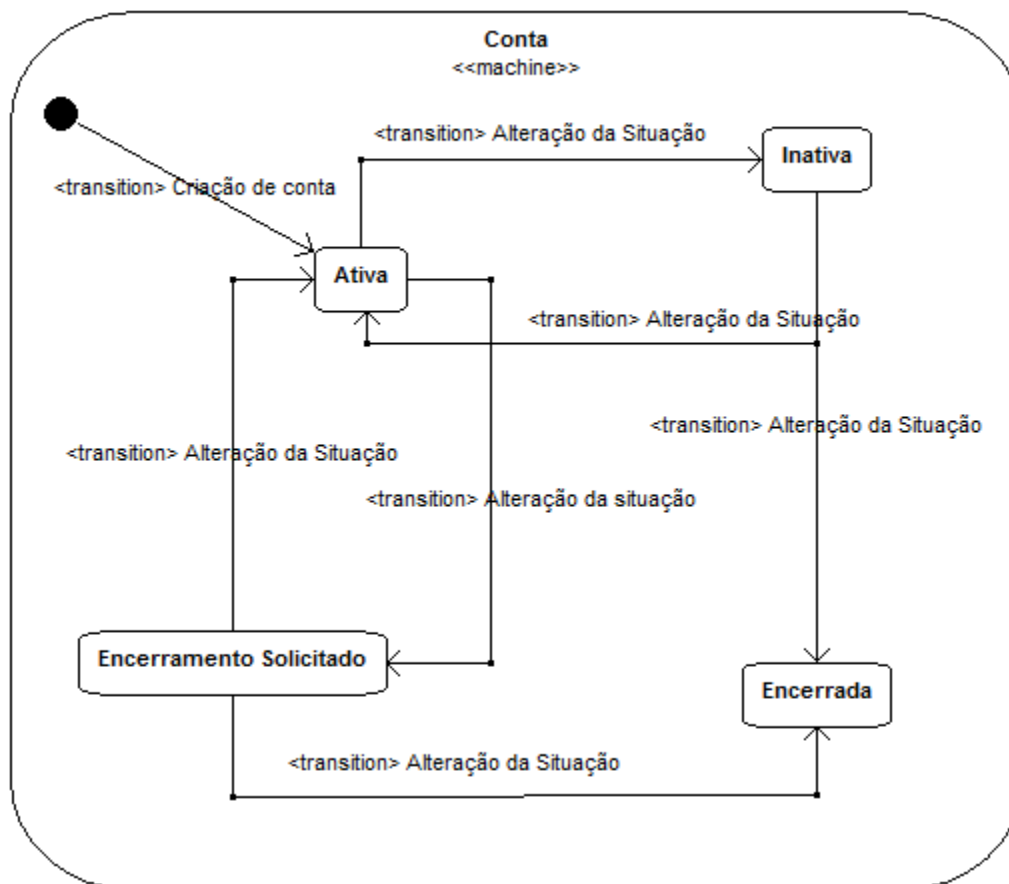
• Utilizar Ferramentas CASE?	1	3	5
• Movimentações Financeiras (contratos, pagamentos, etc.)?	1	3	5
• Realizar gerenciamento de bens patrimoniais?	1	3	5
• Utilizar os serviços de uma vídeo-locadora?	1	3	5

## APÊNDICE B – Diagramas utilizados nos Estudos

*Este apêndice apresenta os dois conjuntos de modelos utilizados nos estudos: Conta e Movimento.*

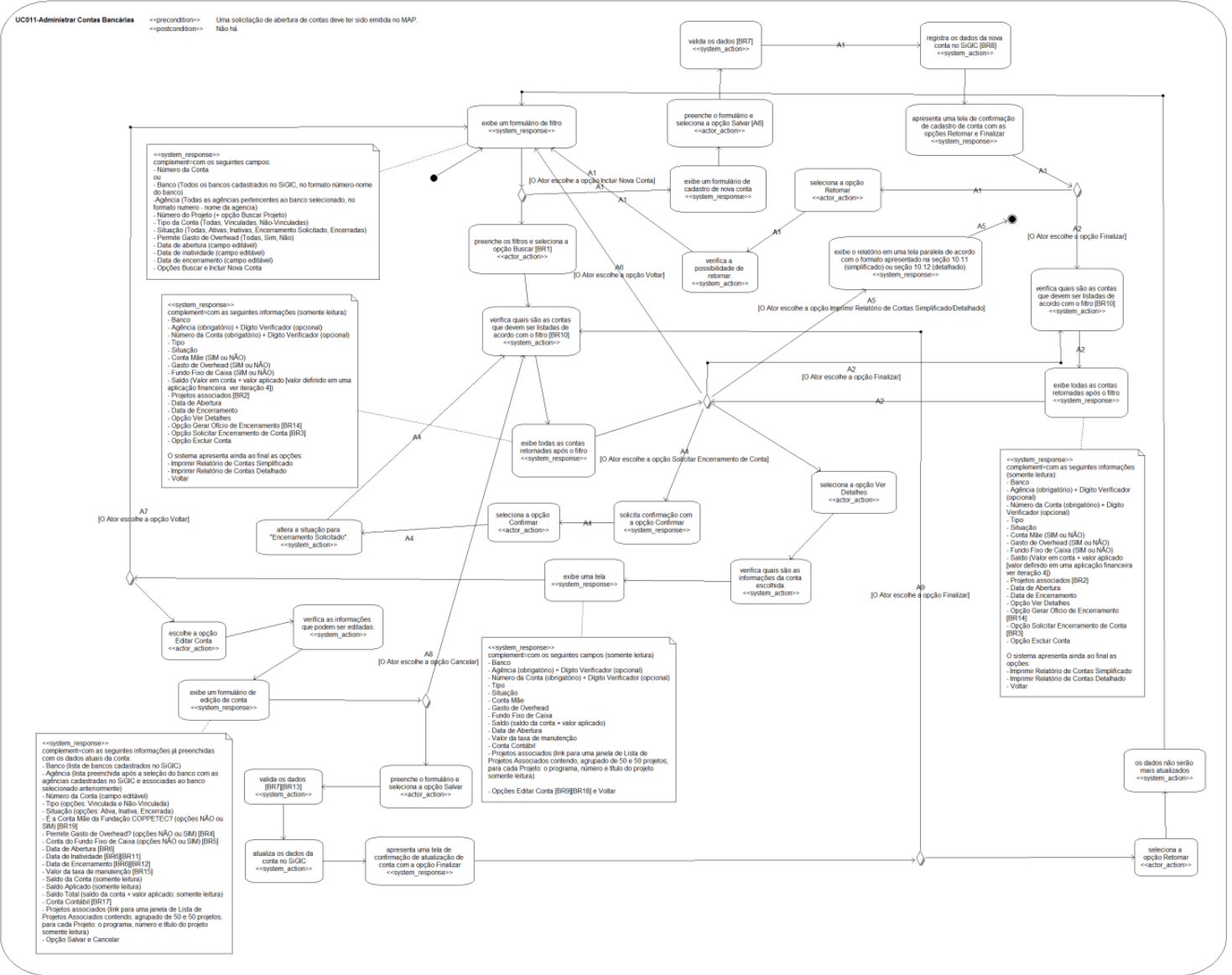
### B.1. Modelos e Regras de Negócio sobre Conta Utilizados na Inspeções

- Diagrama de estados Conta





• Diagrama de atividades Administrar Conta



- Regras de negócio de Administrar Conta

### Administrar Contas – Regras de Negócio

BR1-Todos os campos de filtro são opcionais.

BR2-Esta coluna exibirá o número total de projetos associados à conta em questão.

BR3-Esta coluna só estará visível caso o ator seja Gerente do Setor de Projetos Não-Vinculados ou Gerente de Projetos Vinculados. A opção de solicitação só estará ativa se a conta estiver com saldo zerado, se todos os projetos associados a ela estiverem encerrados e a conta estiver na Situação Ativa ou Inativa.

BR4-Apenas Contas do Tipo Não-Vinculadas podem permitir gasto de overhead. Caso o tipo seja Vinculada, esta opção deve estar desabilitada com a opção NÃO selecionada.

BR5-Apenas uma conta pode ser indicada como conta do fundo fixo de caixa da Fundação COPPETEC. Caso o tipo seja Vinculada, esta opção deve estar desabilitada com a opção NÃO selecionada.

BR6-Esta data deve ser menor ou igual à data atual do sistema.

BR7-Todos os campos são obrigatórios, respeitando as regras de negócio [RN4], [RN5] e [RN6].

BR8-Toda conta é criada com a situação “Ativa”, com saldo “0.00” e sem nenhum projeto associado a ela.

BR9-A opção Editar Conta só estará visível se o ator for Setor Financeiro ou Setor Contábil e caso a conta não esteja com a Situação Encerrada

BR10-Caso o ator seja Gerente do Setor de Projeto Não-Vinculado, apenas contas Não-Vinculadas podem ser filtradas. Caso o ator seja Gerente de Projeto Vinculado, apenas contas Vinculadas podem ser filtradas. Caso o ator seja Setor Financeiro ou Setor Contábil, qualquer conta pode ser filtrada.

BR11-Este campo só estará visível caso a Situação da conta seja Inativa.

BR12-Este campo só estará visível caso a Situação da conta seja Encerrada.

BR13-A situação da conta só pode ser Encerrada caso todos os projetos associados a ela estejam com o status Encerrado e caso seu saldo seja zero (R\$ 0,00). A situação da conta só pode ser Inativa caso todos os projetos associados a ela estejam com o status Inativo.

BR14-Esta opção só estará ativa para o ator Setor Financeiro e se a conta estiver com a situação Encerrada.

BR15-Este campo pode receber valor fracionário maior ou igual a 0.

BR16-Essa lista é preenchida com o conjunto de projetos que tiveram uma abertura de conta solicitada e que ainda estão pendentes de confirmação.

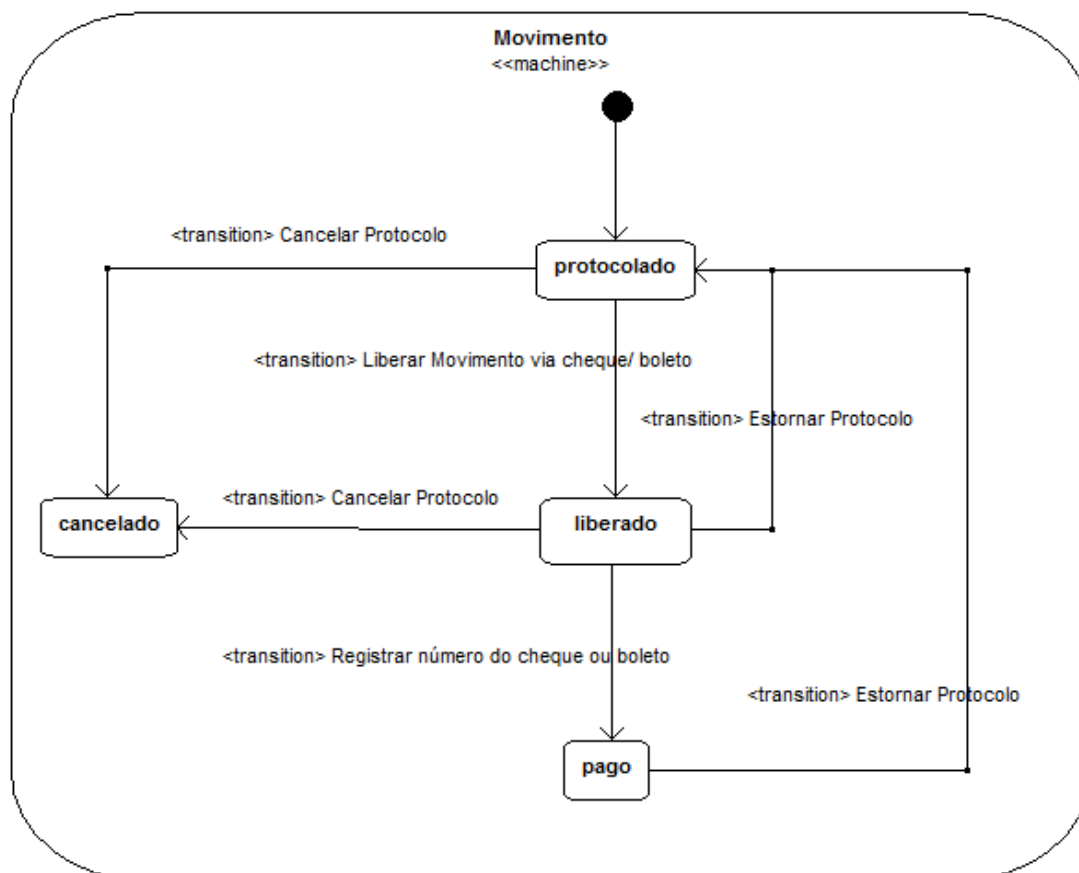
BR17-Este campo só poderá ser editado caso o ator seja Setor Contábil. Caso contrário, este campo será somente leitura.

BR18-Caso o ator seja Setor Financeiro, os campos que podem ser editados são: Banco, Agência, Número da Conta, Tipo, Situação, Indicador de Permite Gasto de Overhead?, Indicador de Conta do Fundo Fixo de Caixa, Data de Abertura, Data de Inatividade, Data de Encerramento e Valor da taxa de manutenção. Caso o ator seja Setor Contábil, o único campo que pode ser editado é Conta Contábil. Os demais deverão ser somente leitura.

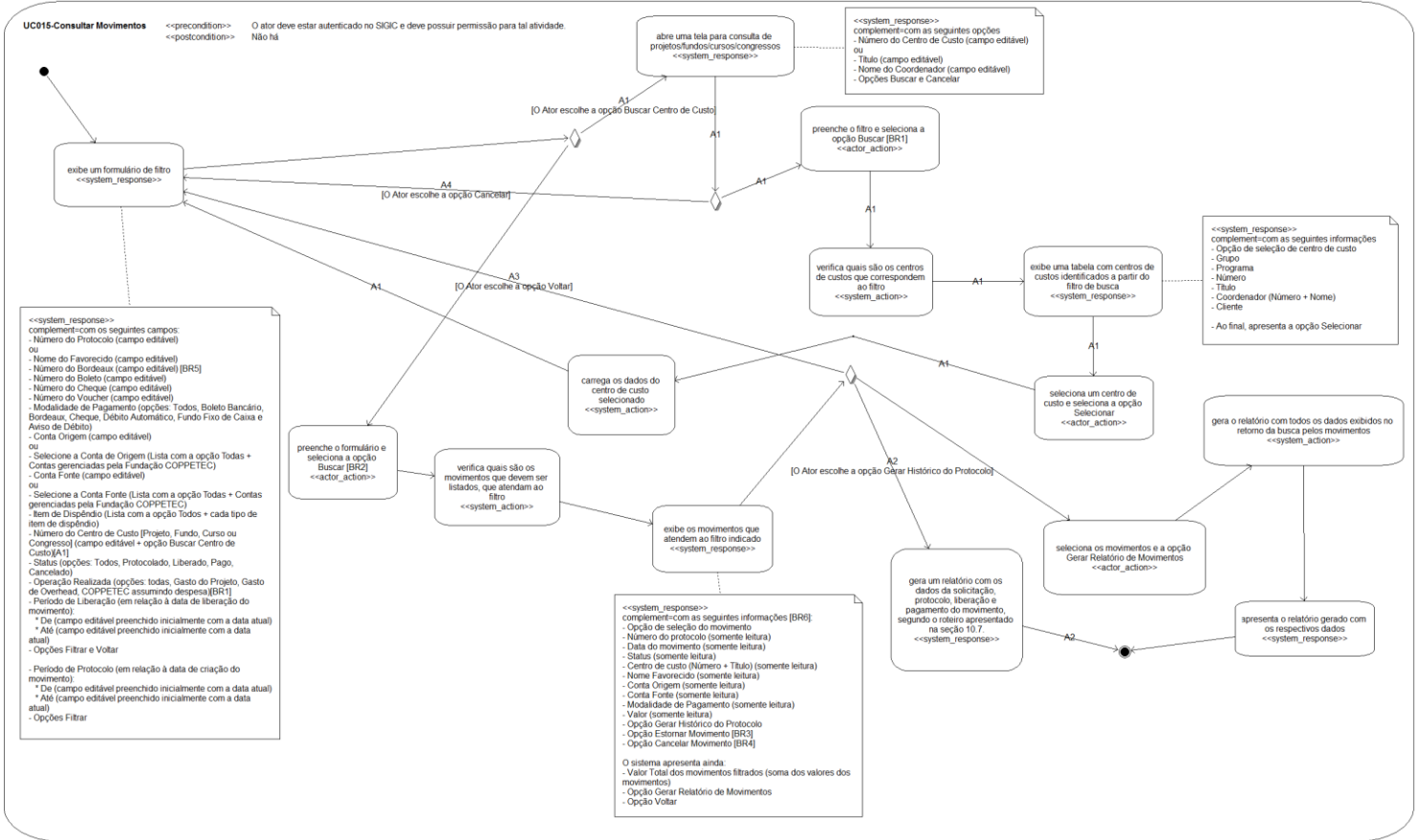
BR19-Apenas uma conta pode ser indicada como Conta Mãe da Fundação COPPETEC. Caso o tipo seja Vinculada, esta opção deve estar desabilitada com a opção NÃO selecionada.

## B.2 Modelos e Regras de Negócio sobre Movimento Utilizados na Inspeções

- Diagrama de estados Movimento



- Diagrama de atividades Consultar Movimento



- Regras de negócio de Consultar Movimento

### Consultar Movimentos – Regras de Negócio

BR1-A operação “Gasto do Projeto” é identificada quando a conta origem e conta fonte de um movimento são as mesmas. A operação “Gasto de Overhead” é identificada quando a conta origem é referente a um Projeto Não-Vinculado e conta fonte é referente a um projeto Vinculado. A operação “COPPETEC assumindo despesa” é identificada quando a conta origem é diferente da conta fonte, e a conta fonte é uma conta de projeto não-vinculado.

BR2-Todos os campos de filtro são opcionais, com exceção do período que é obrigatório.

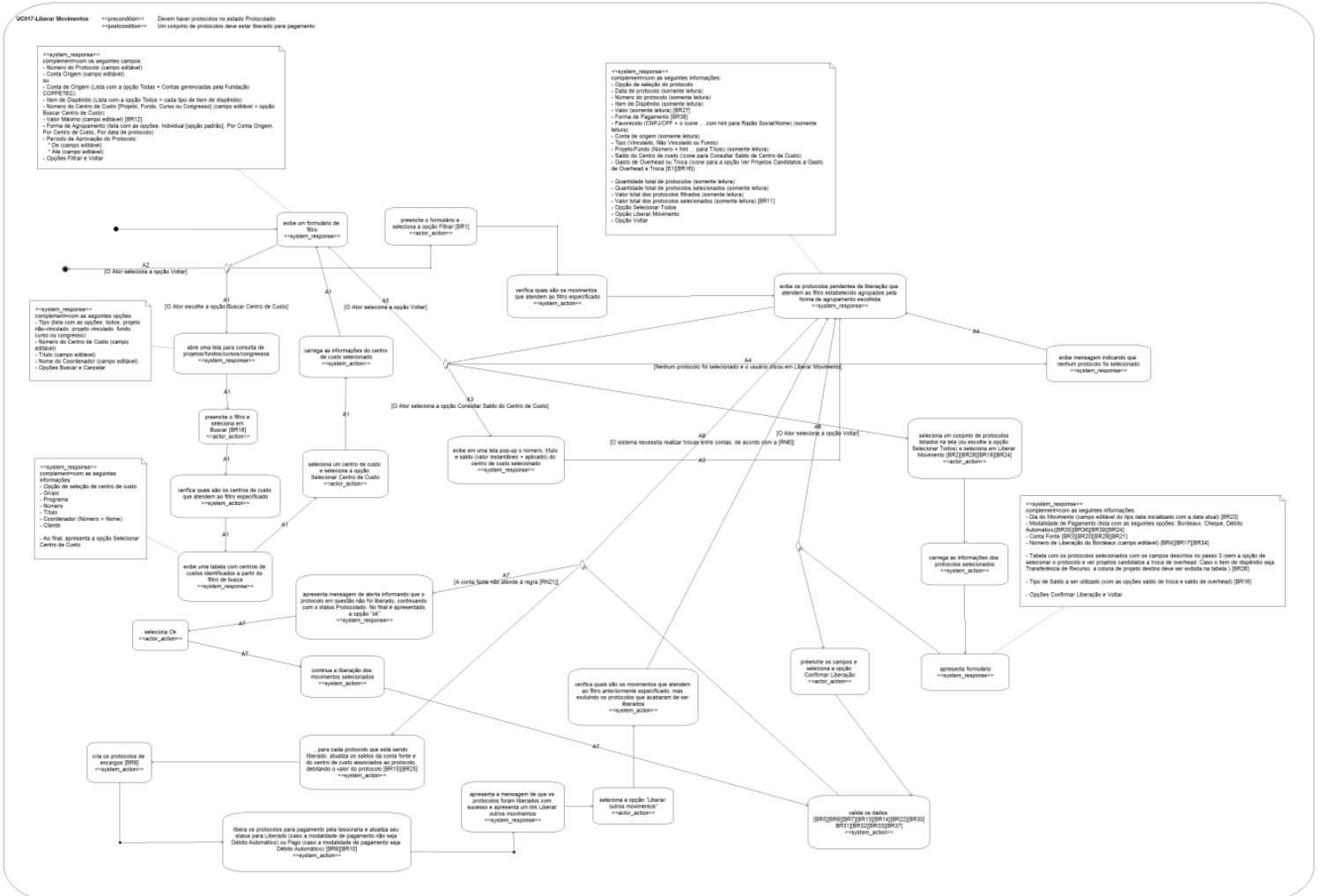
BR3-A opção Estornar Movimento só estará ativa se movimento estiver com o status “Liberado” E se as possíveis trocas entre contas geradas a partir da liberação deste movimento não tenham sido processadas (através do UC13).

BR4-A opção Cancelar Movimento só estará ativa se movimento não estiver com o status “Cancelado”

BR5-O campo Número do Bordeaux aceita apenas números.

BR6-Os movimentos são listados agrupados de acordo com a modalidade de pagamento em que foi liberada (Boleto, Bordeaux, Cheque, Débito Automático, Fundo Fixo de Caixa e Aviso de débito).

- Diagrama de atividades Liberar Movimento



- Regras de negócio de Liberar Movimento

### Liberar Movimentos – Regras de Negócio

BR1-Todos os campos de filtro são opcionais. Apenas protocolos originados de uma solicitação de coordenador podem ser exibidos, ou seja, protocolos de encargos não devem ser visualizados.

BR2-O sistema deve permitir que mais que um protocolo possa ser selecionado simultaneamente para liberação.

BR3-Ao selecionar uma Conta Fonte, o sistema deve exibir ao lado o saldo da conta selecionada. Caso esteja ocorrendo um GASTO DE OVERHEAD, deve ser exibido o saldo de OVERHEAD da conta selecionada, e não o saldo do projeto.

BR4-Este campo só passa a ser visível caso a modalidade de pagamento Bordeaux for escolhida no campo Modalidade de Pagamento. Caso contrário, não deve ser visualizado.

BR5-Todos os campos são obrigatórios. O Dia do Movimento não pode ser inferior à data atual.

BR6-As seguintes regras de validação de contas devem ser atendidas:

- Se a conta origem for referente a um projeto não-vinculado, a conta fonte só pode ser:

(1) a própria conta origem: isso indica que é um gasto do próprio projeto (o protocolo está sendo liberado pela própria conta corrente do projeto), o que não resultará na necessidade de troca entre contas. Esse é o procedimento padrão. Nessa situação o sistema não solicita confirmação pelo usuário.

(2) a conta mãe da Fundação COPPETEC: isso indica que a COPPETEC está assumindo a despesa de um projeto e com isso será necessária uma troca entre contas a partir de um ofício de troca a ser gerado [RN10]. Neste caso, o sistema deve apresentar uma tela de confirmação da operação [A4].

(3) a conta Caixa da Fundação COPPETEC: isso indica que a COPPETEC está assumindo a despesa de um projeto que será paga pelo caixa da fundação COPPETEC e com isso será necessária uma troca entre contas a partir de um ofício de troca a ser gerado [RN10]. Neste caso, o sistema deve apresentar uma tela de confirmação da operação [A4].

(4) uma conta referente a um projeto vinculado que possui saldo de overhead no plano de contas associado ao protocolo que está sendo liberado: isso indica que está ocorrendo um gasto de overhead. Nesse caso, o sistema deve verificar se a conta origem permite gasto de overhead (parâmetro PERMITE GASTO DE OVERHEAD definido no UC15) e existe saldo de overhead na conta fonte suficiente no plano de contas selecionado ao protocolo que está sendo liberado (que foi definido durante a aprovação do protocolo). Neste caso, o sistema deve apresentar uma tela de confirmação da operação [A4]. Caso alguma das duas regras citadas acima não seja satisfeita (conta origem não permite gasto de



overhead ou conta fonte não possui saldo de overhead no plano de contas), o sistema não deve permitir a liberação do movimento, apresentando uma mensagem de alerta com o motivo da não permissão [A9].

(5) uma conta referente a um projeto vinculado que possui saldo de troca no plano de contas associado ao protocolo que está sendo liberado: isso indica que está ocorrendo um gasto de troca. Nesse caso, o sistema deve verificar se existe saldo de troca na conta fonte suficiente no plano de contas selecionado ao protocolo que está sendo liberado (que foi definido durante a aprovação do protocolo). Neste caso, o sistema deve apresentar uma tela de confirmação da operação [A4]. Caso a regra citada acima não seja satisfeita (conta fonte não possui saldo de troca no plano de contas), o sistema não deve permitir a liberação do movimento, apresentando uma mensagem de alerta com o motivo da não permissão [A9].

- Se a conta origem for referente a um projeto vinculado, a conta fonte só pode ser:

(1) a própria conta origem: isso indica que é um gasto do próprio projeto (o protocolo está sendo liberado pela própria conta corrente do projeto), o que não resultará na necessidade de troca entre contas. Esse é o procedimento padrão. Nessa situação o sistema não solicita confirmação pelo usuário.

(2) a conta mãe da Fundação COPPETEC: isso indica que a COPPETEC está assumindo a despesa de um projeto e com isso será necessária uma troca entre contas a partir de um ofício de troca a ser gerado [RN10]. Neste caso, o sistema deve apresentar uma tela de confirmação da operação [A4].

(3) a conta Caixa da Fundação COPPETEC: isso indica que a COPPETEC está assumindo a despesa de um projeto que será paga pelo caixa da fundação COPPETEC e com isso será necessária uma troca entre contas a partir de um ofício de troca a ser gerado [RN10]. Neste caso, o sistema deve apresentar uma tela de confirmação da operação [A4].

BR7-Caso o valor do protocolo seja superior a um parâmetro pré-estabelecido no sistema para pagamento via Cheque definido pelo banco associado à conta fonte selecionada (Isso é definido durante a administração de contas gerenciadas pela Fundação COPPETEC [UC 40 – Administrar Bancos]. Atualmente o valor limite é de R\$ 5.000,00 para o Banco de Brasil), o sistema deve lançar uma despesa a ser paga pelo projeto referente à tarifa bancária para pagamento via cheque definida como parâmetro do banco associado à conta fonte selecionada (Isso é definido durante a administração de contas gerenciadas pela Fundação COPPETEC [UC 40 – Administrar Bancos]. Atualmente a tarifa é de 0,11% para o Banco de Brasil) como um lançamento extra orçamentário com os seguintes dados:

- Centro de Custo: o centro de custo associado ao protocolo que está sendo liberado.
- Valor: x% do valor do protocolo, onde x é definido como parâmetro do SiGIC (atualmente é de 0,11%).
- Data de Extrato: O dia do movimento definido no passo 6 do fluxo principal

- Conta fonte: a conta fonte para pagamento do protocolo definida no passo 6 do fluxo principal

- Plano de Contas:

\* Plano de contas financeiro: 200.001.537

\* Plano de contas institucional: 2.3.1.6.0002 (caso o centro de custo seja um projeto não-vinculado ou fundos) ou 3.2.2.0.0002 (caso o centro de custo seja um projeto vinculado)

- Histórico: “Tarifa Bancária em <dia do movimento> referente a Pagamento a fornecedores do protocolo <número do protocolo>”.

Caso a conta fonte do lançamento extra orçamentário seja diferente da conta fonte do centro de custo, uma troca deve ocorrer entre elas

BR8-Encargos são gerados apenas caso o item de dispêndio seja do tipo Pagamento de PJ, seguindo as regras de definição para cada encargo definidas no “UC 14 – Protocolar Pagamento de Pessoa Jurídica”, no Módulo de Protocolo (MPT). Para cada encargo de protocolo deve ser registrado: dia de criação como sendo o dia do movimento (definida no passo 6 do fluxo principal), número protocolo mãe (número protocolo referente ao movimento que gerou o encargo), valor (calculado durante a criação do protocolo mãe) e tipo do encargo (definido durante a criação do protocolo mãe).

BR9-O protocolo não é liberado para tesouraria apenas em caso de solicitação de transferência entre projetos não-vinculados. Nesse caso, ele recebe o status de “Pago”.

BR10-Um registro de ofício de troca entre contas deve ser gerado e armazenado no sistema caso ocorra:

- Troca entre projetos não-vinculados

- Troca entre projeto vinculado e contas de projeto não-vinculado (quando a COPPETEC está bancando uma despesa de projeto vinculado)

- Gasto de overhead quando a conta dos fundos do plano de aplicação é diferente da conta de origem do protocolo

- Gasto de troca quando a conta do Fundo COPPETEC 03 é diferente da conta de origem do protocolo

O sistema deve registrar o ofício armazenando informações sobre a conta origem, a conta destino, o valor da troca e o tipo (débito [a conta origem deve pagar à conta fonte]), e ficará como pendente. O número do ofício é sequencial no formato NÚMERO/ANO e deve ser inicializado a cada ano, ou seja, o primeiro ofício de 2009 terá o número 1/2009, o segundo, 2/2009 e assim por diante. A confirmação da operação de troca e geração do documento físico de ofício de troca entre contas só ocorre através do UC 13 – Efetuar Troca entre Contas.

BR11-O valor total dos protocolos selecionados corresponde à soma dos valores dos protocolos selecionados pelo usuário.

BR12-O preenchimento do campo Valor Máximo resultará no filtro de todos os protocolos que possuem valor inferior ou igual ao valor informado pelo usuário.

BR13-Caso o valor do protocolo seja menor ou igual ao parâmetro pré-estabelecido no sistema para pagamento via Fundo Fixo de Caixa (atualmente o valor limite é de R\$ 100,00), o sistema deve permitir a liberação do protocolo para a modalidade Fundo Fixo de Caixa” sem qualquer necessidade de confirmação e a conta fonte obrigatoriamente será a conta definida como o caixa da Fundação COPPETEC (Isso é definido durante a administração de contas gerenciadas pela Fundação COPPETEC [UC 15 – Administrar Contas Bancárias]. Atualmente a conta caixa é a conta 133.212-5 do UNIBANCO). Caso o valor seja maior que o parâmetro estabelecido via Fundo Fixo de Caixa (atualmente o valor é de R\$ 3000,00), o sistema não deve permitir a liberação.

BR14-Caso a modalidade de pagamento escolhida for Bordeaux, o sistema deve lançar uma despesa a ser paga pelo projeto referente à tarifa bancária para transação eletrônica definida como parâmetro do banco associado à conta fonte selecionada (atualmente a tarifa é de R\$ 2,50 por movimento para o Banco de Brasil) como um lançamento extra orçamentário, com os seguintes dados.

- Centro de Custo: o centro de custo associado ao protocolo que está sendo liberado.
- Valor: R\$ X, onde x é definido como parâmetro do SiGIC (atualmente é de R\$ 2,50).
- Data de Extrato: O dia do movimento definido no passo 6 do fluxo principal
- Conta fonte: a conta fonte para pagamento do protocolo definida no passo 6 do fluxo principal
- Plano de Contas:
  - \* Plano de contas financeiro: 200.001.537
  - \* Plano de contas institucional: 2.3.1.6.0002 (caso o centro de custo seja um projeto não-vinculado ou fundos) ou 3.2.2.0.0002 (caso o centro de custo seja um projeto vinculado)
- Histórico: “Tarifa Bancária em <dia do movimento> referente a Pagamento a fornecedores do protocolo <número do protocolo>”.

BR15-Caso na liberação do protocolo esteja ocorrendo um gasto de overhead (Conta Fonte de um Projeto Vinculado e Conta Origem de um Projeto Não-Vinculado), o sistema deve:

1. Atualizar o saldo de overhead do projeto associado à conta fonte no plano de contas definido para o protocolo que está sendo liberado debitando o valor do protocolo.
2. Identificar o plano de aplicação definido para o Projeto Vinculado associado à Conta Fonte selecionada

3. Distribuir o mesmo valor do protocolo entre os fundos que compõem o plano de aplicação do Projeto Vinculado (ex: Fundo 1 recebe 30% do valor, Fundo 2 recebe 40% e Fundo 3 recebe 30%). Isso implicará na atualização do saldo dos fundos e do saldo de overhead do projeto vinculado

4. Caso as contas correntes associadas a cada fundo seja diferente da conta fonte do protocolo liberado, uma troca deve ocorrer da conta fonte para a conta corrente de cada fundos que compõem o plano de aplicação do projeto vinculado que esteja associado a uma conta diferente da conta origem do protocolo [RN10] (ex: A conta origem do protocolo é 312.518, mas a conta dos fundos é 330.871, então deve ocorrer uma troca da conta origem para a conta dos fundos no valor do protocolo liberado)

BR16-Esta opção só estará ativa caso o projeto associado ao protocolo seja não-vinculado, e sua conta origem possibilite o gasto de overhead (tal informação é definida durante a administração de contas gerenciadas pela Fundação COPPETEC [UC 15 – Administrar Contas Bancárias]). O saldo de overhead e de troca apresentados são referentes ao projeto selecionado que está relacionado à conta em questão. Este projeto é o único na conta que possibilita gasto de overhead e de troca.

BR17-O número do Bordeaux deve ser gerado automaticamente e é um valor sequencial. Reutilizar o número de Bordeaux, caso haja um outro número de Bordeaux utilizado anteriormente, desde que satisfaça as seguintes condições: com conta fonte igual ao do projeto, com a conta origem igual para todos os protocolos e que não tenha sido emitido e enviado ao banco.

BR18-Todos os campos de filtro são opcionais.

BR19-Protocolos de Ajuda a Vistante Estrangeiro não podem ser liberados em conjunto com outros tipos de item de dispêndio.

BR20-O sistema deve apresentar apenas as seguintes contas (nessa ordem) (após a seleção da conta, o seu saldo da conta será apresentado. Caso seja negativo, aparecerá na cor vermelha):

- A opção “Pagar com a própria conta do projeto”, que indica que cada movimento listado será pago através da sua conta origem, ou seja, a conta corrente associada ao projeto que realizou tal despesa.

- A Conta Mãe da Fundação COPPETEC, exibida no formato “Conta Mãe: [Banco-Agência-Conta] (definida no UC 15 – Administrar Contas Bancárias).

- A Conta Caixa da Fundação COPPETEC, exibida no formato “Conta Mãe: [Banco-Agência-Conta] (definida no UC 15 – Administrar Contas Bancárias).

- Lista com Contas Vinculadas que estão ativas e que possuem previsão de gastos de overhead em algum dos planos de contas dos protocolos selecionados (pois possibilitará gasto de overhead), exibidas no formato “Conta Vinculada: [Banco-Agência-Conta] [RN21].

- Lista com Contas Vinculadas que estão ativas e que possuem saldo de troca em algum dos planos de contas dos protocolos selecionados, exibidas no formato “Conta Vinculada: [Banco-Agência-Conta] [RN21].

BR21-Contas Vinculadas só devem ser exibidas caso algum dos protocolos selecionados sejam associados a um projeto não-vinculado cuja conta origem que possibilite gasto de overhead (parâmetro PERMITE GASTO DE OVERHEAD definido no UC15) ou caso a conta vinculada possua saldo de troca no plano de contas associado a um dos protocolos selecionados.

BR22-Se o centro de custo do protocolo for o fundo COPPETEC 2799 e a operação for um gasto de overhead [RN6], o sistema deve (além de seguir todo o processamento normal de débito do valor na conta fonte, trocas e pagamento de tarifas), criar um lançamento extra orçamentário de crédito com o mesmo valor do protocolo na conta origem do fundo com os seguintes dados:

- centro de custo: Fundo COPPETEC 2799;
- valor: valor do protocolo que está sendo liberado;
- plano de contas: 900 (plano de contas financeiro) e 4.1.1.0.0002 (plano de contas institucional);
- conta fonte: conta origem do fundo COPPETEC 2799;
- data do extrato: dia do movimento (preenchido no passo 6 do fluxo principal)
- histórico: “Pagamento de Nota de Débito em <dia do movimento> referente ao protocolo <número do protocolo>”.

BR23-Caso a data de liberação de movimento seja uma data futura, as operações em conta não são realizadas. É feito apenas o comprometimento do saldo do projeto considerando o valor do protocolo e também os encargos. As demais operações de liberação só serão efetivadas na data definida como sendo data de liberação.

BR24-Protocolos de Ajuda a Visitante Estrangeiro só podem ter como modalidade de pagamento o cheque.

BR25-Caso na liberação do protocolo esteja ocorrendo um gasto de troca (Conta Fonte de um Projeto Vinculado com saldo de troca suficiente e Conta Origem de um Projeto Não-Vinculado), o sistema deve:

1. Atualizar o saldo de troca do projeto associado à conta fonte no plano de contas definido para o protocolo que está sendo liberado, debitando o valor do protocolo.
2. Transferir o valor do protocolo a partir do centro de custo do protocolo para o Fundo COPPETEC 03. Isso implicará na atualização do saldo do fundo e do saldo do centro de custo do protocolo
3. Atualizar o saldo das contas correntes (conta fonte, conta origem e conta do Fundo COPPETEC 03) envolvidas na operação. Caso a conta corrente associada ao Fundo COPPETEC 03 seja diferente da conta fonte do protocolo liberado, uma troca deve ocorrer da conta fonte para a conta corrente do fundo [RN10] (ex: A conta origem do protocolo é 312.518, mas a conta do fundo

é 330.871, então deve ocorrer uma troca da conta origem para a conta do fundo no valor do protocolo liberado)

BR26-Somente protocolos de Pagamento de PJ e Importação devem permitir gasto de overhead.

BR27-Valor corresponde ao valor líquido (valor bruto menos os encargos, no caso de Pagamento de PJ).

BR28-Protocolos do item de dispêndio Transferência de Recurso não podem ser liberados juntamente com outros tipos de protocolo.

BR29-Protocolos de item de dispêndio Transferência de Recurso só podem ser pagas pela própria conta do projeto.

BR30-Ofício de troca entre bancos devem ser gerados caso o item de dispêndio seja Transferência de Recurso e caso a conta do projeto destino seja diferente da conta do projeto de origem.

BR31-Dois os mais protocolos só podem ser liberados por Bordeaux e utilizando a própria conta do projeto para pagar, caso os protocolos tenham a mesma conta origem.

BR32-Liberação de protocolos por Bordeaux não podem utilizar como conta fonte a Conta Caixa.

BR33-O protocolo só poderá ser liberado caso, haja saldo de troca disponível. Caso seja um protocolo de pagamento de fornecedores, na mensagem de erro, deve ser informado o valor bruto do protocolo.

BR34-Caso a liberação por Bordeaux, o nº de Bordeaux deve ser reutilizado caso haja uma outra liberação no mesmo dia e com a mesma conta fonte. Caso seja utilizando a própria conta do projeto, as contas origens também devem ser iguais para a reutilização do nº.

BR35-Em movimentos de Ajuda de Visitante Estrangeiro, apenas a modalidade de pagamento Cheque poderá ser escolhida.

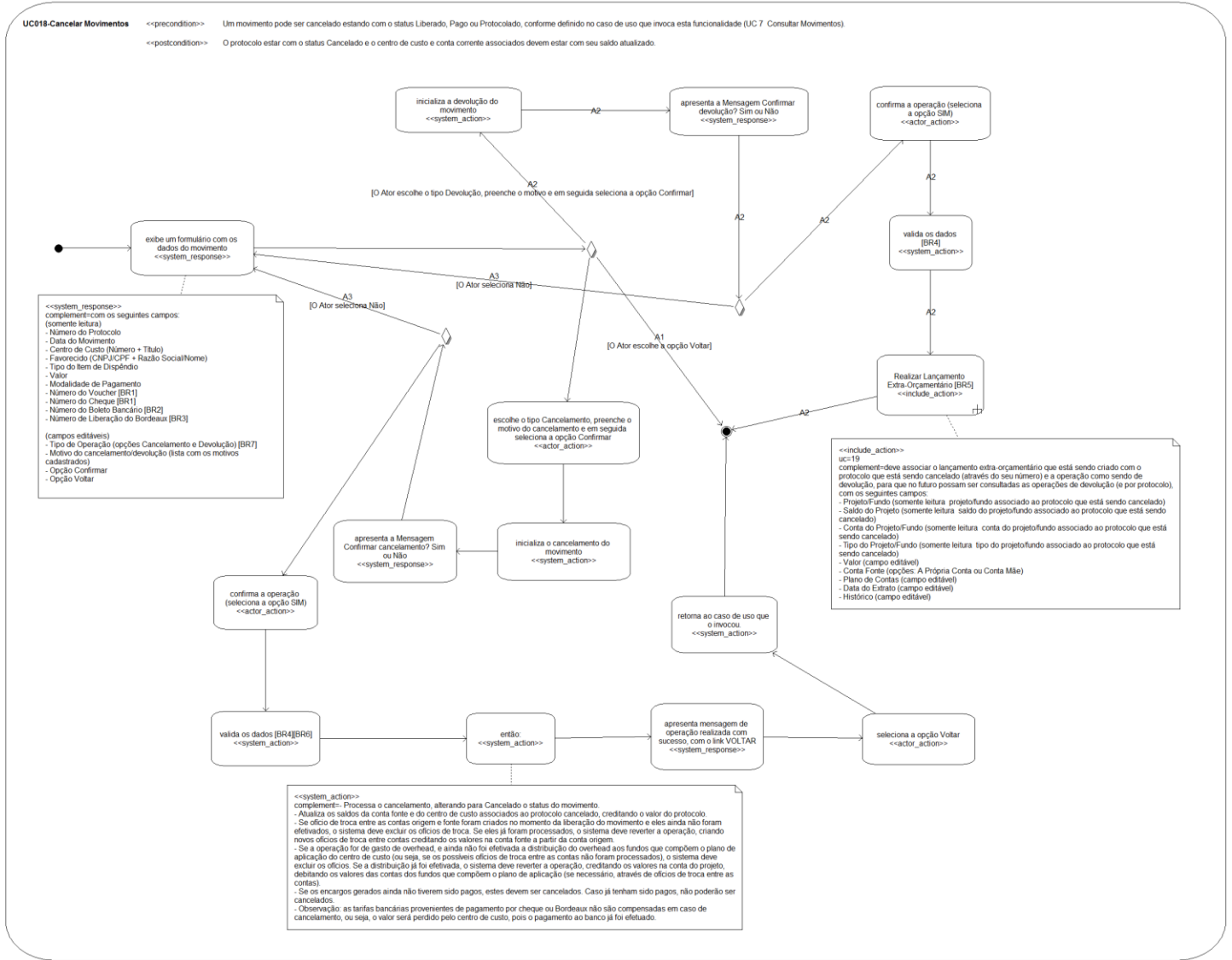
BR36-Em movimentos de Transferência de Recurso, apenas a modalidade de pagamento Débito Automático poderá ser escolhida e a conta fonte como própria conta do projeto.

BR37-No caso de um protocolo de pagamento de PJ, na liberação, a conta que está pagando deverá ter saldo suficiente para pagar o valor bruto (com encargos), porém apenas o valor líquido (sem encargos) deverá ser debitado

BR38-Forma de pagamento referente à escolha durante a criação do protocolo. A forma de pagamento se refere como será recebido o valor do protocolo.

BR39-Protocolos criados com forma de pagamento Cheque, só poderão ser liberados via Cheque, visto que não possuem dados bancários para outras modalidades de pagamento.

- Diagrama de atividades Cancelar Movimento



- Regras de negócio de Cancelar Movimento

#### Cancelar Movimento – Regras de Negócio

BR1-Este campo só estará visível caso a modalidade de pagamento seja Cheque.

BR2-Este campo só estará visível caso a modalidade de pagamento seja Boleto Bancário.

BR3-Este campo só estará visível caso a modalidade de pagamento seja Bordeaux.

BR4-O campo Motivo é obrigatório.

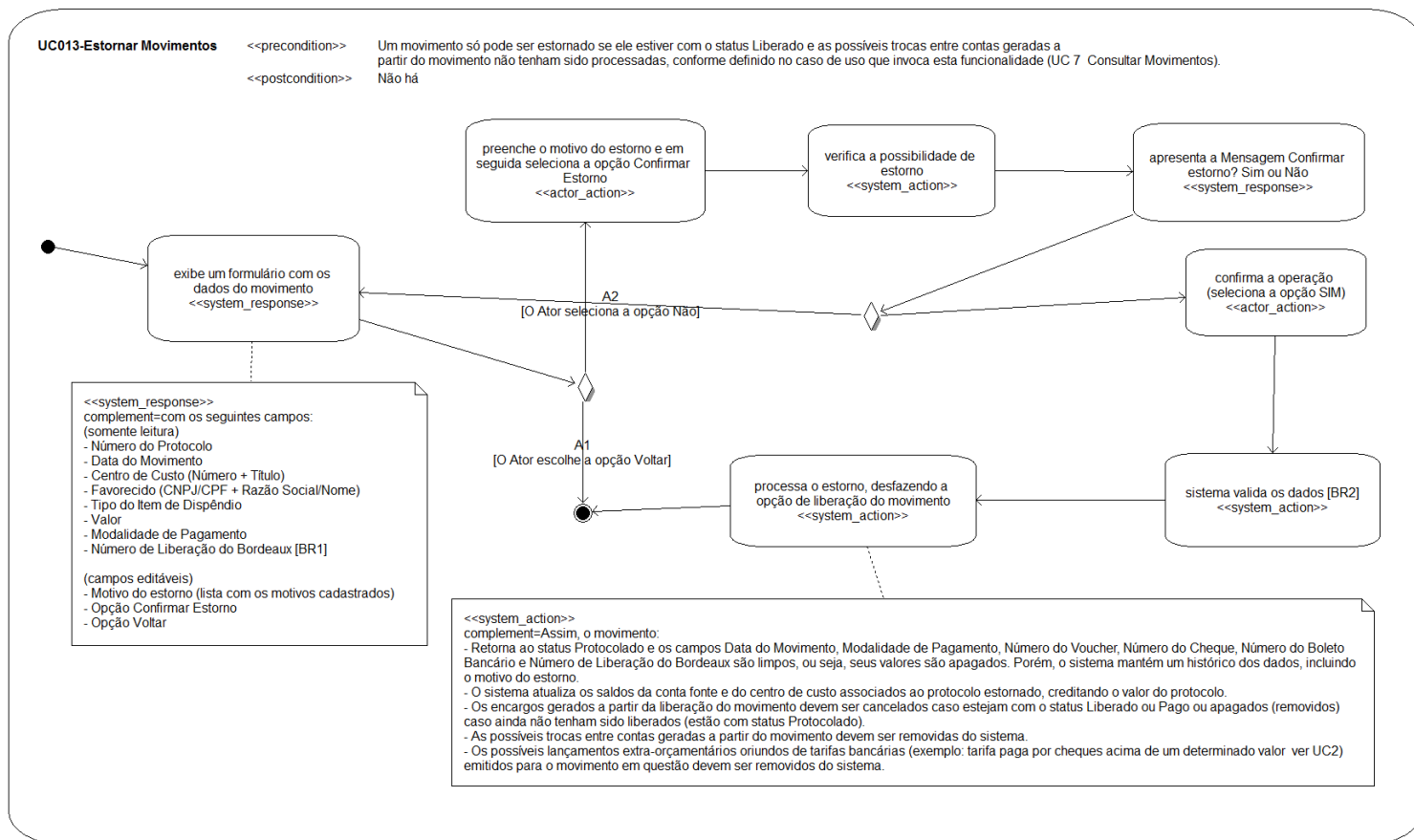
BR5-A operação de Devolução não cancela o movimento e nem altera seu status para CANCELADO. Ela simplesmente resulta na criação de um Lançamento Extra Orçamentário de débito para compensar o valor devolvido por um cliente.

BR6-O cancelamento só pode ser realizado caso o Bordeaux com o pagamento do protocolo em questão não tenha sido já enviado ao banco para pagamento (quando a modalidade de pagamento for Bordeaux).

BR7-Caso o status do movimento seja Liberado, a opção Devolução não poderá ser escolhida, apenas a opção Cancelamento



- Diagrama de atividades Estornar Movimento



- Regras de negócio de Estornar Movimento

Estornar Movimentos – Regras de Negócio

BR1-Este campo só estará visível caso a modalidade de pagamento seja Bordeaux.

BR2-O campo Motivo é obrigatório.



- Regras de negócio de Efetuar Pagamento

### Efetuar Pagamento – Regras de Negócio

BR1-Todos os campos de filtro são opcionais, com exceção do campo Modalidade de Pagamento.

BR2-Ao apresentar este campo, um número de voucher deve ser gerado automaticamente para o protocolo selecionado e preenchido no campo “Número do Voucher”. Os Vouchers devem ser numerados sequencialmente.

BR3-O campo Número do Cheque é obrigatório e não pode ter sido utilizado anteriormente para outros protocolos.

BR4-Este campo de filtro é obrigatório. O usuário deve escolher entre as modalidades Bordeaux, Cheque, Fundo Fixo de Caixa ou Boletão

BR5-Nesta funcionalidade, apenas movimentos cuja modalidade de pagamento for Bordeaux, Cheque, Boletão ou Fundo Fixo de Caixa serão filtrados, mesmo que o número de um protocolo não pago por essas modalidades de pagamento seja informado.

BR6-Este campo só estará preenchido, caso a modalidade de pagamento tenha sido Bordeaux.

BR7-Campo obrigatório a ser preenchido e o mesmo deve ser único.

BR8-Vários movimentos podem ser selecionados, respeitando as seguintes regras:

- Para movimentos com modalidade de Bordeaux: todos têm que ter a mesma conta fonte e número de liberação de Bordeaux.
- Para movimentos com modalidade de cheque: todos têm que ter a mesma conta fonte, favorecido, item de dispêndio e centro de custo.
- Para movimentos com modalidade de Bordeaux: todos têm que ter a mesma conta fonte e número de liberação de Bordeaux.
- Para movimentos com modalidade de cheque: todos têm que ter a mesma conta fonte, favorecido, item de dispêndio e centro de custo.

BR9-Caso esteja ocorrendo um pagamento via cheque, o sistema deve registrar o cheque o status “pendente de emissão

BR10-Caso a modalidade de pagamento escolhida seja Bordeaux, o campo Número de Liberação de Bordeaux fica disponível, admite números e letras

BR11-Caso exista protocolos relacionados à mesma solicitação, os mesmos devem ser listados agrupados.





# APÊNDICE D – Questionário de Avaliação da Técnica de Leitura Shiô

*Este apêndice apresenta o questionário de avaliação da técnica Shiô,  
utilizado nos estudos realizados.*

## D.1. Questionário de Avaliação da Técnica Shiô

Nome: \_\_\_\_\_

Data:

### Questionário de Avaliação Pós-Experimento

Por favor, responda o questionário abaixo de forma a nos permitir o registro de sua opinião sobre a aplicação da técnica KMN, além dos procedimentos envolvidos na condução do estudo experimental. A sua opinião é muito importante.

1. Como você classificou o grau de dificuldade da aplicação da técnica?  
 Muito Fácil  
 Fácil  
 Médio  
 Difícil  
 Muito Difícil
2. Você seguiu a técnica de leitura completamente?  
 Não. Eu a ignorei completamente.  
 Parcialmente. Eu tentei, mas não segui a técnica na maior parte do tempo.  
 Completamente. Segui a técnica passo a passo.

3. Como a técnica de leitura o auxiliou a identificar defeitos nos requisitos?
- Negativamente. A técnica de leitura atuou como um obstáculo. O meu desempenho teria sido melhor se eu não a tivesse utilizado.
- Neutro. Acho que encontraria os mesmos defeitos caso não tivesse utilizado a técnica de leitura.
- Positivamente. A técnica de leitura me auxiliou na detecção de defeitos. Talvez, não teria detectado alguns defeitos caso não tivesse utilizado a técnica.
4. Você utilizaria a técnica em inspeções futuras?
- Não                       Talvez                       Sim
5. Na sua opinião, quais aspectos da técnica tornam sua aplicação fácil de usar (pontos positivos)?
6. Na sua opinião, quais aspectos da técnica tornam sua aplicação difícil de usar (pontos negativos)?
7. Na sua opinião, como a técnica poderia ser melhorada?



8. Você esteve presente no treinamento?

Sim  Não

9. Como você avalia o treinamento conduzido (responda apenas se esteve presente no treinamento)?

Insuficiente  Neutro  Adequado

10. Na sua opinião, como o treinamento poderia ser melhorado (responda apenas se esteve presente no treinamento)?

11. Você teve dificuldade em entender os modelos?

Sim  Não

12. O domínio dos modelos dificultou no uso da técnica?

Sim  Não

13. Você considera que a complexidade (quantidade, tamanho) dos modelos dificulte muito na utilização da técnica?

Sim  Não

14. Qual dos conjuntos de modelos você considerou mais fácil?

Movimento

Conta

Ambos tem a mesma complexidade