

**UM MODELO PARA APOIAR PROJETISTAS DE
SISTEMAS COLABORATIVOS NA
ANTECIPAÇÃO DE CENÁRIOS**

MANOEL PEREIRA JUNIOR

**UM MODELO PARA APOIAR PROJETISTAS DE
SISTEMAS COLABORATIVOS NA
ANTECIPAÇÃO DE CENÁRIOS**

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

ORIENTADOR: RAQUEL OLIVEIRA PRATES

Belo Horizonte

Junho de 2016

Ficha catalográfica elaborada pela Biblioteca do ICEX - UFMG

Pereira Júnior, Manoel.

P436m Um modelo para apoiar projetistas de sistemas colaborativos na antecipação de cenários / Manoel Pereira Júnior. – Belo Horizonte, 2016.
xxv, 168 f.: il.; 29 cm.

Tese (doutorado) - Universidade Federal de Minas Gerais – Departamento de Ciência da Computação.

Orientadora: Raquel Oliveira Prates.

1. Computação - Teses. 2. Engenharia Semiótica. 3. Interação humano-computador. 4. Modelagem de sistemas colaborativos. I. Orientadora. II. Título.

t

CDU 519.6*75(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Um modelo para apoiar projetistas de sistemas colaborativos na antecipação de cenários

MANOEL PEREIRA JUNIOR

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROFA. RAQUEL OLIVEIRA PRATES - Orientadora
Departamento de Ciência da Computação - UFMG

PROF. EDUARDO MAGNO LAGES FIGUEIREDO
Departamento de Ciência da Computação - UFMG

DR. RENATO FONTOURA DE GUSMÃO CERQUEIRA
IBM Research - Brazil

PROFA. SIMONE DINIZ JUNQUEIRA BARBOSA
Departamento de Informática - PUCRJ

PROF. WAGNER MEIRA JUNIOR
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 02 de junho de 2016.

*Dedico esse trabalho ao meu filho Enzo e à minha filha Manuela, que está por vir!
Espero que sirva como uma semente do conhecimento em suas vidas.*

Agradecimentos

Agradeço imensamente à minha esposa Amanda e ao meu filho Enzo pelos momentos de apoio e por compreender meus momentos de ausência para dedicação a este trabalho.

Aos meus pais e irmã um agradecimento profundo, por terem sempre me incentivado na busca do conhecimento e por sempre acreditarem que eu era capaz.

Agradeço, em especial, à minha orientadora Dra. Raquel O. Prates, que me apoiou incansavelmente, que sempre tinha as palavras certas para os momentos de dificuldade e que me mostrou o “caminho das pedras” da ciência.

Aos colegas do IFMG Bruno F., Everthon V. dos Santos, Paloma M. de Oliveira, Wallace A. Rodrigues e Washington S. Silva que me apoiaram durante essa caminhada.

Aos colegas do Núcleo de Pesquisa em Engenharia Semiótica e Interação (PENSi) pelas críticas e comentários que ajudaram no desenvolvimento desse trabalho, em especial Luiz Damilton e Maria Lúcia B. Villela.

Aos colegas do *Semiotic Engineering Research Group* (SERG) da PUC-Rio, em especial à Clarisse S. de Souza, pelas contribuições que trouxe a este trabalho e também aos colegas Cleiton Slaviero e Marcelle P. da Mota.

Finalmente expresse minha gratidão aos membros da minha banca de defesa de tese - Dra. Simone Diniz Junqueira Barbosa, Dr. Renato Fontoura de Gusmão Cerqueira, Dr. Wagner Meira Júnior e Dr. Eduardo Magno Lages Figueiredo, pelo interesse e disponibilidade.

“A vida sem ciência é uma espécie de morte.”
(Sócrates)

Resumo

A presença de sistemas colaborativos na vida cotidiana moderna tem se tornado cada vez mais frequente. Ao projetar um sistema como esse, os projetistas devem considerar vários aspectos, que incluem os papéis e grupos existentes, as ações disponíveis e a sua consequência no restante do sistema, dentre outros. Todas as decisões em relação às ações e suas consequências devem ser pensadas em tempo de projeto. No entanto, como o sistema é colaborativo, uma ação de um usuário pode ter diferentes impactos no restante do contexto do sistema. Antecipar todas as consequências de todas as ações previstas para o sistema ao longo do tempo pode ser uma tarefa difícil, mesmo para o projetista. Assim, com o objetivo de apoiar o projetista na antecipação de cenários potencialmente problemáticos durante a etapa de projeto de um sistema colaborativo, propomos um modelo denominado SIGMa (*ScenarIo Generator Model*). O SIGMa é composto por uma linguagem de modelagem, denominada SIGMa-dl e por um gerador de cenários, denominado SIGN (*ScenarIo GeNerator*). A SIGMa-dl permite que projetistas modelem a interação em um sistema colaborativo, incluindo as ações dos usuários ao longo do tempo e os seus impactos. Com base nessa descrição, o SIGN gera o conjunto de cenários que potencialmente vão se concretizar em tempo de uso. Desta forma, o projetista pode examiná-los e verificar se algum deles pode representar situações indesejáveis para os usuários finais. Caso ele identifique cenários que não queira que sejam disponibilizados, ele pode então alterar o seu modelo para evitá-los. Para avaliar a expressividade da nossa linguagem de modelagem e a utilidade do SIGN, conduzimos três avaliações iniciais. Na primeira realizamos a análise do SIGMa usando o *framework* Cognitive Dimensions of Notations (CDN) com o objetivo de verificar possíveis problemas e limitações da linguagem. Em seguida, fizemos a reengenharia de dois sistemas colaborativos existentes - o Facebook e o Google Inactive Account Manager. Finalmente, realizamos testes com projetistas de sistemas interativos e potenciais usuários do modelo. Os resultados desses testes forneceram indícios sobre o potencial que o nosso modelo tem de apoiar os projetistas de sistemas colaborativos na antecipação de cenários potencialmente problemáticos.

Palavras-chave: Interação Humano-computador, Engenharia Semiótica, Modelagem de Sistemas Colaborativos, Antecipação da Interação, Impacto ao longo do tempo.

Abstract

The presence of collaborative systems in modern everyday life has become more and more frequent. When designing such system, designers should consider different aspects, including the roles and existing groups, the available actions and their consequences in the system as a whole, among others. All decisions regarding the actions and their consequences should be considered at design time. However, as the system is collaborative, a user's action may cause different impacts within the context of the system. Anticipating all the consequences of all actions planned for the system over time can be a difficult task even for the designer. Thus, in order to support the designer in the anticipation of potentially problematic scenarios during a collaborative system design, we propose a model called SIGMa (*Scenario Generator Model*). SIGMa consists of a modeling language, called SIGMa-dl and a scenario generator, called SIGN (*Scenario GeNerator*). SIGMa-dl allows designers to describe the interaction within a collaborative system, including users' actions over time and their impact. Based on this description, SIGN generates the scenarios that potentially may be created at use time. Thus, the designer can examine the scenarios and check if any of them may represent undesirable situations for end users. If the designer identifies scenarios that he/she thinks that would be undesirable, the designer can change the model to avoid them. To evaluate the expressiveness of SIGMa-dl and the usefulness of SIGN, we conducted three preliminary evaluation studies. In the first evaluation we performed an analysis of the SIGMa using the Cognitive Dimensions of Notations (CDN) *framework* in order to identify possible problems and limitations. Then we used SIGMa-dl to reengineer the model for two existing collaborative systems - the Facebook and Google Inactive Account Manager. Finally, we conducted tests with designers of interactive systems and potential users of the model. The results of these evaluations have provided us with evidence of the potential that our model has to support designers of collaborative systems in anticipation of potentially problematic scenarios.

Keywords: Human Computer-Interaction, Semiotic Engineering, Design of Colabo-

rative Systems, Interaction Anticipation, Impact over time.

Lista de Figuras

| | | |
|------|--|----|
| 1.1 | Ilustração da metodologia utilizada. | 5 |
| 2.1 | Possível relação de amizade no Facebook. | 9 |
| 2.2 | Interface do PrivSim. | 14 |
| 2.3 | Porcentagem de acertos por tarefa usando o Facebook e o PrivSim. | 22 |
| 3.1 | Subelementos componentes do elemento comunicativo fala. | 33 |
| 3.2 | Subelementos componentes do elemento comunicativo conversa. | 34 |
| 3.3 | O detalhamento do elemento comunicativo fala. | 35 |
| 3.4 | A emissão proposta e seus respectivos subelementos comunicativos. | 36 |
| 3.5 | A recepção proposta e seus respectivos subelementos comunicativos. | 36 |
| 3.6 | Ontologia do GTA (adaptado de [Welie et al., 1998]). | 38 |
| 3.7 | Imagem da interface da ferramenta EUTERPE mostrando um modelo de exemplo. | 39 |
| 3.8 | Representações gráficas dos quatro tipos de tarefas no CTT. | 41 |
| 3.9 | Representações gráficas dos operadores temporais binários do CTT. | 42 |
| 3.10 | Representações gráficas dos operadores temporais unários do CTT. | 42 |
| 3.11 | Exemplo de um modelo CTT para a operação de retirada de dinheiro em um caixa eletrônico. | 43 |
| 3.12 | Fragmento da interface do CTTE mostrando o simulador em funcionamento. | 43 |
| 4.1 | Modelo proposto. | 50 |
| 4.2 | Cenários que seriam gerados pelo SIGN para as duas relações de exemplo mostradas na Figura 4.24. | 66 |
| 4.3 | Exemplos de relações e suas consequências. | 68 |
| 4.4 | Grafo resultante das relações de exemplo definidas na Figura 4.3. | 69 |
| 4.5 | Imagem da tela do protótipo implementado, mostrando as duas principais áreas de interação. | 70 |

| | | |
|------|---|----|
| 4.6 | Abas que permitem a edição de cada dimensão prevista e a caracterização de cada espaço de tempo definido. | 71 |
| 4.7 | Estrutura de funcionamento do protótipo. | 71 |
| 4.8 | Aba de caracterização dos espaços de tempo. | 72 |
| 4.9 | Imagem da tela do protótipo que permite a exploração livre dos cenários gerados destacando a visualização gráfica correspondente ao cenário explorado. | 73 |
| 4.10 | Ícones que caracterizam cada tipo de mudança. | 74 |
| 4.11 | Fragmento da interface do SIGN mostrando a evolução de um cenário ao longo de dois tempos distintos. | 75 |
| 4.12 | Imagem da tela do protótipo que permite a exploração livre dos cenários gerados destacando a listagem de ações possíveis subsequentes a partir de uma ação selecionada na árvore de cenários. | 75 |

Lista de Tabelas

| | | |
|------|--|----|
| 2.1 | Resumo das tarefas e das perguntas realizadas. | 16 |
| 2.2 | Resultados da Interação com o Facebook. | 18 |
| 2.3 | Resultados da Interação com o PrivSim. | 20 |
| 4.1 | Sintaxe de definição da dimensão Espaços de tempo. | 52 |
| 4.2 | Exemplos de definição de instância da dimensão Espaços de tempo. | 52 |
| 4.3 | Sintaxe de definição da dimensão Papéis. | 53 |
| 4.4 | Exemplos de definição de instâncias da dimensão Papéis. | 53 |
| 4.5 | Sintaxe de definição da dimensão Grupos. | 54 |
| 4.6 | Exemplos de definição de instâncias da dimensão Grupos. | 54 |
| 4.7 | Sintaxe de definição da dimensão Artefatos Atômicos. | 55 |
| 4.8 | Exemplos de definição de instâncias da dimensão Artefatos Atômicos. | 55 |
| 4.9 | Sintaxe de definição da dimensão Artefatos Compostos. | 55 |
| 4.10 | Exemplo de definição de uma instância da dimensão Artefatos Compostos. | 56 |
| 4.11 | Sintaxe de definição da dimensão Posses. | 56 |
| 4.12 | Exemplo de definição de uma instância da dimensão Posses. | 57 |
| 4.13 | Sintaxe de definição da dimensão Ações. | 57 |
| 4.14 | Exemplos de definição de instâncias da dimensão Ações. | 57 |
| 4.15 | Sintaxe de definição da dimensão Mudanças no conjunto de ações. | 59 |
| 4.16 | Exemplos de definição de instâncias da dimensão Mudanças no conjunto de ações. | 59 |
| 4.17 | Sintaxe de definição da dimensão Mudanças nos espaços de tempo. | 60 |
| 4.18 | Exemplos de definição de instâncias da dimensão Mudanças nos espaços de tempo. | 60 |
| 4.19 | Sintaxe de definição da dimensão Mudanças de posse. | 61 |
| 4.20 | Exemplo de definição de uma instância da dimensão Mudanças de posse. | 61 |
| 4.21 | Sintaxe de definição da dimensão Mudanças de papel/grupo. | 62 |

| | | |
|------|--|----|
| 4.22 | Exemplo de definição de uma instância da dimensão Mudanças de papel/grupo em um ambiente virtual de aprendizagem. | 62 |
| 4.23 | Sintaxe de definição da dimensão Relações. | 64 |
| 4.24 | Exemplos de definição de instâncias da dimensão Relações. As duas relações de exemplo não possuem condições, por isso este campo foi definido para fins ilustrativos como “Nula” | 65 |
| 4.25 | Exemplo de como caracterizar as instâncias válidas para cada dimensão em um espaço de tempo. | 67 |
| 5.1 | Principais aspectos da modelagem do Facebook usando a SIGMa-dl. | 89 |
| 5.2 | Principais aspectos da modelagem do Google IAM usando a SIGMa-dl. | 92 |

Lista de Abreviaturas

CSCW Computer Supported Cooperative Work

CTT Concurrent Task Trees

CTTE Concurrent Task Trees Environment

GTA Groupware Task Analysis

HAMSTERS Human-centered Assessment and Modeling to Support Task Engineering for Resilient Systems

IHC Interação Humano-Computador

MAC Método de Avaliação da Comunicabilidade

MIS Método de Inspeção Semiótica

SIGMa-dl ScenarIo Generator Model design language

SIGMa ScenarIo Generator Model

SIGN ScenarIo GeNerator

Sumário

| | |
|---|-----------|
| Agradecimentos | ix |
| Resumo | xiii |
| Abstract | xv |
| Lista de Figuras | xvii |
| Lista de Tabelas | xix |
| 1 Introdução | 1 |
| 1.1 Objetivos | 3 |
| 1.2 Metodologia | 4 |
| 1.3 Principais Resultados e Contribuições | 6 |
| 1.4 Organização da Tese | 6 |
| 2 O problema | 9 |
| 2.1 Estudo Exploratório - Facebook | 11 |
| 2.2 Considerações Finais | 23 |
| 3 Fundamentação Teórica e Trabalhos Relacionados | 25 |
| 3.1 Teoria da Engenharia Semiótica | 25 |
| 3.2 Modelos para Projetos de Sistemas Colaborativos | 28 |
| 3.2.1 Modelos fundamentados na Engenharia Semiótica | 28 |
| 3.2.2 Modelos gerais | 37 |
| 3.2.3 Reflexões sobre os modelos existentes | 45 |
| 4 SIGMa: Modelo proposto | 49 |
| 4.1 Linguagem de Modelagem Proposta - <i>SIGMa-dl</i> | 51 |
| 4.1.1 Espaços de tempo | 51 |

| | | |
|----------|---|------------|
| 4.1.2 | Papéis | 53 |
| 4.1.3 | Grupos | 53 |
| 4.1.4 | Artefatos | 54 |
| 4.1.5 | Posses | 56 |
| 4.1.6 | Ações | 56 |
| 4.1.7 | Mudanças | 58 |
| 4.1.8 | Relações | 62 |
| 4.1.9 | Caracterização dos Espaços de tempo | 65 |
| 4.2 | <i>SIGN - ScenarIo GeNerator</i> | 67 |
| 4.3 | Protótipo implementado | 69 |
| 4.4 | Considerações Finais | 76 |
| 5 | Avaliação | 77 |
| 5.1 | Avaliação usando o <i>framework</i> Cognitive Dimensions of Notations . . . | 78 |
| 5.1.1 | O <i>framework</i> <i>Cognitive Dimensions of Notations</i> | 78 |
| 5.1.2 | Resultados | 81 |
| 5.2 | Reengenharia de sistemas existentes | 87 |
| 5.2.1 | Resultados | 89 |
| 5.3 | Avaliação com Projetistas Reais | 95 |
| 5.3.1 | Resultados | 99 |
| 5.4 | Revisitando as Questões de Avaliação 1 e 2 | 111 |
| 6 | Conclusões | 113 |
| 6.1 | Constatações e limites do SIGMa | 115 |
| 6.2 | Considerações sobre as avaliações realizadas | 117 |
| 6.3 | Contribuições do trabalho | 118 |
| 6.4 | Trabalhos Futuros | 120 |
| | Referências Bibliográficas | 123 |
| | Apêndice A Termo de Consentimento Livre e Esclarecido | 127 |
| | Apêndice B Questionário Pré-teste | 131 |
| | Apêndice C Apresentação do SIGMa para os projetistas | 135 |
| | Apêndice D Material do teste de inspeção de um modelo pronto | 147 |
| | Apêndice E Material do teste de modelagem de um sistema | 153 |

| | | |
|------------|------------------------|-----|
| Apêndice F | Questionário pós-teste | 159 |
| Apêndice G | Roteiro do grupo focal | 167 |

Capítulo 1

Introdução

O uso de sistemas computacionais colaborativos tem permitido a interação de pessoas para os mais diversos fins, como envio de mensagens, realização de trabalhos em grupo, discussão de temas por comunidades virtuais e jogos online. O desenvolvimento de aplicações deste tipo (i.e., multiusuário) é intrinsecamente mais complexo que o desenvolvimento de aplicações monousuário pois, como envolve a interação entre pessoas, deve apoiar a comunicação, colaboração e coordenação entre elas [Grudin & Poltrock, 2013].

Além disso, um sistema multiusuário deve considerar ainda os aspectos sociais envolvidos nas relações humanas. Isso foi relatado no trabalho de Ackerman [2000], que já no ano de 2000 previa a dificuldade em conciliar requisitos técnicos com os requisitos sociais envolvidos em sistemas nos quais usuários interagem entre si. Segundo ele, reduzir a distância entre estes dois lados é um importante desafio das áreas de Sistemas Colaborativos e Interação Humano-Computador (IHC). Para ilustrar o problema, Ackerman [2000] mostra um caso real do projeto *Platform for Privacy Preferences Project (P3P)* do *World Wide Web Consortium*. Este projeto tinha como objetivo criar um padrão de privacidade para a Web. A ideia era que o usuário estabelecesse alguns critérios de privacidade em seu computador local e que cada página da Web teria uma política de privacidade. Se a política estabelecida pelo usuário estivesse de acordo com a da página Web, criaria-se um “acordo P3P” e então seria estabelecida uma transferência de dados. Isso foi pensado para que o usuário não tivesse que ler a política de privacidade de cada site que visita. Essa parte da versão 1 do P3P foi arquivada, pois não havia ninguém confiante que funcionaria bem. O problema é que, em sociedade, as pessoas ajustam seus parâmetros de privacidade, dependendo da situação atual. Em um exemplo, o autor mostra que um professor pode decidir não passar seu número de telefone para os alunos, mas só para seus colegas de trabalho. No entanto, em al-

gum momento ele pode querer passar esse número para um aluno que se destacou no curso. O mesmo não acontece em um sistema, no qual um usuário deve explicitamente selecionar ou não cada aspecto da privacidade desejado. Segundo o autor, o espaço de informação a ser manipulado neste caso é potencialmente infinito, inviabilizando assim que ele seja traduzido em regras que seriam necessárias para o funcionamento do sistema. Assim, ele conclui que não há mecanismos técnicos para representar as atividades da vida social cotidiana.

Frequentemente, projetistas de sistemas colaborativos optam por oferecer possibilidades de configuração para possibilitar aos usuários lidarem com as diferenças geradas em cada contexto. Assim, o usuário pode gerar mudanças de comportamento do sistema para diferentes situações. No entanto, como o sistema envolve diversos usuários, pode ser difícil para os usuários compreenderem como as suas configurações impactam outros usuários. Por exemplo, ao postar uma foto no Facebook, o usuário posta uma foto apenas para seus amigos, mas (potencialmente) ela poderá vir a ser acessada por pessoas que não fazem parte desse grupo. E nem todos os usuários percebem que isso pode acontecer, como mostraremos no Capítulo 2. Antecipar os impactos de suas ações é um problema que pode estar presente em um sistema colaborativo no qual as ações de um usuário geram consequências para outros. Na busca de melhorar a percepção dos usuários sobre as consequências de suas configurações e ações, de Souza et al. [2010] levantam a necessidade de que os usuários simulem os processos comunicativos e sociais envolvidos em sistemas colaborativos, para que compreendam como suas decisões influenciam os processos e outros usuários em diferentes contextos. Em um trabalho mais recente, esse problema foi caracterizado como a necessidade da **“antecipação da interação”** [Prates et al., 2015]. As autoras argumentam que sistemas colaborativos devem oferecer aos usuários os meios para antecipar as consequências de suas escolhas ou ações ao longo do tempo, e que os projetistas de sistemas colaborativos devem abordar as questões envolvidas durante a etapa de projeto de sistemas que suportam a antecipação. Apesar de estes trabalhos abordarem o potencial da antecipação para os usuários finais, as combinações possíveis quando se oferece várias decisões em tempo de uso podem ser muitas e seria importante que o próprio projetista pudesse antecipar os cenários que permitem que o usuário experimente em tempo de uso.

Esta é a principal motivação deste trabalho - a falta de um ferramental de simulação que possa apoiar a antecipação, em tempo de projeto, de cenários que podem ser criados em tempo de uso. Embora existam vários modelos que tratam da modelagem de sistemas colaborativos, como [Paternò et al., 2001; Veer et al., 1996; Martinie et al., 2011] e outros, não encontramos nenhum que permita que o projetista anteveja possíveis cenários criados ao longo do tempo que estarão disponíveis ao usuário após

a entrega do sistema (sem que ele tenha que descrever todos os caminhos de interação possíveis), e que possa assim definir quais deles são ou não desejáveis. Portanto, com esta motivação, este trabalho pretende ser um primeiro passo no apoio aos projetistas de sistemas colaborativos no projeto de ações e configurações a serem oferecidos aos usuários através do sistema e seus impactos.

Descrevemos o objetivo da nossa pesquisa na próxima seção. Na seção 1.2, descrevemos a metodologia que aplicamos a fim de elaborar e investigar as questões de pesquisa envolvidas neste trabalho. Na seção 1.3 apresentamos as contribuições desse trabalho de doutorado e na seção 1.4 apresentamos a estrutura da tese.

1.1 Objetivos

O desenvolvimento de aplicações multiusuário é intrinsecamente mais complexo que o desenvolvimento de aplicações monousuário. Assim, em busca de apoiar a atividade de projeto de sistemas colaborativos, o objetivo geral dessa tese de doutorado consiste em apoiar os projetistas de sistemas colaborativos a descrever, em tempo de projeto, as escolhas que os usuários podem fazer em tempo de uso, bem como antecipar os cenários que estão permitindo aos usuários criar através da combinação dessas escolhas. Ao analisar os diferentes cenários que poderiam ser criados pelos usuários, projetistas podem avaliar se algum deles poderia conduzir a situações indesejáveis e, em caso afirmativo, decidir como lidar com eles durante o projeto conceitual do sistema (ao invés de verificar este problema após o sistema ter sido construído ou após estar em uso).

Para atingir este objetivo, propomos um modelo, denominado SIGMa (*ScenarIo Generator Model*) que é composto por dois componentes principais: uma linguagem de modelagem, denominada SIGMa-dl (*ScenarIo Generator Model design language*) e um gerador de cenários, denominado SIGN (*ScenarIo GeNerator*). A **SIGMa-dl** permite que o projetista desenvolva um modelo para um sistema colaborativo no qual descreve as ações que usuários podem fazer ao longo do tempo e os impactos que estas geram. Enquanto desenvolve o modelo, o projetista pode submetê-lo ao **SIGN**, que foi inspirado pelo trabalho de Prates [1998]. Já naquela época a autora previu as vantagens da construção de um gerador de cenários. No entanto, naquele trabalho, embora um gerador tenha sido previsto, foi definido apenas na arquitetura como um componente a ser acrescentado, mas não foi especificado de forma detalhada.

As dimensões contempladas pela SIGMa-dl referem-se a dois aspectos do sistema de colaboração: sua estrutura (papéis, grupos, artefatos e a quem pertencem) e seu

comportamento (definição dos tempos e das ações que podem ser executadas por cada papel ou grupo e suas consequências). Permitem ainda a modelagem das ações possíveis aos papéis/grupos e as mudanças que essas geram na aplicação como um todo. Já o gerador de cenários permite que, a partir da descrição das ações e seus impactos, os diversos cenários que poderão vir a ser gerados sejam antecipados. Assim o projetista pode experimentar os diversos caminhos de interação que pretende disponibilizar aos futuros usuários, mas em tempo de projeto, sem se preocupar com qualquer detalhe de implementação. Durante a simulação desses cenários, ele pode decidir quais deles são ou não desejáveis, apoiando assim suas decisões em relação ao que será oferecido no sistema final. Mais detalhes sobre a SIGMa-dl e o SIGN podem ser vistos no Capítulo 4.

Como é fundamentado na teoria da Engenharia Semiótica [de Souza, 2005], apresentada na Seção 3.1, o SIGN se enquadra na categoria das ferramentas epistêmicas, pois permite que o projetista reflita sobre as implicações de suas decisões no sistema final, mas tudo isso antes do sistema em si ser concebido, ou seja em tempo de projeto. O SIGN tem caráter descritivo, ou seja, tem como objetivo permitir que o projetista identifique problemas nos cenários gerados, mas nunca de fazer um julgamento sobre a sua validade. Então caberá ao projetista decidir se os potenciais problemas levantados fazem sentido, ou não, no contexto do sistema sendo projetado.

1.2 Metodologia

Como primeiro passo dessa pesquisa, efetuamos estudos exploratórios e teóricos com o objetivo de explorar o problema em si e melhor definir a questão de pesquisa.

Analisamos os diversos modelos existentes para a modelagem de sistemas colaborativos, como será visto no Capítulo 3, verificando suas características e particularidades. Neste ponto, pudemos perceber a falta de trabalhos que lidem com a antecipação de cenários ao longo do tempo. Após o estudo destes modelos, verificamos que para apoiar o projetista na geração de cenários a serem criados em tempo de uso seria necessário oferecer ao projetista uma linguagem de modelagem nova, que permitisse a modelagem das consequências de cada ação ao longo do tempo.

No nosso estudo exploratório, investigamos sistematicamente o Facebook¹, uma vez que identificamos como decisões dos usuários poderiam gerar diferentes cenários no contexto de visualização de fotos. Desta forma, este estudo permitiu não apenas que analisássemos o problema do ponto de vista do usuário final, como levantássemos as

¹<http://www.facebook.com>

características e conceitos que deveriam estar presentes na linguagem de modelagem que iríamos desenvolver. Além disso, como subproduto desse estudo, pudemos comprovar o benefício que a antecipação de cenários pode trazer para os usuários dessa rede social [Pereira Junior et al., 2014]. Nossa revisão da literatura também fundamentou a definição de quais dimensões seriam contempladas pela linguagem de modelagem.

Após definirmos o conjunto de características que deveriam ser contempladas pela linguagem de modelagem proposta neste trabalho, fizemos a sua definição formal, e a nomeamos SIGMa-dl. Após refinarmos a SIGMa-dl, desenvolvemos o algoritmo de geração de cenários, descrito em detalhes na seção 4.2. Finalmente, desenvolvemos um protótipo que permite a modelagem de um sistema colaborativo, usando a SIGMa-dl e que é capaz de gerar os cenários correspondentes à modelagem feita, materializando assim o SIGN.

Após o desenvolvimento do protótipo, o próximo passo foi a condução de avaliações do modelo proposto, com o objetivo de coletar indicadores qualitativos sobre a expressividade da SIGMa-dl e sobre a utilidade e benefício do SIGN. Realizamos inicialmente uma avaliação do SIGMa usando o *framework Cognitive Dimensions of Notations* (CDN)[Green, 1989]. Em seguida definimos as questões de avaliação que investigaríamos e conduzimos duas avaliações: (1) utilizar a SIGMa-dl para fazer a reengenharia de sistemas colaborativos existentes e (2) executar testes com potenciais projetistas de sistemas colaborativos. A Figura 1.1 ilustra a metodologia desse trabalho.



Figura 1.1. Ilustração da metodologia utilizada.

A avaliação que levou em conta a reengenharia de sistemas colaborativos existentes teve como objetivo verificar se a SIGMa-dl tinha expressividade suficiente para permitir a modelagem de sistemas existentes, onde as decisões de projeto já foram tomadas pelo projetista e logo representam decisões reais e que precisariam ser representadas. Além disso, pudemos observar o comportamento do SIGN na geração dos cenários, verificando se ele era capaz de gerar cenários que de fato existem nos sistemas

reais. Já na avaliação que envolveu projetistas reais, também tínhamos o objetivo de colher indicadores qualitativos sobre a expressividade da SIGMa-dl e o apoio oferecido pelos cenários gerados pelo SIGN, porém levando em consideração a percepção de projetistas e potenciais usuários do SIGMa.

1.3 Principais Resultados e Contribuições

Os resultados das avaliações conduzidas geraram indicadores positivos sobre a expressividade da SIGMa-dl e da utilidade da antecipação de cenários possibilitada pelo SIGN, como será visto no capítulo 5. Assim este trabalho é um primeiro passo na direção de apoiar projetistas de sistemas colaborativos na antecipação de cenários ao longo do tempo.

Nossa proposta contribui trazendo para o projetista de sistemas colaborativos uma nova maneira de refletir sobre a modelagem que está realizando. Os modelos propostos até então não focam na antecipação dos cenários que estarão presentes na aplicação final que será entregue aos usuários. Ao usar o SIGMa, o projetista pode modelar sua aplicação colaborativa e pode antever, em tempo de projeto, os cenários que serão gerados a partir da modelagem feita ao longo dos diversos contextos diferentes de execução pelos quais um sistema colaborativo pode passar. Os resultados das nossas avaliações mostram o potencial que o SIGMa tem em permitir essa nova forma de pensar no desenvolvimento de sistemas colaborativos. Ao criarmos o SIGMa, contribuimos diretamente para as áreas de Sistemas Colaborativos e IHC.

Finalmente, essa tese contribui diretamente para a Engenharia Semiótica, teoria de IHC, acrescentando mais uma ferramenta epistêmica ao rol de ferramentas baseadas nessa teoria. Os resultados de nossas avaliações com projetistas reais nos permitiram observar indicadores de que o SIGMa é capaz de fomentar novas ideias na mente do projetista, auxiliando-o assim a refletir sobre os impactos que suas decisões de projeto trazem ao sistema como um todo.

1.4 Organização da Tese

Neste capítulo inicial discutimos a dificuldade em antecipar os cenários de uma aplicação colaborativa. Este problema está presente em qualquer sistema que envolve mais de um usuário e onde as ações de um podem ter consequências para outros. O objetivo desta tese de doutorado é criar uma ferramenta epistêmica que permita ao projetista antecipar os cenários da aplicação colaborativa que está modelando e que o apoie na

reflexão sobre o impacto que as decisões que está oferecendo aos usuários tem sobre os demais, e os cenários que poderão vir a ser criados a partir da combinação destas decisões.

O problema tratado neste trabalho de doutorado é descrito em detalhes no Capítulo 2. Além de descrever o problema, este capítulo apresenta os estudos exploratórios que foram conduzidos até a concepção do modelo proposto.

No Capítulo 3 apresentamos a fundamentação teórica necessária para o entendimento dessa tese, bem como os trabalhos que de alguma forma se relacionam nossa proposta e explicamos as suas limitações e diferenças em relação à nossa proposta.

O SIGMa, por sua vez, é apresentado no Capítulo 4. São descritas em detalhes as dimensões da SIGMa-dl e o SIGN, que permite a geração e interação com os cenários consistentes com a modelagem feita pelo projetista.

Para verificar a expressividade da linguagem proposta e a utilidade do SIGN, realizamos uma análise do SIGMa sob a ótica do *framework Cognitive Dimensions of Notations* (CDN)[Green, 1989]. Realizamos ainda a reengenharia do Facebook e finalmente uma avaliação com projetistas reais de sistemas colaborativos. Nesta última avaliação colhemos as observações dos participantes sobre a linguagem de modelagem proposta e sobre a utilidade do SIGN. Os testes e os resultados obtidos são descritos no Capítulo 5.

No Capítulo 6, apresentamos as conclusões do nosso trabalho, discutimos as diferenças da nossa proposta em relação às existentes na literatura e fazemos uma reflexão sobre os limites do SIGMa, apontando assim os trabalhos futuros. Finalmente, apresentamos as contribuições desta tese.

Capítulo 2

O problema

Neste trabalho estamos investigando o uso de simulação como uma forma apoiar o projetista na antecipação dos potenciais impactos que podem advir das diversas ações disponíveis aos usuários. Para melhorar o entendimento do problema que estamos tratando, tomemos como exemplo o Facebook¹ e uma relação de amizade comum, como a mostrada na Figura 2.1.

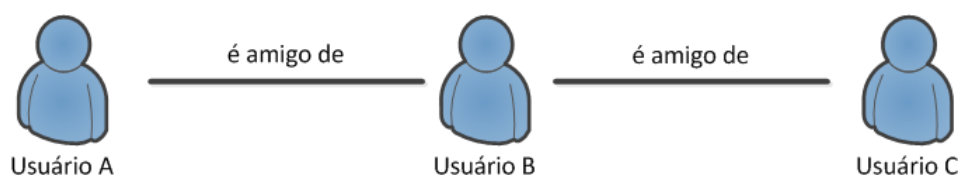


Figura 2.1. Possível relação de amizade no Facebook.

O **usuário A** pode fazer diversos tipos de postagens (mensagens, fotos, vídeos, comentários, etc.) e os outros usuários podem interagir com elas, por exemplo dizendo se gostaram ou não. A visibilidade (i.e. quais usuários podem ver) da postagem, e consequentemente a interação com ela, dependerá da configuração feita pelo **usuário A**. Ele pode, por exemplo, postar uma foto apenas para seus amigos. Neste caso, apenas o **usuário B** (além de **A**) teria acesso à foto. O **usuário C** não poderia ver esta foto, pois não faz parte dos amigos do **usuário A**. No entanto, existe uma funcionalidade no Facebook que permite que usuários marquem outros em fotos. Assim, tendo acesso à foto, o **usuário B** pode marcar outras pessoas na foto postada pelo **usuário A**, inclusive o **usuário C**. Se ele o fizer, o **usuário C** passará a ter acesso à foto, mesmo não tendo sido “autorizado” pelo **usuário A**, que é o dono da foto. O

¹O exemplo mostrado aqui considera o funcionamento real do Facebook, de acordo com a análise detalhada feita na época [Pereira Junior et al., 2014].

usuário A apenas é informado sobre a marcação e a ele não é dada a oportunidade de permitir ou não a marcação de um “amigo de amigo” em sua foto. O **usuário A** tem apenas a opção de excluir a foto, mas o **usuário C** já pode tê-la visto antes que isso aconteça. Neste caso, problemas de ordem social podem ocorrer, como violação de privacidade (já que pessoas indesejadas podem ter acesso à foto), constrangimento (a foto pode comprometer o **usuário A** com o **usuário C**) e problemas na relação pessoal do **usuário A** com o **usuário B** (já que este último foi o responsável por causar a violação de privacidade).

O trabalho de de Souza et al. [2010] traz exemplos de como é difícil para o usuário prever as consequências de seus atos e/ou configurações dentro de sistemas colaborativos reais. Em um dos exemplos os autores avaliaram a configuração de um grupo do *Google Groups* pelo seu proprietário. Na funcionalidade que permite o convite de membros para o grupo, o proprietário simplesmente não pode visualizar (ou simular) a mensagem de convite antes de enviá-la. E esta mensagem enviada pelo sistema contém informações que podem não ser desejáveis pelo proprietário, como um trecho que diz que o convidado pode estar sendo vítima de “abuso eletrônico”. Como não tem acesso à interface dos convidados para o grupo, o proprietário pode nunca saber que isso fazia parte da mensagem enviada. Sem a opção de antever os impactos de suas configurações, os usuários têm que pensar em estratégias dispendiosas para testar os efeitos dos parâmetros de configuração que ele estabeleceu (e.g. criar mais de uma conta ou pedir licença para ver como outros usuários têm acesso às informações de um grupo). Além deste exemplo, os autores mostram ainda um relato de um grupo que foi fechado pelo seu criador após identificar que seu grupo não era restrito às pessoas que efetivamente tinham um usuário e senha para acessá-lo, como inicialmente acreditava. Como já dissemos no capítulo anterior, os autores argumentam que, para que usuários possam compreender como suas configurações influenciam os processos sociais de um sistema colaborativo, é necessário a criação de modelos computacionais capazes de simular os processos sociais intrínsecos à ele.

É importante salientar que os exemplos do problema acima são vivenciados pelos usuários. A combinação de configurações disponibilizadas pelos projetistas permitiu que esses cenários fossem criados em tempo de uso. Dependendo das configurações e possibilidades de combinação pode haver um número infinito de cenários que podem ser gerados. Por exemplo, no caso do Facebook, se cada amigo marcasse outras pessoas, estas por sua vez poderiam então marcar novas pessoas e assim por diante. Para o projetista pode ser difícil antever todos os cenários que poderão vir a ser criados em determinadas situações a partir das decisões de usuários, principalmente se este número é alto. Assim, o ideal seria que o projetista tivesse apoio para antecipar que

cenários seriam esses e mesmo analisar se todos fariam sentido ou seriam desejáveis para o usuário. Na nossa pesquisa da literatura, não encontramos nenhum trabalho que permite ao projetista gerar os cenários que pretende disponibilizar aos usuários finais e simulá-los ao longo do tempo, sem ter que necessariamente descrever todos os cenários possíveis.

Assim, nosso modelo tem o objetivo de apoiar o projetista na antecipação dos cenários que pretende disponibilizar aos usuários finais, mas sem que ele tenha que pensar em todos os cenários possíveis. Ao usar o modelo que estamos propondo, o projetista caracteriza apenas as consequências que uma ação individual traz para o sistema. O nosso gerador de cenários - o SIGN - se encarrega de “costurar” todas as consequências de todas as ações, permitindo gerar assim todos os cenários possíveis de acordo com o modelo fornecido. Assim, a principal característica do SIGMA é que ele permite ao projetista pensar nos impactos das ações que está modelando ao longo do tempo, como será mostrado no capítulo 4.

Como primeiro passo deste trabalho de doutorado, identificamos as características que deveriam ser contempladas pela linguagem de modelagem que compõe o nosso modelo. Nossa primeira questão de pesquisa foi: *“Quais características uma linguagem deve permitir ao projetista representar para que ele consiga modelar um sistema colaborativo?”*.

Para responder esta pergunta investigamos “o lado do usuário” em uma aplicação existente - o Facebook. O objetivo desta análise era identificar características que estavam representadas no sistema e cenários reais que poderiam ser gerados. A linguagem deveria permitir que este sistema fosse descrito de forma que o gerador de cenários pudesse apresentar estes cenários ao projetista. Além de investigar o Facebook, analisamos a literatura atual sobre modelagem de sistemas colaborativos para que pudessemos identificar as características, que daqui pra frente denominamos *dimensões*, que deveriam ser consideradas pela linguagem proposta neste trabalho.

Mais detalhes sobre o estudo realizado sobre o Facebook são mostrados na próxima seção.

2.1 Estudo Exploratório - Facebook

As redes sociais online estão cada vez mais presentes no cotidiano das pessoas. Elas já são as aplicações na web mais populares entre os usuários. As pessoas se conectam a estas redes com os mais diversos fins, como por exemplo encontrar amigos e acompanhar o que estão postando [Rauber et al., 2011; Reynolds et al., 2011]. O Facebook é um

bom exemplo deste crescimento. Atualmente conta com 1,65 bilhões de usuários ativos mensalmente². No entanto, o aumento da popularidade das redes sociais tem despertado inquietações diversas entre seus usuários, sendo uma das grandes preocupações a privacidade [Gill et al., 2011].

O primeiro passo deste estudo de caso [Pereira Junior et al., 2012, 2013a] foi fazer uma avaliação do próprio Facebook, com o objetivo de conhecer maiores detalhes sobre o perfil dos usuários e sua experiência de uso. Para atingir este objetivo, inicialmente disponibilizamos um questionário online entre abril e maio de 2012³. Além da seção que apresentava perguntas que visavam identificar o perfil do usuário e sua experiência com o Facebook, havia também uma seção com questões específicas sobre a privacidade vivenciada dentro desta rede social. Analisando os 166 questionários válidos que recebemos, constatamos que as funcionalidades relativas a fotos eram as que mais preocupavam os usuários, em termos de privacidade.

Em seguida, realizamos a inspeção da interface do Facebook utilizando o Método de Inspeção Semiótica (MIS) [de Souza et al., 2006]. Como a análise dos questionários indicou que os usuários se preocupam principalmente com fotos, o escopo da inspeção foi reduzido às funcionalidades relacionadas a elas, incluindo postagem, marcação e compartilhamento. A inspeção foi feita por 3 alunos de doutorado, incluindo o autor desta tese, de forma independente, e em seguida as rupturas identificadas foram consolidadas em um único documento final. Após a consolidação, pudemos constatar que o Facebook apresenta rupturas importantes, que podem levar o usuário a usar o sistema de forma indesejada. Uma das principais rupturas relativas à privacidade identificada diz respeito à marcação de pessoas em fotos. Se um usuário posta uma foto **apenas** para seus amigos, poderia se esperar que ninguém fora deste grupo deveria ter acesso a ela. No entanto, se um desses amigos marcar um terceiro na foto (que não é amigo do postador), a foto passa a ser acessível a este terceiro usuário marcado, mesmo este não fazendo parte do grupo inicialmente definido pelo postador. Utilizamos este mesmo exemplo para retratar uma instância do problema maior que estamos abordando nessa tese, descrito no início deste capítulo.

Após a inspeção, realizamos testes com usuários reais do Facebook, utilizando o Método de Avaliação da Comunicabilidade (MAC) [de Souza & Leitão, 2009]. Nosso objetivo era verificar se os usuários realmente sofreriam os efeitos das rupturas que foram identificadas durante a inspeção. Para o teste, foram elaboradas cinco tarefas que

²<http://newsroom.fb.com/Key-Facts>

³Vale ressaltar que os dados colhidos em todas as etapas deste estudo de caso dizem respeito à interface do Facebook da época (abril e maio de 2012). Alguns recursos já foram modificados desde então.

previam a postagem, marcação e compartilhamento de fotos. O teste foi conduzido em ambiente controlado, composto por uma sala de observação e uma sala de testes, ambas com os devidos equipamentos de captura de áudio e vídeo, além de um software de gravação da interação dos usuários. Esta parte do estudo também foi desenvolvida pelos mesmos avaliadores que aplicaram o MIS. Todas as questões éticas foram observadas e todos os usuários testados assinaram um termo de consentimento de participação.

Para preservar a identidade e as contas reais dos usuários, criamos contas fictícias para o teste com nomes e fotos de personagens da literatura infantil brasileira, conhecidos dos usuários. O teste contou com cinco participantes que usavam o Facebook há pelo menos um ano e meio. Inicialmente conduzimos uma entrevista pré-teste, para identificar o perfil dos participantes. Em seguida, cada um deles realizou as cinco tarefas previstas. Ao término, conduzimos uma entrevista pós-teste, com o intuito de identificar as dificuldades encontradas durante a realização das tarefas. Obtidos os dados, prosseguimos para as etapas de etiquetagem, interpretação e geração do perfil semiótico, como previsto pelo MAC. Em geral os resultados mostraram que os usuários conseguiram executar as tarefas, mas quando questionados sobre os seus efeitos (e.g. quem teria acesso à foto depois de um compartilhamento) em uma entrevista pós-teste, todos tiveram dúvidas na resposta e a maioria não soube responder ou respondeu incorretamente.

Dessa forma, conseguimos verificar que os usuários reais participantes do teste realmente experimentaram as rupturas que identificamos durante a inspeção. Como dissemos no início desta seção, nosso objetivo até então era verificar o entendimento dos usuários sobre os mecanismos de configuração de privacidade do Facebook. Nossos resultados mostraram que os usuários não entendem os efeitos das configurações ou ações que fazem. Esta descoberta motivou a execução da segunda parte deste estudo de caso: avaliar se um simulador poderia melhorar o entendimento dos usuários neste caso específico.

Com base nos estudos realizados anteriormente, e com o objetivo de levantar indícios sobre a utilidade de uma ferramenta que permitisse a simulação de cenários diversos, construímos um protótipo funcional, denominado PrivSim. Seu objetivo era permitir que fosse possível simular, a partir das configurações de privacidade, quem de fato teria visibilidade da informação. No caso, o PrivSim leva em consideração não apenas a configuração feita, mas também como as ações disponíveis sobre a informação podem (ou não) alterar esta visibilidade. Assim como nas etapas anteriores (inspeção da interface e testes com usuários), limitamos o escopo de ação do PrivSim às funcionalidades relativas às fotos. Vale ressaltar que o protótipo apresenta uma visão simplificada (árvore) da rede de amizades real (grafo) presente no Facebook.

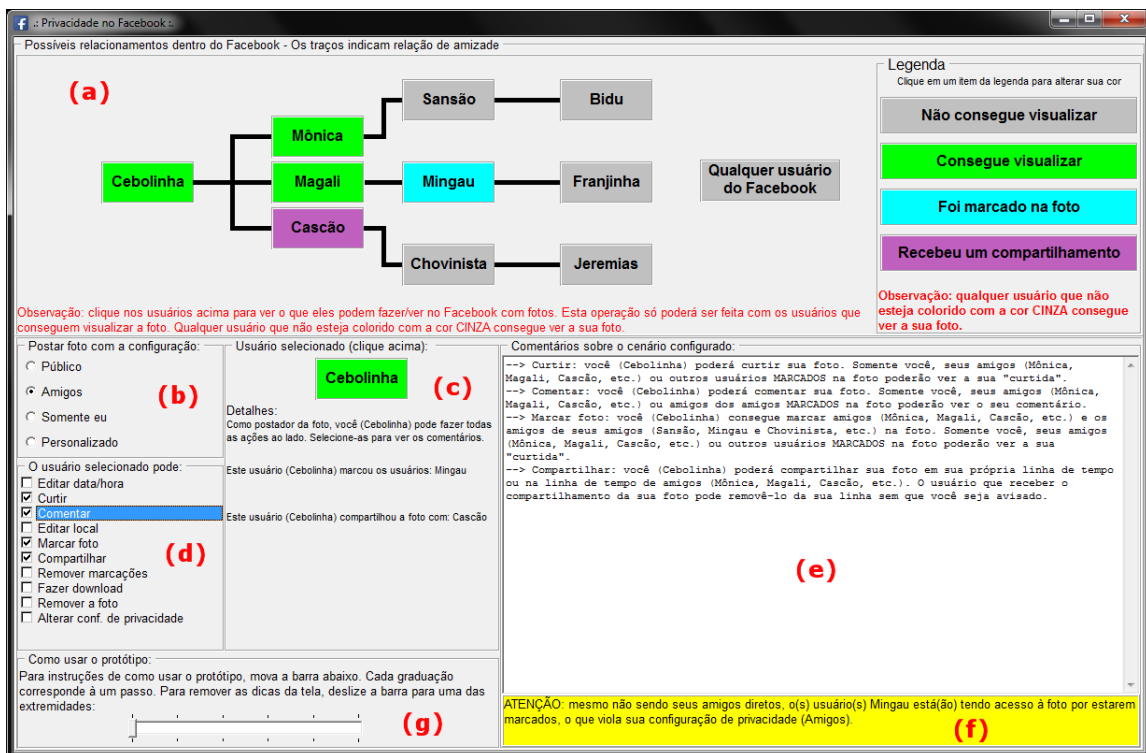


Figura 2.2. Interface do PrivSim.

A Figura 2.2 mostra a tela do protótipo desenvolvido. Cada parte da interface está identificada com uma letra e é explicada abaixo:

- Árvore que representa o usuário e sua rede de amizades, criada com nomes fictícios. Para selecionar um usuário, basta clicar em seu nome. No canto direito, uma legenda indica o significado das cores da árvore.
- É nesse local que deve ser escolhida qual a configuração de privacidade que se deseja simular (é possível selecionar a opção Amigos dos Amigos, que engloba também os amigos, na configuração Personalizado).
- Mostra o usuário que está selecionado no momento (na Figura 1 em questão, o usuário postando a foto).
- As ações disponíveis para o usuário selecionado. Com esse recurso é possível ver, para a configuração de privacidade escolhida em "b", o que o usuário selecionado pode fazer.
- Mostra explicações sobre todas as ações que foram marcadas no item "d".

- (f) Avisos relativos a inconsistências que as ações podem gerar na privacidade desejada indicada em (b).
- (g) Pequeno passo-a-passo de como utilizar o PrivSim, ao arrastar a barra, textos explicativos sobre a própria tela do PrivSim são exibidos.

Para avaliar como um simulador como o PrivSim pode ajudar usuários do Facebook a compreender o impacto de suas ações/configurações nos processos sociais, conduzimos testes com usuários reais dessa rede social. Este estudo foi conduzido em dois passos. No primeiro, os participantes realizaram tarefas relacionadas à postagem de fotos usando o Facebook, e então lhes foi perguntado os efeitos esperados destas ações. Em seguida, realizaram as mesmas tarefas usando o PrivSim e foi solicitado que explicassem se o entendimento sobre os efeitos tinha sido alterado e como.

O objetivo da primeira etapa (interagindo com Facebook) era recolher dados para responder a seguinte questão de pesquisa: *“Os usuários do Facebook compreendem os efeitos das suas configurações de privacidade no acesso que concedem a outros usuários sobre suas informações?”*. Já a segunda etapa da avaliação (interagindo com o PrivSim), teve como objetivo obter dados para uma segunda questão de pesquisa: *“Um simulador que apresenta o impacto das configurações dos usuário no acesso às suas informações aumentaria sua compreensão dos efeitos de suas decisões?”*.

Doze usuários reais do Facebook participaram da avaliação realizada. Todos eles foram previamente informados do objetivo da avaliação e das suas condições, participaram voluntariamente e assinaram um termo de consentimento. Foi realizada uma entrevista inicial com cada participante sobre suas preocupações com a privacidade e sua experiência com o Facebook.

Todos os participantes eram brasileiros e tinham contas no Facebook por pelo menos um ano e meio. Todos eram estudantes de graduação ou pós-graduação, com idade variando de 19 a 29 anos. Destes, seis pessoas tinham formação e experiência em tecnologia da informação (TI) e seis tinham formação em outros campos do conhecimento. O motivo para selecionar esses dois grupos de usuários era verificar se pessoas com experiência em TI possuíam um melhor entendimento de como Facebook lida com as suas informações, além de observar se o PrivSim poderia ajudar a melhorar sua compreensão dos impactos das configurações e ações de privacidade no Facebook. Os participantes foram igualmente distribuídos entre homens e mulheres.

Em relação à experiência dos usuários com o Facebook, o tempo de uso variou de um ano e meio a quatro anos. Todos os usuários disseram acessar o Facebook pelo menos uma vez ao dia e sete deles disseram acessar o sistema mais de dez vezes ao dia. Além disso, apenas quatro participantes nunca relataram ter mudado suas

Tabela 2.1. Resumo das tarefas e das perguntas realizadas.

| Nº | Tarefas/Perguntas |
|-----|---|
| 1 | Poste uma foto com a configuração “Amigos”. |
| 1.1 | Quem pode visualizar a foto, depois de postada? |
| 1.2 | Com quem a Magali poderá compartilhar a foto? |
| 1.3 | Quem Magali poderá marcar na foto? |
| 2 | Marque a Mônica na foto que foi postada na tarefa 1. |
| 2.1 | Após marcar sua amiga Mônica, alguém além de você e seus amigos terão acesso à foto? |
| 2.2 | A Mônica pode marcar pessoas que não têm acesso à foto (através das configurações da primeira tarefa)? |
| 3 | Poste uma foto apenas para si mesmo e “Mônica”. |
| 3.1 | A Mônica (quem tem acesso à foto) pode marcar amigos dela? Se sim, eles terão acesso à foto? Se não, explique sua resposta. |
| 4 | Poste uma foto com a configuração “Amigos de amigos”. |
| 4.1 | A Mônica pode marcar seu amigo Sansão, que não é amigo de Cebolinha? |
| 4.2 | A Mônica pode compartilhar a foto com Bidu, que é um amigo de um amigo dela? |
| 4.3 | O Sansão, amigo de uma amiga sua, poderia marcar Bidu, que é seu amigo? |

configurações de privacidade no Facebook e apenas dois disseram que não postam fotos pessoais.

Para a avaliação, um cenário foi preparado e um perfil de teste do Facebook foi criado. Este perfil de teste foi montado com os mesmos personagens fictícios presentes no PrivSim, sendo que ao logar no sistema, o usuário assumia o papel do “Cebolinha”. A avaliação consistiu basicamente de um conjunto de tarefas relacionadas com a postagem de fotos e a definição de suas configurações de privacidade. As tarefas escolhidas representavam as configurações de privacidade e ações que os usuários frequentemente realizam em relação a fotos (postar para “amigos” ou “amigos de amigos”, marcação e compartilhamento), e também ações que mudavam o conjunto de configurações padrão do Facebook. Depois de cada tarefa, os participantes foram convidados a responder um formulário eletrônico contendo questões de múltipla escolha e perguntas abertas em relação à sua compreensão de quem teria acesso a foto e o impacto esperado de determinadas ações sobre este acesso. O áudio de todo o teste foi gravado, bem como a interação dos participantes com o sistema. A Tabela 2.1 apresenta um resumo das tarefas e das questões relacionadas com elas.

Antes de interagir com o Facebook, os participantes foram informados de que durante o teste que eles poderiam fazer perguntas sobre quaisquer dúvidas que poderiam surgir sobre como executar as tarefas solicitadas na interface do Facebook. Eles tam-

bém foram informados de que poderiam voltar ao sistema para responder as perguntas, caso julgassem necessário. Depois de executar as tarefas no Facebook, os usuários passaram por uma entrevista, que teve como objetivo verificar se eles tiveram problemas para responder as questões relacionadas com as tarefas.

Na etapa de avaliação seguinte, os participantes ouviram uma breve explicação (cerca de 5 minutos) sobre a interface e funcionamento do PrivSim. Mais uma vez, os participantes foram informados de que eles poderiam pedir ajuda se tivessem dúvidas sobre a interface do PrivSim ou sobre como executar qualquer uma das tarefas previstas. Após o término das tarefas, foi realizada uma entrevista pós-teste, que tinha o objetivo de colher o entendimento dos participantes após o uso do PrivSim e em que momentos de todo o teste que eles sentiram que tinham mais dificuldades em responder às perguntas.

Os resultados obtidos para a interação com cada um dos sistemas é mostrada a seguir. A fim de fazer referência às tarefas e perguntas apresentadas na Tabelas 2.2 e 2.3 usaremos “T” para fazer referência ao número da tarefa e “TQ” para referenciar as perguntas das tarefas, onde “T” é o número da tarefa e “Q” o número da pergunta para essa tarefa. Por exemplo “1.2” refere-se à segunda questão da primeira tarefa.

Dos nove perguntas feitas, apenas duas foram respondidas corretamente por todos os participantes: a perguntas 1.1 e 4.1. A pergunta 1.1 foi respondida prontamente por todos os participantes. Já ao responderem à pergunta 4.1, cinco deles disseram que não tinham certeza se sua resposta estava correta.

A Tabela 2.2 apresenta os resultados sobre os acertos, erros e dúvidas que os participantes tiveram em cada uma das perguntas. As duas primeiras colunas mostram o número de acertos em cada pergunta agrupados pela formação dos participantes TI (ou não). A coluna “Não tiveram dúvida e erraram” mostra o número total de participantes em cada pergunta que se sentiu confiante sobre sua resposta, mas estavam errados. Finalmente, a coluna “Ficaram em dúvida” indica, para cada questão, o número total de participantes que expressaram suas dúvidas sobre a resposta correta, ou que não sabia responder corretamente.

Conduzimos um teste-t de amostras independentes para comparar a taxa de acerto entre os grupos com formação em TI e não-TI. Não houve uma diferença significativa na taxa de acerto entre o grupo não-TI ($M = 2.1111$, $SD = 2.36878$) e o grupo com formação em TI ($M = 3.0$, $DP = 2.17945$); $t(16) = -0.828$, $p = 0.42$. Estes resultados sugerem que os participantes, independentemente da sua formação e experiência em TI, tiveram dificuldades equivalentes para responder as perguntas feitas. A coluna “Ficaram em dúvida” chama a atenção para o fato de que, com exceção da questão 1.1, em todas as outras questões pelo menos metade dos participantes se sen-

Tabela 2.2. Resultados da Interação com o Facebook.

| Interação com o Facebook | | | | |
|--------------------------|-------------------|---------------|---------------------------------|----------------------|
| Nº | Acertos NÃO TI | Acertos TI | Não tiveram dúvida e erraram | Ficaram em dúvida |
| 1.1 | 6 | 6 | 0 | 0 |
| 1.2 | 0 | 0 | 4 | 8 |
| 1.3 | 0 | 2 | 5 | 6 |
| 2.1 | 1 | 1 | 5 | 6 |
| 2.2 | 1 | 3 | 2 | 10 |
| 3.1 | 1 | 5 | 0 | 10 |
| 4.1 | 6 | 6 | 0 | 5 |
| 4.2 | 3 | 2 | 3 | 8 |
| 4.3 | 1 | 2 | 0 | 11 |

tiram inseguros sobre como Facebook se comportaria na situação relativa à pergunta. Comentários como “*Não tenho dúvidas*” ou “*Eu não sei*”, seguido por pausas em que os usuários pensavam sobre o que esperar eram comuns durante o teste. Um participante (com formação em TI) explorou os textos explicativos da interface do Facebook para tentar responder corretamente algumas perguntas, mas afirmou que tais textos não o ajudaram a respondê-las.

Outro aspecto importante identificado é que, em vários casos, os participantes se sentiram confiantes sobre sua resposta, mas na verdade estavam errados, como pode ser visto na coluna “Não tiveram dúvidas e erraram”. Esta situação é ainda pior que aquela em que os participantes têm ciência de que não sabiam como o Facebook iria se comportar, pois significa que os usuários têm um mal-entendido sobre o seu funcionamento, mas não percebem isso. Esta combinação é potencialmente propensa a levar o usuário a situações indesejáveis, tais como ter pessoas acessando sua foto que a princípio não deveriam acessá-la. Ambos os casos podem potencialmente levar a transtornos sociais ou problemas para os usuários.

Um exemplo no qual esse problema pode acontecer é visto na pergunta 2.2. Nesta, o participante deveria responder que, uma vez que ele postou uma foto com acesso somente para amigos e marcou um amigo, se essa pessoa poderia marcar alguém que não fizesse parte do grupo autorizado. Oito dos doze participantes forneceram uma resposta errada, ou seja, eles acreditavam que o usuário marcado não seria capaz de marcar qualquer um de seus amigos ou amigos de amigos, independentemente de eles terem acesso à foto ou não). Entre eles, dois estavam certos de que somente os usuários autorizados (amigos) poderiam ter acesso à foto, independente de qualquer ação de

marcação na foto. Em outras palavras, neste caso os usuários não tinham a noção de que pessoas fora do grupo original (amigos) habilitado a acessar a foto poderiam acessá-la, como resultado de ações de outros usuários.

Na entrevista, após ter interagido com o Facebook, todos os participantes relataram ter tido dificuldades em responder às perguntas. Eles se referiam aos recursos de marcação ou compartilhamento como a maior fonte de dúvidas. Quando perguntados sobre o que eles pensavam que poderia ser a causa dessas dificuldades, alguns afirmaram que nunca tinha pensado sobre isso, ou nunca tinha procurado descobrir, ou mesmo que poderia ser um problema de falta de prática. Por exemplo, um dos participantes disse: *“Porque eu nunca procurei saber sobre isso, não tem muito claro em um lugar falando, você tem que procurar né, aí eu não procurei”*. Ao perguntar a ele e a outro usuário que havia dado uma resposta parecida, onde eles achavam que poderiam encontrar essa informação, se seria através da interface, ou da página de ajuda, ambos disseram que acreditavam que só conseguiriam descobrir através da ajuda do Facebook.

Ainda sobre essa questão, outro usuário, com formação em TI, respondeu sobre onde ou como conseguiria achar no Facebook explicações sobre aquilo que ele tinha tido dificuldade dizendo que *“Com uma conta só não [conseguiria descobrir], talvez eu precisaria de ter um Facebook reserva ou pedir pra um irmão, um primo ou um amigo.”*. Além disso, três usuários também indicaram que a interface do Facebook não é clara nesse aspecto, mostrando que ela não explica aos usuários o funcionamento das questões de privacidade que foram abordadas nas perguntas. Por exemplo, um deles disse *“Assim, realmente a interface não é tão clara. Pelo menos se tivesse umas mensagens de aviso [...]. Às vezes você acaba fazendo uma coisa que não é bem aquilo que você quer por falta de conhecimento, aí é meio tentativa e erro, você faz e vê o que vai dar”*.

É interessante notar que os comentários dos participantes não eram sobre como fazer o que tinha sido pedido, mas sim sobre as dificuldades que tiveram na obtenção de informações sobre o impacto das configurações e ações de privacidade. Um dos participantes relatou recorrer a uma estratégia de simulação para entender o impacto dessas configurações. Para tanto, ele usa um perfil fictício para testes ou uma conta de outro usuário que está disposto a ajudá-lo para compreender os efeitos de suas ações. No entanto, essa estratégia não seria suficiente para cobrir todos os cenários possíveis, já que isso exigiria envolvendo várias pessoas ou várias contas fictícias, cada uma tendo uma relação diferente com o seu perfil real. Um outro usuário relatou usar uma estratégia de tentativa e erro. No entanto, essa estratégia pode causar problemas até que ele encontre um erro ou uma configuração errada. Neste caso, pessoas indesejadas podem já ter tido acesso às suas informações.

Tabela 2.3. Resultados da Interação com o PrivSim.

| Interação com o PrivSim | | | | |
|-------------------------|-------------------|---------------|---------------------------------|----------------------|
| Nº | Acertos NÃO TI | Acertos TI | Não tiveram dúvida e erraram | Ficaram em dúvida |
| 1.1 | 6 | 6 | 0 | 0 |
| 1.2 | 6 | 5 | 1 | 0 |
| 1.3 | 5 | 6 | 1 | 0 |
| 2.1 | 6 | 6 | 0 | 0 |
| 2.2 | 6 | 3 | 0 | 0 |
| 3.1 | 6 | 6 | 0 | 0 |
| 4.1 | 6 | 5 | 1 | 0 |
| 4.2 | 6 | 6 | 0 | 0 |
| 4.3 | 6 | 6 | 0 | 0 |

Ao analisar a execução das mesmas tarefas com PrivSim, percebe-se que nenhum dos participantes tiveram dúvidas em responder as perguntas. Além disso, a precisão das respostas foi de aproximadamente 97%. Houveram apenas três erros ao longo do teste, aparentemente devido a alguma dificuldade em compreender a interface do PrivSim. Os resultados da avaliação com PrivSim são apresentados na Tabela 2.3.

Após a avaliação com PrivSim, foi realizada uma entrevista pós-teste. Nesta entrevista todos os participantes disseram ter tido mais dificuldades em responder as perguntas ao interagir com o Facebook do que com PrivSim. Além disso, todos eles afirmaram que o PrivSim aumentou a compreensão do impacto das configurações e ações de privacidade, e, portanto, os ajudou a responder as perguntas. Alguns deles também comentaram que durante a interação com PrivSim perceberam que algumas das respostas que haviam dado durante a interação com o Facebook estavam erradas. Um participante disse: *“Eu sei o que vai acontecer [com o uso do PrivSim]. Eu vi que o Facebook é cheio de erros, cheio de coisas que não fazem o menor sentido. Eu acho que eu acertei a maioria das perguntas que antes eu tinha errado. Então eu sei quem vai poder ver as fotos, que possibilidades a pessoa vai ter”*. Outro comentário do participante mostra como o PrivSim permitiu que ele ficasse ciente de como ele poderia estar mais exposto do que pensava. Ao responder se ele usaria o PrivSim, respondeu: *“Eu gostaria de usá-lo [PrivSim], uma vez que hoje descobri que não estou tão seguro como eu acreditava”*.

Exceto por três dos participantes que disseram que não estão preocupados com a privacidade, todos os outros disseram que gostariam que o Facebook tivesse um simulador dos efeitos das configurações de privacidade e que eles iriam usá-lo. Eles também disseram que seria melhor se o Facebook apresentasse o simulador de forma

nativa e não em forma de aplicativo, pois eles tendem a não confiar neles.

No final da entrevista, os participantes responderam se gostariam de usar um simulador de privacidade fornecido pelo próprio Facebook. Os participantes que tinham declarado se preocupar com a privacidade em sites de redes sociais mostraram mais interesse em usar o recurso, caso existisse. Além disso, alguns deles responderam que iriam utilizá-lo, uma vez que durante o experimento perceberam que não entendiam bem todos os possíveis impactos das suas configurações. Outros disseram que não iriam utilizá-lo ou que utilizariam apenas em casos específicos. Por exemplo, um usuário disse que uma potencial situação em que ele estaria interessado em usar um simulador seria quando ele quisesse postar fotos que ele não gostaria que seus pais ou namorada vissem.

A primeira questão de pesquisa deste estudo exploratório foi “Os usuários do Facebook compreendem os efeitos das suas configurações de privacidade no acesso que concedem a outros usuários sobre suas informações?” A razão para levantar essa questão é motivada pelo fato de que o acesso à informação não depende exclusivamente nas configurações de privacidade definidas pelos usuários. Um dos resultados da configuração é a definição do subconjunto de ações que estarão disponíveis para todos os usuários que têm acesso à informação. Além disso, as próprias ações, se feitas pelos usuários, podem gerar alterações no grupo de pessoas que têm acesso à foto. Em outras palavras, as configurações de privacidade, de fato, definem uma série de caminhos (ou processos sociais) que são possíveis. Diferentes caminhos podem ou não permitir o acesso a um conjunto distinto de pessoas. Portanto, explicitar todos os possíveis cenários resultantes para os usuários é uma tarefa muito difícil, se não impossível. Uma descrição em linguagem natural poderia ser uma maneira de explicar todos os cenários para os usuários, mas isso exigiria que eles tivessem o interesse em ler todo o texto.

Os resultados deste estudo exploratório sugerem que não só os usuários tiveram dúvidas sobre como Facebook se comporta, mas também tiveram um entendimento errado do seu funcionamento. Como discutido, essa última situação é ainda menos desejável do que a primeira. Estes resultados, no entanto, eram esperados. Entender como decisões sobre as definições de configuração feitas em um determinado momento vai impactar todos os outros usuários de um sistema colaborativo ao longo de um período de tempo é um problema complexo. E esse problema se torna ainda maior quando as ações de usuários finais (que podem ou não ser executadas, dependendo da sua vontade) podem alterar os cenários de uso. Acreditamos que esse problema não é específico da interface do Facebook ou de suas configurações de privacidade, mas que é um problema genérico que pode acontecer sempre que um conjunto de parâmetros é utilizado para configurar processos colaborativos envolvendo vários usuários [de Souza et al., 2010] e quando as ações dos usuários podem causar alterações nos cenários

possíveis.

Uma forma de lidar com esse problema seria permitir que os usuários simulassem os processos colaborativos e como eles são afetados pela configurações e ações de cada usuário. Neste sentido, a segunda pergunta dessa análise exploratória era “Um simulador que apresenta o impacto das configurações dos usuário no acesso às suas informações aumentaria sua compreensão dos efeitos de suas decisões?”. Os resultados da interação com o PrivSim mostraram que os usuários tiveram um entendimento completo de como as configurações e ações associadas a eles poderiam afetar a visibilidade de uma foto. Além disso, ao usar o PrivSim eles perceberam que muitas das suas impressões ao usar apenas o Facebook estavam erradas. Vale ressaltar que essas impressões (errôneas) eram baseadas não só sobre a sua interação durante o teste, mas em toda a sua experiência com o Facebook. A Figura 2.3 mostra um gráfico comparativo entre as respostas corretas obtidas através da interação com o Facebook e com o PrivSim, separado por cada tarefa (de 1 a 4). Estes resultados forneceram indícios de que uma interface que permite ao usuário simular e explorar os possíveis cenários (antes de tomar decisões) poderia ser uma boa solução para o problema levantado.

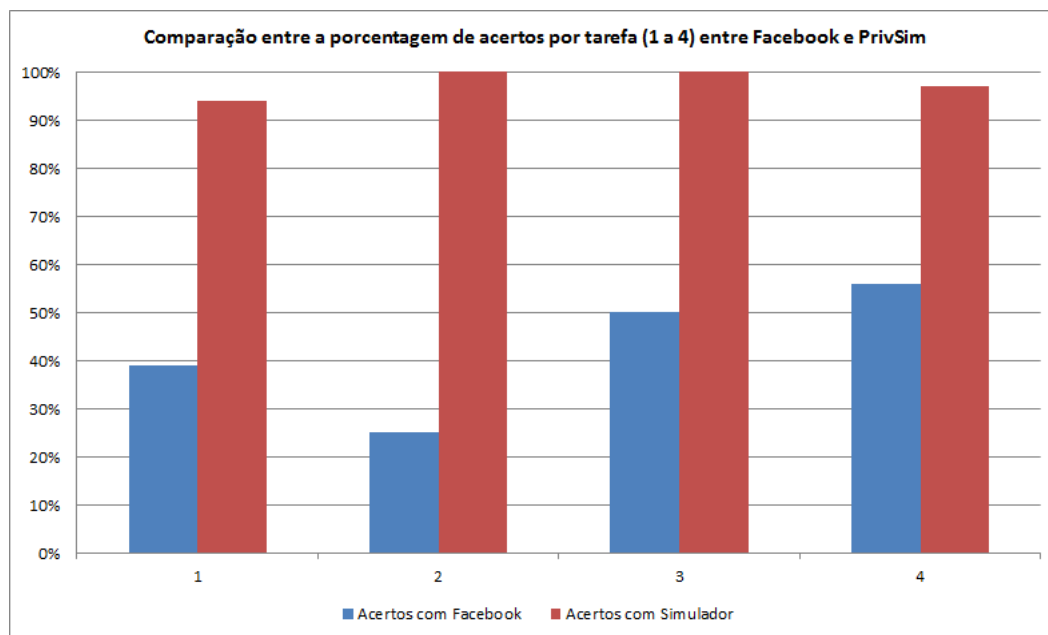


Figura 2.3. Porcentagem de acertos por tarefa usando o Facebook e o PrivSim.

Conduzimos um teste-t pareado para comparar a taxa de acerto usando o Facebook e o PrivSim. Consideramos os valores das duas primeiras colunas (como fizemos para plotar o gráfico presente na Figura 2.3), que mostram o número total de acertos para cada pergunta. Houve uma diferença significativa na taxa de acerto para PrivSim

($M = 11.6667$; $DP = 0.5$) e Facebook ($M = 5.1111$, $SD = 4.28799$); $t(8) = 4.6$, $p = 0.002$. Estes resultados sugerem que, quando utilizaram o PrivSim, os participantes tiveram um valor médio de acertos maior, se comparado com a sua interação com o Facebook.

Pode-se argumentar que o tamanho e simplicidade da rede de amizades utilizada no estudo exploratório, tanto no Facebook quanto no PrivSim é uma limitação. No entanto, no primeiro passo onde os participantes interagiram apenas com o Facebook, mesmo para uma rede simplificada de amizades, os usuários possuíam pouco entendimento sobre os impactos das ações disponíveis e configurações de privacidade relacionadas às fotos. Além do mais, os participantes eram usuários frequentes do Facebook por mais de uma ano, e a maioria deles postou fotos nessa rede social (mesmo os dois participantes que disseram não postar fotos, disseram que já haviam compartilhado fotos nas quais foram marcados). Assim, era esperado que esses participantes tivessem obtido este conhecimento nas experiências de uso real do Facebook.

Mesmo possuindo uma rede de amizades simplificada, os resultados mostram que o PrivSim foi capaz de melhorar a compreensão dos participantes sobre os impactos de suas ações e/ou configurações. Além disso, não era nosso objetivo explorar qual seria a melhor maneira de representar a simulação (no protótipo ou em sistemas reais). Avaliar se os resultados obtidos usando uma versão simplificada da rede de amizades poderiam ser generalizados para qualquer rede de amizades presente no Facebook exigiria uma investigação mais aprofundada.

2.2 Considerações Finais

Os resultados observados nos estudos exploratórios com o Facebook [Pereira Junior et al., 2012, 2013a,b, 2014] forneceram indicadores sobre a importância e utilidade de uma ferramenta de simulação. Esse estudo foi importante para que pudéssemos identificar as características que a linguagem de modelagem proposta neste trabalho, detalhada no Capítulo 4, deveria contemplar.

Apesar de o Facebook ter sido estudado sistematicamente, a nossa revisão da literatura e a nossa experiência no uso de sistemas colaborativos em geral também serviram como fonte de inspiração para a definição das nuances contempladas pela linguagem de modelagem proposta (SIGMa-dl). Já o SIGN foi totalmente definido por nós a partir das dimensões da SIGMa-dl. O próximo capítulo mostra os fundamentos teóricos desse trabalho de doutorado e apresenta os modelos existentes para projeto de sistemas colaborativos. No Capítulo 4, apresentamos em detalhes o SIGMa.

Capítulo 3

Fundamentação Teórica e Trabalhos Relacionados

Este capítulo está dividido em duas partes: primeiro, na seção 3.1 apresentamos a teoria da Engenharia Semiótica, teoria na qual este trabalho está fundamentado e os conceitos teóricos necessários para o bom entendimento do trabalho que estamos propondo. Posteriormente, na Seção 3.2 expomos os trabalhos que apresentam modelos voltados ao projeto de sistemas colaborativos e na subseção 3.2.3 discutimos as limitações de cada um dos modelos apresentados que motivam a proposta do SIGMa.

3.1 Teoria da Engenharia Semiótica

Este trabalho de doutorado se ancora na teoria da Engenharia Semiótica. A Engenharia Semiótica [de Souza, 2005] é uma teoria de IHC centrada na comunicação presente na interação humano-computador. Esta teoria nos permite compreender os fenômenos envolvidos na concepção, uso e avaliação de um sistema interativo. Além disso, enriquece o entendimento do projetista sobre as possíveis soluções para determinado problema, trazendo para a sua reflexão elementos não considerados pelos métodos tradicionais de IHC [de Souza, 2005] [Prates & Barbosa, 2007] [Barbosa & Silva, 2010].

Para a Engenharia Semiótica, o projetista se comunica com o usuário através da interface. Por este motivo, a interface age como sua representante, ou seja, ela repassa a intenção do projetista ao usuário da aplicação, termo que a Engenharia Semiótica define como **preposto** do projetista [de Souza, 2005, p.112]. Segundo de Souza [2005, p.84], a mensagem sendo enviada do projetista para o usuário através da interface pode ser parafraseada no seguinte modelo genérico:

Este é o meu entendimento de quem você é, do que aprendi que você quer ou precisa fazer, de que maneiras prefere fazer, e por quê. Este, portanto, é o sistema que projetei para você, e esta é a forma como você pode ou deve utilizá-lo para alcançar uma gama de objetivos que se encaixam nesta visão.

O usuário entende esta mensagem à medida que interage com a interface de um sistema. Como essa comunicação entre projetista-usuário se dá através da interface do sistema (usuário-sistema), essa interface é considerada um artefato de metacomunicação. Cabe ressaltar que esta mensagem (transmitida através da interface) é indireta, já que a interface age como preposto do projetista, e unidirecional, já que o usuário não pode dar continuidade à comunicação com o projetista [de Souza, 2005; Prates & Barbosa, 2007].

Claramente nota-se que a Engenharia Semiótica entende o papel do projetista como sendo tão importante quanto o do usuário no contexto de um sistema interativo. Através da metacomunicação, o processo de design deixa de focar em “*como o usuário vai aprender um determinado sistema*” e foca em “*como o projetista vai apresentar a solução ao usuário*”. Portanto, através da metacomunicação, o projetista procura mostrar ao usuário as possibilidades de interação presentes na interface. A **comunicabilidade** é a qualidade de sistemas interativos de comunicar de forma eficiente e eficaz ao usuário as intenções de projeto e princípios de interação. Para avaliar a comunicabilidade, a Engenharia Semiótica dispõe de dois métodos: o Método de Inspeção Semiótica (MIS) [de Souza et al., 2006] e o Método de Avaliação da Comunicabilidade (MAC) [Prates et al., 2000][de Souza, 2005] [de Souza & Leitão, 2009].

A Engenharia Semiótica também endereça as particularidades dos sistemas colaborativos, contexto no qual este trabalho de doutorado se encaixa. Para a Engenharia Semiótica, projetistas de sistemas colaborativos devem se preocupar com detalhes adicionais ao projetar a sua aplicação. Segundo de Souza [2005, p.210], a metamensagem do projetista deste tipo de aplicação para o usuário pode ser parafraseada no seguinte modelo genérico:

Este é o meu entendimento, como projetista, de quem você, usuário, é, do que aprendi que você quer ou precisa fazer, de que maneiras prefere fazer, e por quê. Este, portanto, é o sistema que projetei para você, e esta é a forma como você pode ou deve utilizá-lo para alcançar uma gama de objetivos que se encaixam nesta visão. Você pode se comunicar e interagir com outros usuários através do sistema. Durante a comunicação, o sistema irá ajudá-lo a verificar:

- *Quem está falando? Para quem?*
- *O que o falante está dizendo? Qual o meio e código usados? Os códigos e meios são apropriados para a situação? Existem alternativas?*
- *O(s) ouvinte(s) está recebendo a mensagem? E se não estiver?*
- *Como o(s) ouvinte(s) pode(m) responder o falante?*
- *Existe recurso se o falante percebe que o(s) ouvinte(s) não compreendeu(ram) a mensagem? Qual é ele?*

Para apoiar o projetista nas suas reflexões sobre as suas decisões de design, seja em sistemas monousuário ou colaborativos, a Engenharia Semiótica preconiza que lhes sejam oferecidas **ferramentas epistêmicas**. Essas ferramentas são usadas pelo projetista não para produzir diretamente a resposta para um problema, mas para aumentar a sua compreensão sobre o problema em si e suas implicações [de Souza, 2005]. Neste sentido, o modelo proposto neste trabalho surge como uma nova ferramenta epistêmica, permitindo ao projetista antecipar os cenários que pretende disponibilizar aos usuários finais. Dessa forma ele pode refletir sobre suas decisões de design (que estão gerando os cenários que estão sendo antecipados) e pode verificar se esses cenários são ou não desejáveis na implementação final do sistema.

Finalmente, a Engenharia Semiótica dispõe de uma metagemagem para auxiliar o projetista de sistemas customizáveis e extensíveis na sua tarefa de considerar as customizações e extensões previstas para o sistema que está projetando. A metagemagem pode ser parafraseada no seguinte modelo genérico:

*Este é o **meu entendimento**, como projetista, de **quem você**, usuário, é, do que aprendi que você **quer ou precisa fazer**, de que **maneiras prefere fazer**, e **por quê**. Este, portanto, é o sistema que projetei para você, e esta é a forma como **você pode ou deve utilizá-lo** para alcançar uma gama de objetivos que se encaixam nesta visão. Mas, eu sei que **você pode querer modificar** a minha visão, a fim de fazer coisas (de formas) que eu não pensei. Eu **posso lidar com as mudanças** que você pode querer fazer, desde que você possa dizer o que quer neste código particular.*

3.2 Modelos para Projetos de Sistemas Colaborativos

Na seção anterior, apresentamos brevemente a teoria da Engenharia Semiótica e seus principais conceitos que são relevantes para este trabalho. A seguir, mostramos os principais modelos propostos pela Engenharia Semiótica para a modelagem de sistemas colaborativos (subseção 3.2.1) e logo em seguida os principais modelos que não se baseiam nessa teoria (subseção 3.2.2).

3.2.1 Modelos fundamentados na Engenharia Semiótica

Nesta subseção descrevemos dois modelos propostos pela Engenharia Semiótica para o desenvolvimento de sistemas colaborativos, o MetaCom-G e a Manas.

3.2.1.1 MetaCom-G e MArq-G

Em 1998, Prates [1998] propôs um modelo para a construção de sistemas multi-usuários, denominado Modelo Abstrato do Componente Multi-usuário do Artefato de Metacomunicação, ou MetaCom-G. Surgiu para ser aplicado ao desenvolvimento de sistemas que envolvessem o trabalho cooperativo, ou seja, sistemas onde vários usuários interagem para atingir um determinado objetivo. O modelo é composto de uma linguagem de design, uma base de conhecimento, um simulador de cenários e um conselheiro de *widgets* [Prates, 1998, p.31].

A linguagem de design é independente do domínio¹, através da qual o projetista descreve a mensagem que, na sua concepção, deve ser enviada aos usuários. Além da linguagem de design, o modelo conta com regras heurísticas que agem sobre elas. Com isso, o projetista recebe indicadores qualitativos sobre a sua metamensagem. O objetivo das regras era apontar aparentes inconsistências nas combinações de valores feitas pelo projetista. Ao receber as potenciais inconsistências do seu modelo, o projetista deve julgá-las, a fim de determinar se realmente fazem sentido (ou não) para o seu domínio. Por isso, o modelo é tido como descritivo, ou seja, ele deixa a cargo do projetista decidir o que faz ou não sentido e o que deve ou não ser descartado. Esta característica está de acordo com a teoria da reflexão-em-ação, de Schön [1983]. Através do seu modelo e do feedback gerado, o projetista pode refletir sobre suas decisões e esta reflexão pode influenciar suas decisões futuras.

¹Ao descrever o seu sistema, em particular, o projetista atribui valores às características da linguagem de acordo com o seu domínio de interesse.

De acordo com Prates [1998], na descrição de seu modelo, o projetista deve se preocupar com as seguintes dimensões:

- **Papéis:** durante a interação, cada usuário pode assumir diferentes funções no grupo. Cabe ao projetista definir quais serão estes papéis.
- **Hierarquia:** além dos papéis, os usuários podem ainda estarem inseridos em uma hierarquia. O projetista deve definir qual é a hierarquia entre os membros do grupo, se é que ela existirá. Esta hierarquia pode ainda ser tratada a nível tecnológico (rígida - definida pelo sistema), ou pode ser deixada para o protocolo social, onde os membros entram em acordo para defini-la.
- **Níveis de interação:** os membros do grupo podem interagir de três formas: individual, interpessoal e a nível de contexto. No *nível individual*, o usuário interage apenas com sua parte privada da aplicação. Quando os membros se comunicam através do sistema, tem-se o *nível interpessoal*. Por fim, no *nível de contexto*, o usuário percebe informações sobre a organização do grupo, sua estrutura hierárquica, além de informações sobre quem está usando o sistema, em que tarefas está trabalhando, o que deve ser feito, e assim por diante.
- **Objetos:** tudo sobre o que o usuário pode agir é denominado objeto. Estes podem ser *privados* – pertencentes a um único usuário, *compartilhados* – pertencentes a dois ou mais membros ou *públicos* – pertencentes a todos os membros do grupo. Cabe ao projetista definir quais serão os objetos e como será o acesso a eles.
- **Habilidades de Comunicação:** são as formas que os membros do grupo tem disponíveis para se comunicarem sobre objetos. Os elementos básicos de comunicação são: visão, discurso e ação [Prates et al., 1997]. Ao agir sobre um objeto, um membro automaticamente deve poder *vê-lo*. Além disso, eles podem falar sobre determinado objeto (*discurso*). O projetista deve definir se membros podem ver e falar sobre objetos de outros. Já a *ação* pode ser considerada uma comunicação indireta, quando não há a visão e/ou o discurso. Isso se dá pelo fato de que os atos de cada membro sobre determinado objeto é que darão base para o entendimento dos demais sobre seus objetivos e intenções.
- **Modelos de colaboração:** o projetista deve definir como se dará a interação entre os usuários de um sistema colaborativo. Esta pode se dar de formas distintas. No *modelo de ilha*, os usuários trabalham isolados uns dos outros. As

tarefas não exigem a comunicação entre os usuários. O projetista pode permitir, no entanto, a comunicação entre membros no protocolo social. No *modelo de encaixe nebuloso*, a interação entre as tarefas dos membros se dá a nível social, ou seja, é determinada pelos usuários. Para que esta negociação entre os membros aconteça, eles precisarão se comunicar. Já no *modelo de encaixe rígido*, a interação entre as tarefas é definido pelo protocolo tecnológico, ou seja, pelo sistema. Outro modelo é o chamado *modelo de sobreposição*. Neste, o trabalho de um indivíduo é feito no nível individual, enquanto outra parte é compartilhada no nível interpessoal. Por fim, no *modelo único ou coincidente*, as atividades acontecem totalmente no nível interpessoal.

- **Widgets:** o projetista deve escolher quais serão os componentes (*widgets*) presentes na interface para que os usuários possam interagir. A primeira classe de *widgets* são os *indicadores* – aqueles que não permitem interação com outros membros do grupo, mas fornecem informações sobre eles (onde estão, o que estão fazendo, etc). Tem-se ainda a classe dos *acionadores* – aqueles que permitem ao usuário agir sobre a informação sendo apresentada e comunicar-se com a aplicação e demais usuários. Por fim, tem-se os *applets*, que podem ser acoplados ao sistema (como um calendário, por exemplo).
- **Tempo x Espaço:** os intervalos de tempo e espaço dizem respeito aos estados pelos quais os membros podem passar e quais subambientes podem frequentar.

Um outro aspecto que deve ser levado em consideração são as mudanças que podem ocorrer no contexto do sistema durante o tempo. Os membros podem estar trabalhando em uma tarefa que lhes exige exercer diferentes papéis ao longo do tempo, ou que lhes impõe diferentes formas de interação². Este conceito é definido como **eras** [Prates, 1998, p.39]. Cada era é representada no modelo como uma descrição estática. Quando o contexto do sistema envolve a evolução entre eras distintas, tem-se um aspecto dinâmico. O projetista deve descrever todas as eras estáticas e o que muda entre elas (comportamento dinâmico). As transformações entre eras podem se dar de duas maneiras distintas. A primeira delas é quando o projetista é quem define o que muda, denominada simplesmente **mudança**. A segunda é quando o projetista oferece aos membros um conjunto transformacional, ou seja, um conjunto que permite aos usuários definirem o que será alterado de uma era para a outra. Neste caso esta transformação é chamada de **meta-mudança**.

²Prates [1998, p.27] apresenta um sistema de exemplo onde estas mudanças são facilmente notadas.

Como comentado no início desta seção, o modelo aqui descrito é abstrato. Por este motivo, serve de base para a construção de diferentes modelos concretos de ambientes de apoio ao projetista de interfaces multi-usuário. Além da MetaCom-G, em seu trabalho, Prates [1998] também propõe uma instância da MetaCom-G, o modelo de arquitetura de design de interfaces multi-usuário – MARq-G.

O MARq-G, como instância da MetaCom-G, tem o propósito de apoiar o projetista durante a fase de planejamento da interface multi-usuário. O modelo é composto de uma linguagem de design, uma base de conhecimento, um simulador de cenários e um conselheiro de *widgets* [Prates, 1998, p.31]³.

O projetista faz a descrição conceitual do seu modelo de grupo utilizando **constructores**, que compõem a parte léxica da linguagem de design. Existem construtores disponíveis para todas as dimensões do MetaCom-G, comentadas anteriormente. As **regras heurísticas**, que compõem a parte semântica da linguagem e que são independentes de contexto e descritivas, agem sobre a definição do grupo feita através destes construtores. Com isso, o projetista recebe um feedback dos potenciais problemas do seu modelo, juntamente com uma explicação, para que ele possa refletir se o potencial problema identificado é realmente um problema, ou se não faz sentido no seu contexto. Em ambos os casos, ao tomar esta decisão, o projetista registra na base de conhecimento as suas motivações e justificativas sobre o problema, para que possa ter toda a lógica de design armazenada.

O modelo permite tanto mudanças quanto meta-mudanças. Ele prevê, conceitualmente, que possíveis inconsistências que podem acontecer com as meta-mudanças possam ser antevistas pelo projetista através do uso do **cenógrafo**. Finalmente, também no nível conceitual, o modelo prevê o **conselheiro de widgets**, que tem como objetivo auxiliar o projetista na realização da interface da aplicação multi-usuário, fornecendo sugestões sobre os *widgets* a serem escolhidos. No entanto, tanto o cenógrafo quanto o conselheiro de *widgets* nunca foram implementados.

Ao término da modelagem, o projetista recebe um relatório dividido em eras e as transformações entre elas. Apresenta também todos os valores atribuídos pelo projetista aos construtores em cada uma das eras (membros e papéis, hierarquia entre eles, etc.). Finalmente, são apresentados os potenciais problemas identificados, juntamente com as justificativas fornecidas pelo projetista. O projetista pode, também, optar por ver as sugestões do conselheiro de *widgets*.

Mais tarde, Barbosa [2002] concluiu que a caracterização do processo de comunicação do grupo através das dimensões da MetaCom-G é feita em um nível de abstração

³Uma descrição detalhada e abrangente da MARq-G foge ao escopo deste trabalho, mas pode ser consultada no capítulo quinto da obra de Prates [1998].

muito alto. Por isso, propôs alterações no MetaCom-G e também na MArq-G, para ampliar o seu potencial. Os novos modelos gerados foram denominados MetaCom-G* e MArq-G*.

Posteriormente, analisando outros sistemas multi-usuários, como as comunidades online, Barbosa [2006, p.46] concluiu que a MArq-G* não contemplava aspectos sociais importantes nestes ambientes, como confiança, privacidade e polidez. Por isso, elaborou uma ferramenta epistêmica mais abrangente, a **Manas**, que não é focada no trabalho colaborativo de um grupo, mas sim no processo de comunicação entre usuários, mediado pelo sistema. Além disso, a Manas requer que o projetista reflita sobre a interface como um projeto de comunicação. Apresentamos a Manas na subseção seguinte.

3.2.1.2 Manas

A Manas [Barbosa, 2006] é uma ferramenta epistêmica fundamentada na Engenharia Semiótica. Seu objetivo é permitir que projetistas descrevam como se dará a comunicação intermediada por computador entre usuários. A este modelo, dá-se o nome de Modelo da Comunicação Usuário-Sistema-Usuário, ou m-ComUSU. Este modelo é descrito através da Linguagem de Design da Comunicação Usuário-Sistema-Usuário, ou L-ComUSU. Com base no modelo desenvolvido a Manas é capaz de apontar potenciais problemas sociais que podem acontecer durante a comunicação [Barbosa et al., 2007].

A arquitetura da Manas prevê quatro componentes, que são: (1) a Linguagem de design da comunicação usuário-sistema-usuário, a L-ComUSU, (2) o interpretador, (3) a base de conhecimento e (4) o gerador de relatórios [da Silva, 2009].

A L-ComUSU define itens léxicos e regras sintáticas. Os itens léxicos representam os elementos comunicativos, enquanto que as regras sintáticas definem como estes elementos se relacionam entre si [Barbosa et al., 2007].

A comunicação entre os usuários, denominados *interlocutores*, se dá através de estruturas comunicativas. A primeira delas é a *fala* - o ato de comunicação individual, realizado por um interlocutor. E a segunda é a *conversa* - ato de comunicação entre interlocutores, mediada pelo sistema. Ao definir seu modelo, o projetista pode organizar os elementos comunicativos (*fala* e/ou *conversa*) temporalmente, ou seja, pode criar uma hierarquia entre eles, que define o momento em que podem ocorrer. Quando o *interlocutor* profere uma *fala*, ele é denominado *falante*, enquanto que, quando ouve, é denominado *ouvinte*. [Barbosa, 2006]

Tanto a *fala* quanto a *conversa* são definidos como elementos comunicativos. Estes, por sua vez, são compostos por subelementos comunicativos, que os caracterizam [Barbosa, 2006]. A *fala*, tem um *falante* e um conjunto de *ouvintes*,

endereçados ou não-endereçados. Os ouvintes endereçados são aqueles a quem a comunicação se destina, enquanto que os não-endereçados, participam da comunicação, porém a fala não é direcionada a eles. No ato da fala, tem-se ainda o propósito ao falar, que é o principal objetivo do falante ao enunciá-la. Também o tópico, que representa o assunto ao qual a fala se refere. Por fim, a fala possui também o subelemento comunicativo conteúdo, que representa o próprio conteúdo da comunicação [Barbosa, 2006] [Barbosa et al., 2007] [da Silva, 2009]. A Figura 3.1, adaptada de [da Silva, 2009], mostra os subelementos comunicativos que compõem o elemento comunicativo fala.

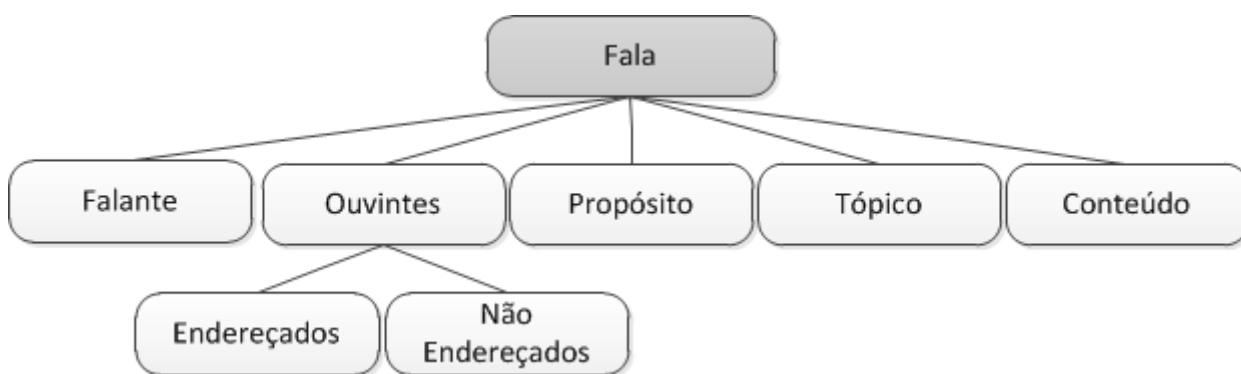


Figura 3.1. Subelementos componentes do elemento comunicativo fala.

Já o elemento comunicativo conversa tem um conjunto de interlocutores. Os interlocutores envolvidos na conversa podem proferir falas ou ouvi-las. Assim como na fala, os interlocutores podem ser endereçados ou não-endereçados. Os interlocutores endereçados estão diretamente envolvidos na comunicação e, portanto, são responsáveis pelo seu desenvolvimento em busca do propósito. Já os não-endereçados não estão diretamente envolvidos na comunicação, porém a conversa não é direcionada a eles. Assim como a fala, a conversa também possui propósito. Porém, em vez de tópico, a conversa possui um tema, que representa o assunto sobre o qual os interlocutores conversam. Obviamente, uma conversa pode conter falas [da Silva, 2009]. A Figura 3.2, adaptada de [da Silva, 2009], mostra os subelementos comunicativos que compõem o elemento comunicativo conversa.

Cada um dos subelementos comunicativos são representados por atributos. Estes são resumidamente descritos abaixo⁴:

- **representação explícita:** indica se o subelemento comunicativo está explicitamente representado na interface (SIM ou NÃO). Pode ainda assumir o valor NA

⁴Uma descrição completa e detalhada dos atributos pode ser encontrada em [Barbosa, 2006] e [da Silva, 2009]

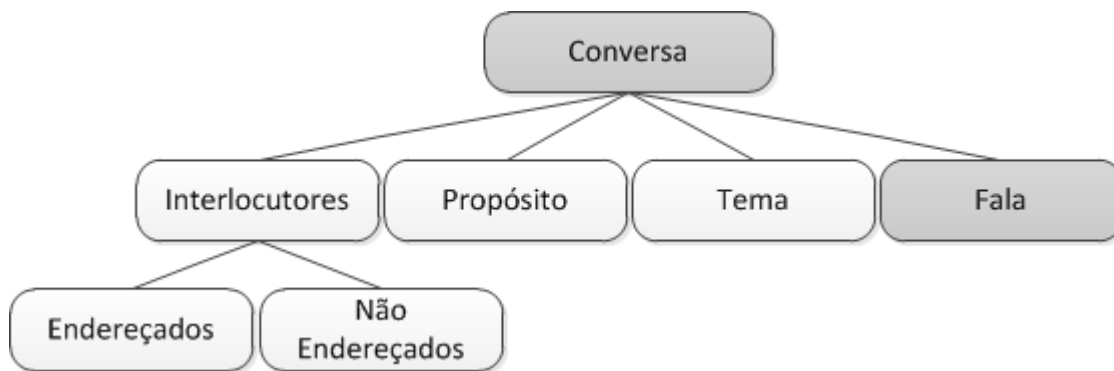


Figura 3.2. Subelementos componentes do elemento comunicativo conversa.

(não se aplica).

- **escopo:** indica os valores possíveis que um subelemento comunicativo pode assumir.
- **propósito:** indica a intenção comunicativa. Os valores possíveis para este atributo são diferentes entre a fala e a conversa. Para a fala, pode assumir os valores assertivo, diretivo, compromissivo, expressivo, declarativo e livre. Já para a conversa, pode assumir os valores descritivo, deliberativo, declarativo, expressivo e livre.
- **determinador do valor:** indica quem, dentro do contexto do sistema, é responsável por atribuir o valor do subelemento comunicativo. Estes responsáveis, por sua vez, podem ser o usuário, o preposto, ou ambos.
- **valor obrigatório:** indica se o usuário é obrigado a inserir um valor para o subelemento comunicativo. Este atributo só faz sentido (e, portanto, só fica disponível) se o atributo determinador do valor receber o valor usuário. Pode assumir os valores SIM e NÃO.
- **valor padrão:** indica se o subelemento comunicativo possui um valor padrão estabelecido. Este atributo só faz sentido (e, portanto, só fica disponível) se o atributo determinador do valor receber o valor usuário. Pode assumir os valores SIM e NÃO.
- **nível de processamento:** indica o nível de processamento que o preposto realiza sobre o conteúdo do subelemento comunicativo. Pode assumir os valores básico (quando apenas exibe o conteúdo), intermediário (quando oferece

mecanismos para recuperar, filtrar ou organizar a informação) ou inferencial (quando desencadeia processos, ações ou raciocínios).

Em seu trabalho, da Silva [2009, p.33] avaliou a Manas em dois estudos de caso e, com base nestes, propôs algumas alterações, resumidamente comentadas a seguir⁵:

- criação de um novo atributo para o elemento comunicativo *fala*. O atributo foi denominado *tipo de signo* e pode assumir os valores *estático*, *dinâmico* e *metalinguístico*, além do valor nulo, que indica que o projetista não pretende especificar o tipo do signo naquele momento.
- diferenciação da visão do falante (*emissão*) da visão do ouvinte (*recepção*) em uma *fala*. Como esperado, na *emissão*, o projetista descreve a visão do falante ao enunciar uma *fala*. Já na *recepção*, o projetista descreve a visão do ouvinte da *fala* (que pode ser diferente da *emissão*, por isso a diferenciação). Assim, a *fala* foi detalhada em *emissão* e *recepção*, como visto na Figura 3.3. Estas por sua vez possuem os subelementos comunicativos *falante*, *ouvintes* (endereçados e não-endereçados), *propósito ao falar*, *tópico* e *conteúdo*. As propostas podem ser vistas nas Figuras 3.4 e 3.5.



Figura 3.3. O detalhamento do elemento comunicativo *fala*.

- alteração de valores do atributo *representação explícita* na *recepção*. Independente da *emissão*, cada subelemento comunicativo pode estar ou não presente na *recepção*. Portanto, os valores possíveis para a *representação explícita* na *recepção* são: **herdado** (quando o subelemento comunicativo está representado explicitamente na *emissão* e continua explicitado na *recepção* sem alterações ou quando não está representado explicitamente na *emissão* e assim continua na *recepção*), **sim** (quando o subelemento comunicativo não está representado explicitamente na *emissão* e passa a estar na *recepção* ou quando está representado explicitamente na *emissão* e continua

⁵Este texto se limita a citar as alterações propostas. Para compreender os motivos de cada alteração proposta veja [da Silva, 2009]



Figura 3.4. A emissão proposta e seus respectivos subelementos comunicativos.



Figura 3.5. A recepção proposta e seus respectivos subelementos comunicativos.

representado na recepção, mas com escopo diferente - neste caso o projetista pode informar o novo valor do escopo) e **não** (quando o subelemento comunicativo está representado explicitamente na emissão e não na recepção).

- consideração do momento da fala. Em algumas falas, o falante pode ter acesso a alguns subelementos comunicativos quando já não é possível alterá-los (por exemplo, ao enviar um convite sem antes visualizar a mensagem que será enviada). Mesmo que não possa alterá-los, se tiver ciência de um problema, o falante pode usar outros meios para corrigi-lo, explicitando as limitações do sistema da Silva [2009, p.37]. Portanto foi criado o atributo momento da representação, que pode assumir os valores preliminar (caso o subelemento esteja representado explicitamente **antes** que a fala seja enunciada), posterior (caso o subelemento esteja representado explicitamente **depois** que a fala seja enunciada) e **não se aplica** ou **NA** (quando o projetista não desejar especificar o valor naquele momento).

- nível de processamento. Na **emissão** de uma fala, o sistema pode realizar algum tipo de processamento ou permitir que alguma ação seja realizada. Portanto, o escopo foi alterado para **permissivo** (quando o sistema permite que somente o falante altere alguma informação ou exclua sua fala após enunciada ou, quando o sistema permite que os ouvintes, endereçados ou não, visualizem a fala após enunciada) e **inferencial** (quando o sistema desencadeia raciocínios, processos ou ações a partir do conteúdo do subelemento comunicativo). Já na **recepção**, o atributo nível de processamento deu lugar à um outro atributo, denominado nível de visualização. Este pode assumir os valores **simples** (quando o sistema apenas exibe a informação), **elaborado** (quando o sistema permite ações como organização, recuperação da informação ou filtros, para os ouvintes relacionados na recepção) e **não se aplica** ou **NA** (quando o projetista não desejar especificar o valor naquele momento).
- mudança nas regras semânticas. Algumas regras semânticas já existentes tiveram que ser alteradas, para acomodar as mudanças propostas. Foram ainda propostas duas novas regras, que contemplam o tipo de signo e o momento da representação.

Além das alterações propostas, foi implementada uma ferramenta, denominada SMART [da Silva et al., 2010], que permite que o projetista faça a modelagem de seu sistema colaborativo através de uma interface gráfica (na proposta anterior a interação só era possível através de linhas de comando) [da Silva, 2009, p.41]. Dessa forma, a SMART facilita a modelagem usando a Manas.

3.2.2 Modelos gerais

Nesta subseção apresentamos os principais modelos concebidos para a modelagem de sistemas colaborativos.

3.2.2.1 *Groupware Task Analysis* - GTA

O *Groupware Task Analysis* (GTA) [Veer et al., 1996] é um *framework* que permite a modelagem de diferentes aspectos de sistemas colaborativos. O *framework* se concentra em três pontos de vista: **agentes**, **trabalho** e **situação**, que, segundo o autor, permite que projetistas vejam o problema de diferentes ângulos.

Ao usar este *framework*, o projetista captura conhecimento de uma situação problema observando os três pontos de vista que formam a sua base e cria o chamado “modelo de tarefas 1”. Esse modelo contém a descrição da situação atual de trabalho ou a descrição de uma situação problema. Em seguida, deve-se criar um outro modelo,

chamado de “modelo de tarefas 2”. Esse segundo modelo é baseado no primeiro, mas conta com mais detalhes sobre a aplicação, incluindo as decisões de design tomadas ao observar os problemas e conflitos identificados no “modelo de tarefas 1”. Os detalhes incluem os artefatos que serão utilizados, a sua relação com usuários e demais interessados no sistema (*stakeholders*), seus objetivos e tarefas e o contexto de uso, bem como as tecnologias que serão utilizadas para resolver o problema.

Os três pontos de vista considerados pelo GTA foram organizados em uma ontologia que apresenta os principais conceitos relevantes para a análise de tarefas e as relações entre eles, como visto na Figura 3.6.

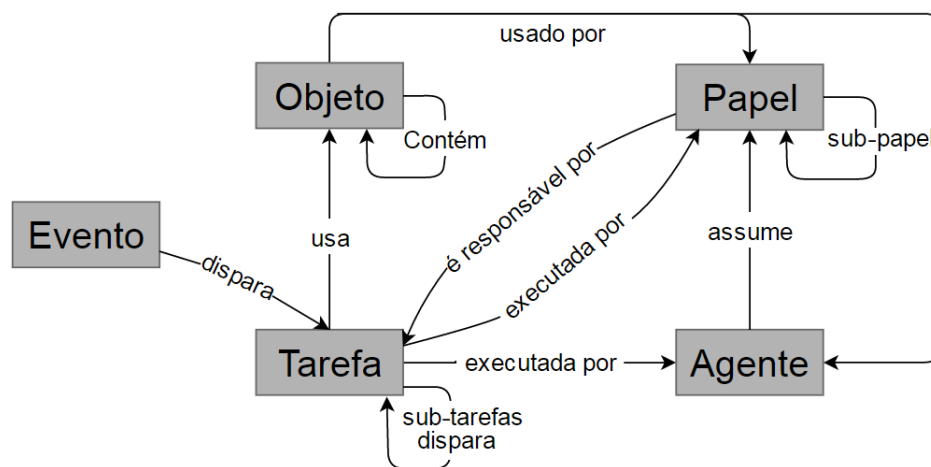


Figura 3.6. Ontologia do GTA (adaptado de [Welie et al., 1998]).

Os **objetos** se referem a entidades físicas ou abstratas (por exemplo uma senha ou mensagem). Ações especificam o que pode ser feito com os objetos. Objetos podem ser envolvidos em uma hierarquia e podem possuir outros objetos.

Os **agentes** se referem à entidades ativas e indicam uma classe de indivíduos com certas características. Normalmente são humanos, mas grupos de humanos ou componentes de software também são considerados agentes.

Os **papéis** caracterizam uma coleção de tarefas que podem ser executadas por um ou mais agentes. Os papéis podem ser compostos hierarquicamente.

As **tarefas** caracterizam as atividades executadas por agentes para atingir um determinado objetivo. Normalmente envolvem um tempo para serem executadas e geram alguma alteração. Uma tarefa pode ser composta por subtarefas e pode disparar outras tarefas. Finalmente, os **eventos** indicam uma mudança em um ponto determinado que dispara uma tarefa.

Essa ontologia define apenas os conceitos e as relações entre eles e não impõe qualquer tipo de representação gráfica. Uma ferramenta chamada EUTERPE [Welie

et al., 1998] foi criada para apoiar a modelagem de sistemas com base na ontologia definida. A ferramenta permite a criação de instâncias dos cinco conceitos previstos pela ontologia de forma visual. Para cada instância é possível ainda informar diversos atributos próprios. Para cada tarefa, por exemplo, é possível informar seu objetivo, seu tipo e relevância, quais tarefas são disparadas por esta, etc. Já para os objetos, é possível descrever quem terá acesso a eles e qual o tipo de acesso (criar, alterar, excluir, etc). Para os papéis é possível informar suas responsabilidades. Para os agentes é possível definir o seu papel e o seu tipo (indivíduo, grupo ou sistema). Finalmente para os eventos é possível definir quais tarefas os disparam. A Figura 3.7 mostra um fragmento de um modelo feito na ferramenta EUTERPE. Esse exemplo é distribuído com a própria ferramenta e ilustra as tarefas de edição de um documento. É possível ver na figura os cinco conceitos previstos pela ontologia do GTA (local indicado por uma seta vermelha na mesma Figura 3.7). Para cada um dos conceitos o projetista pode criar instâncias com seus respectivos atributos.

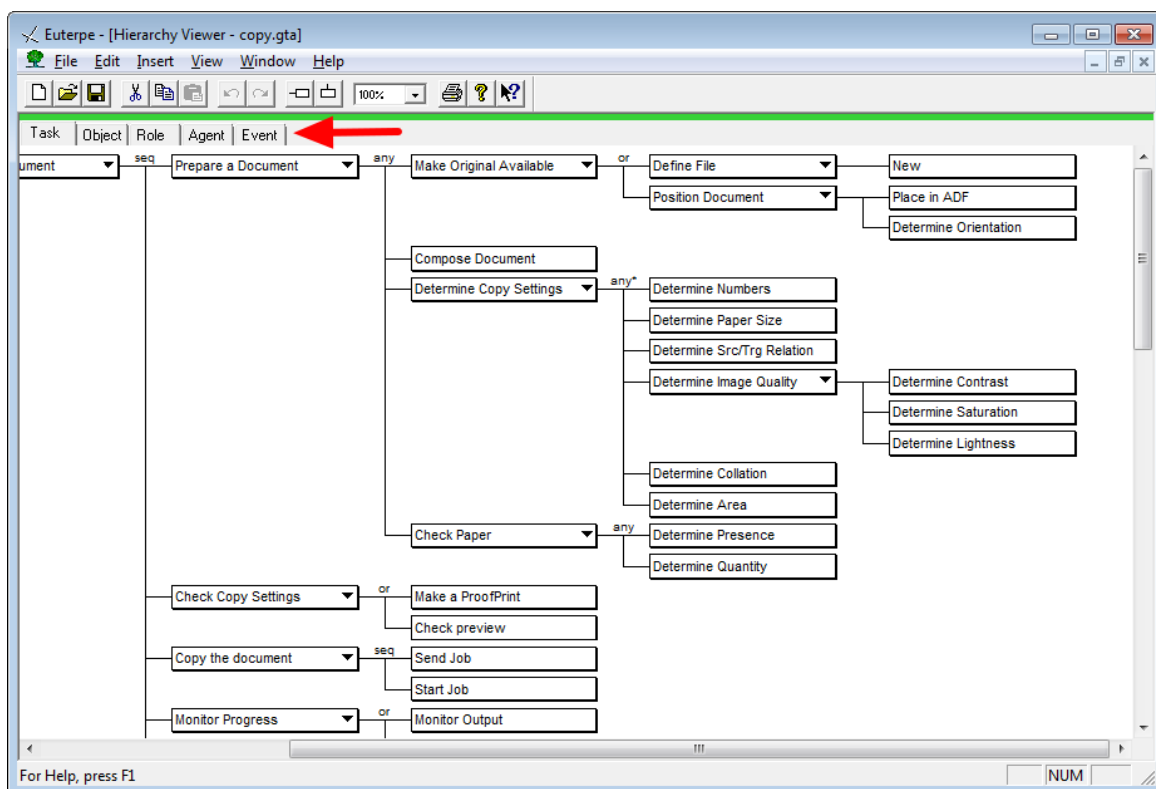


Figura 3.7. Imagem da interface da ferramenta EUTERPE mostrando um modelo de exemplo.

Ao terminar o seu modelo de tarefas, o projetista pode gerar diferentes visualizações gráficas, como a representação da árvore de tarefas e outras representações hierárquicas, assim como diferentes níveis e possibilidades de análise.

3.2.2.2 *Concurrent Task Trees - CTT*

Segundo Paternò [1999], a tarefa de compreender o comportamento dinâmico de uma aplicação que resulta de diversas relações temporais pode ser difícil, pois é preciso manter todas as sequências possíveis de ações e seus efeitos. O autor então propõe o *Concurrent Task Trees* (CTT), ou *ConcurTaskTrees* na sua forma abreviada [Paternò et al., 2001; Paternò, 2003]. Este modelo é focado na modelagem de tarefas, permite a representação de atividades colaborativas ou cooperativas e a organização hierárquica de tarefas. A criação de modelos CTT é feita de forma gráfica através de uma ferramenta chamada *ConcurTaskTrees Environment* (CTTE) [Mori et al., 2002].

O CTT prevê quatro tipos diferentes de tarefas, de acordo com “quem” as executa [Paternò, 2003]. Para cada tarefa, existe uma representação gráfica que a indica na CTTE, conforme Figura 3.8. Os tipos de tarefa previstos são:

- **tarefas do usuário:** são tarefas realizadas inteiramente pelo usuário, sem a intervenção do sistema. Por exemplo, aguardar o processamento de um determinado dado. Sua representação gráfica pode ser vista na Figura 3.8 (a).
- **tarefas do sistema:** ao contrário da anterior, são tarefas realizadas inteiramente pelo sistema, sem a intervenção do usuário. Por exemplo, executar a compilação de um documento $\text{L}^{\text{T}}\text{E}^{\text{X}}$. Sua representação gráfica pode ser vista na Figura 3.8 (b).
- **tarefas interativas:** são as tarefas em que o usuário interage com o sistema (o usuário ativa uma determinada função e o sistema processa). Por exemplo, fornecer uma *string* de consulta a um banco de dados. Sua representação gráfica pode ser vista na Figura 3.8 (c).
- **tarefas abstratas:** na verdade não são tarefas em si, mas uma generalização de uma série de tarefas complexas que devem ser decompostas em subtarefas (de um dos três tipos anteriores). Serve também para caracterizar tarefas que o projetista ainda não enquadró em uma das outras categorias previstas. Sua representação gráfica pode ser vista na Figura 3.8 (d).

Os modelos CTT possuem uma representação gráfica para cada tipo de tarefa e também para os operadores temporais que podem existir entre elas. Os operadores temporais no CTT permitem a descrição de como as tarefas se relacionam de acordo com o momento em que serão executadas, ou seja, sua sequência. Os operadores mostrados abaixo estão em ordem de precedência e são binários, ou seja, se aplicam a



Figura 3.8. Representações gráficas dos quatro tipos de tarefas no CTT.

duas tarefas. Para efeitos de exemplo, consideraremos duas tarefas genéricas, T1 e T2, aplicadas a cada operador temporal.

- **escolha:** indica que uma das duas tarefas (T1 ou T2) deve ser escolhida. Ao escolher uma tarefa, a outra fica indisponível. A representação gráfica desse operador pode ser vista na Figura 3.9 (a).
- **independência de ordem:** as duas tarefas (T1 e T2) podem ser executadas em qualquer ordem, mas quando uma inicia a outra só pode iniciar quando a primeira tiver terminado. A representação gráfica desse operador pode ser vista na Figura 3.9 (b).
- **concorrência independente:** as duas tarefas (T1 e T2) podem ser executadas concorrentemente, sem nenhuma restrição. A representação gráfica desse operador pode ser vista na Figura 3.9 (c).
- **concorrência com troca de informações:** as duas tarefas (T1 e T2) podem ser executadas em paralelo, mas deve haver uma sincronia entre elas para a troca de informações. A representação gráfica desse operador pode ser vista na Figura 3.9 (d).
- **desativação:** indica que a tarefa T2 é definitivamente desativada quando T1 é iniciada. A representação gráfica desse operador pode ser vista na Figura 3.9 (e).
- **suspende/continua:** indica que T1 pode ser interrompida quando T2 inicia. A representação gráfica desse operador pode ser vista na Figura 3.9 (f).
- **habilitação sequencial:** indica que, quando T1 termina, T2 é ativada. A representação gráfica desse operador pode ser vista na Figura 3.9 (g).
- **habilitação sequencial com passagem de informações:** indica que, quando T1 termina envia valores para T2 antes que ela seja ativada. A representação gráfica desse operador pode ser vista na Figura 3.9 (h).

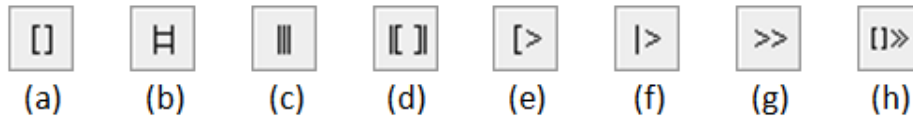


Figura 3.9. Representações gráficas dos operadores temporais binários do CTT.

Além dos operadores binários, o CTT prevê ainda operadores unários (que podem ser aplicados à uma única tarefa). Cada um deles também possui sua própria representação gráfica, como visto na Figura 3.10. Esses operadores são:

- **tarefa iterativa:** indica que uma tarefa pode ser repetida n vezes e só se torna indisponível quando for desabilitada por outra. Sua representação é mostrada na Figura 3.10 (a).
- **tarefa opcional:** indica que uma tarefa é opcional. Sua representação é mostrada na Figura 3.10 (b).
- **tarefa de conexão:** indica uma tarefa cooperativa, que pode ser usada em outro modelo participativo que pode envolver outros usuários. Sua representação é mostrada na Figura 3.10 (c).

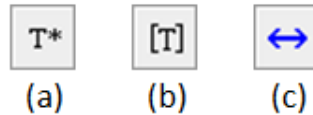


Figura 3.10. Representações gráficas dos operadores temporais unários do CTT.

Usando os quatro tipos de tarefas e os operadores mostrados acima, o projetista pode criar um modelo de sua aplicação. A Figura 3.11 mostra um modelo CTT para acesso a um caixa eletrônico, distribuído com a própria ferramenta. Se observarmos a tarefa (abstrata) de “Retirada de dinheiro”, notamos que a ação (interativa) “Selecionar opção de SAQUE” habilita a ação (do sistema) “Mostrar valores possíveis”. Essa por sua vez envia informações para a tarefa “Decide valor” (do usuário). O valor decidido é enviado para a ação “Seleciona valor” (interativa) e finalmente o valor informado pelo usuário é passado para a ação “Providencia dinheiro” (do sistema).

Os modelos CTT são construídos como uma hierarquia de tarefas onde o primeiro elemento é o mais abstrato e são representados por uma árvore, como visto na Figura 3.11.

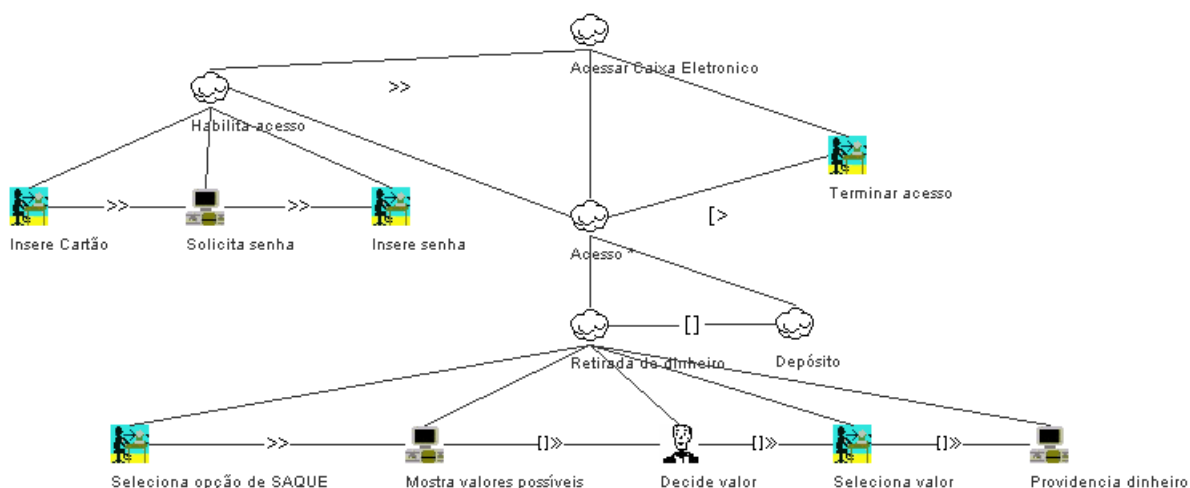


Figura 3.11. Exemplo de um modelo CTT para a operação de retirada de dinheiro em um caixa eletrônico.

Como já dissemos, a tarefa de modelagem no CTT é apoiada por uma ferramenta chamada *ConcurTaskTrees Environment* (CTTE) [Mori et al., 2002]. Esta ferramenta permite a modelagem de tarefas usando os conceitos do CTT e fornece um simulador onde o projetista pode ver a sequência de tarefas de acordo com os operadores temporais. Ao selecionar uma tarefa, o simulador mostra quais as tarefas que são habilitadas após a sua execução. O simulador mostrado na Figura 3.12, basicamente mostra ao usuário a(s) próxima(s) ação(ões) disponível(is). Por exemplo, a ação “Inserir senha”, após ser executada, habilita 3 ações possíveis (indicada na Figura 3.12 por setas vermelhas): “Selecionar opção de SAQUE”, “Depósito” e “Terminar acesso”. O CTTE mostra essas ações circundadas por uma caixa verde. Na área sinalizada com o número 1 da Figura 3.12, o projetista pode ver quais as ações estão habilitadas naquele momento e pode simular a sua execução dando um clique duplo sobre ela.

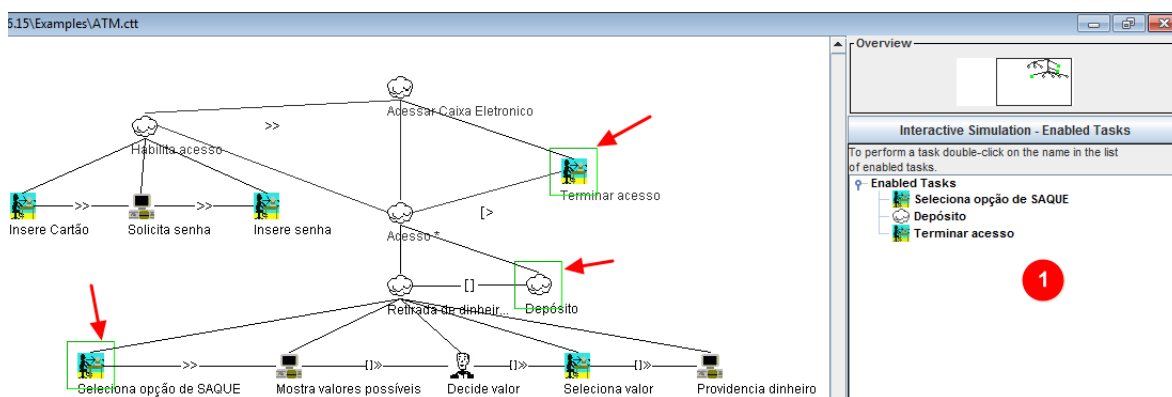


Figura 3.12. Fragmento da interface do CTTE mostrando o simulador em funcionamento.

3.2.2.3 *Human-centered Assessment and Modeling to Support Task Engineering for Resilient Systems - HAMSTERS*

Segundo Martinie et al. [2011], a tarefa de modelagem de sistemas complexos pode ser bastante custosa e complexa, pois os modelos gerados podem ser difíceis de ler e manter. Assim, Martinie et al. [2011] argumentam que é necessário construir diversos modelos em notações diferentes para que se possa ter uma visão mais completa do “mundo”. Neste caso, a complexidade está em manter todos os diferentes modelos consistentes. Diante desse problema, Martinie et al. [2011] propõem dois mecanismos para lidar com esta complexidade. O primeiro deles permite a decomposição de um modelo em vários submodelos. Estes submodelos podem ser reusados e, a cada modificação, todas as suas “cópias” são atualizadas. O outro mecanismo permite definir a comunicação entre os modelos.

Após apresentarem os dois mecanismos propostos para diminuir a complexidade de sistemas complexos, os autores mostram o caso de um modelo de tarefa de um sistema espacial na notação CTT. Este modelo possui muitas tarefas e muitas relações entre elas. Argumentando que modelos como esse podem se tornar grandes e difíceis de ler, Martinie et al. [2011] propõem uma nova notação apoiada por ferramenta, chamada *Human-centered Assessment and Modeling to Support Task Engineering for Resilient Systems* (HAMSTERS) e modelam o mesmo sistema espacial nesta notação. Os autores mostram como o modelo resultante ficou mais simples, sem perder sua semântica original. Essa notação foi projetada para ser compatível com a *ConcurTaskTrees* (CTT) e como tal, permite a caracterização de tarefas e a definição de relacionamentos temporais entre elas.

Em um trabalho posterior, Forbrig et al. [2014] identificaram que os dois mecanismos propostos por Martinie et al. [2011] não são suficientes para lidar com grandes sistemas reais. Assim, eles apresentam um terceiro mecanismo que permite o reuso de partes de modelos. Esse mecanismo é denominado componente genérico e permite o reuso de partes de modelos com um esforço menor. Assim, o projetista pode definir diferentes instâncias para diferentes modelos, mas sempre usando a definição genérica. Para mostrar as contribuições de sua proposta, Forbrig et al. [2014] modelam o mesmo sistema espacial anteriormente modelado Martinie et al. [2011] e argumentam que a sua proposta (do mecanismo de componentes genéricos) facilita a leitura e a qualidade dos modelos gerados, além de diminuir pequenos erros de modelagem.

3.2.3 Reflexões sobre os modelos existentes

Na seção de trabalhos relacionados deste capítulo, mostramos os principais modelos propostos para a modelagem de sistemas colaborativos. Na primeira subseção, apresentamos os modelos baseados na teoria da Engenharia Semiótica. Discutimos a seguir as principais limitações dos modelos apresentados que motivaram a construção de um novo - o SIGMa, proposto neste trabalho de doutorado.

O primeiro modelo apresentado foi o Modelo Abstrato do Componente Multi-usuário do Artefato de Metacomunicação (MetaCom-G) [Prates, 1998]. Como mostramos, este modelo já previa a necessidade de um gerador de cenários (naquela época chamado de Cenógrafo), mas o definiu apenas no nível conceitual (nunca foi implementado). O trabalho de Prates [1998] foi a maior fonte de inspiração para a construção do SIGMa. Como será visto no próximo capítulo, as dimensões da nossa linguagem de modelagem (SIGMa-dl) foram inspiradas pelas dimensões previstas pelo MetaCom-G, mas contemplam as características necessárias para a modelagem de sistemas colaborativos diversos. O MetaCom-G foi definido para a modelagem de sistemas colaborativos com foco no trabalho cooperativo, ou seja, sistemas nos quais diversos usuários interagem para realizar um determinado trabalho. Nosso objetivo neste trabalho é permitir a modelagem de sistemas colaborativos diversos, e não apenas aqueles que têm foco no trabalho, como uma rede social, por exemplo. Além disso, ao observarmos o MARq-G (instância do MetaCom-G), vemos que o seu foco e a sua arquitetura são voltadas para a representação de sistemas colaborativos voltados para o trabalho. Devido a essa característica, optamos por nos basearmos nas dimensões propostas por Prates [1998], mas com um olhar mais abrangente que permitisse a modelagem de sistemas colaborativos em geral. As dimensões que foram diretamente reaproveitadas do MetaCom-G são os papéis e os objetos (que a SIGMa-dl nomeia como artefatos). Além disso, a dimensão de Tempo x Espaço não foi usada como no trabalho de Prates [1998], mas foi fundamental para fomentar a definição da dimensão Espaço de Tempo, prevista pela SIGMa-dl.

Já a Manas [Barbosa, 2006], apesar de ter surgido com base nas deficiências do MetaCom-G na representação de fatores sociais como confiança, privacidade e polidez, tem como foco a comunicação entre usuários de um sistema colaborativo. Neste sentido, foca na emissão e recepção da mensagem que está sendo enviada entre os usuários através da interface. A definição de toda essa interação é modelada como uma comunicação. Embora ajude o projetista na visão do sistema como metacomunicação, é de difícil entendimento pois os conceitos (como papéis, ações e objetos) usados em sistemas colaborativos não são apresentados diretamente. Além disso, a Manas não permite

que o projetista vislumbre os cenários que podem ser gerados a partir da interação de cada papel/grupo presente no sistema.

Ainda nos trabalhos relacionados, mostramos os principais modelos propostos para a modelagem de sistemas colaborativos que não são baseados na Engenharia Semiótica. Começamos apresentando o *Groupware Task Analysis* (GTA) [Veer et al., 1996]. Esse *framework* evoluiu para uma ontologia, que ajuda os projetistas a refletir sobre uma situação problema (chamada de “modelo de tarefas 1”) e para formular uma proposta de melhoria para essa situação (chamada de “modelo de tarefas 2”). Apesar de existir uma ferramenta (EUTERPE [Welie et al., 1998]) baseada na ontologia que permite a criação de modelos de tarefa graficamente, o projetista deve definir cada instância prevista para as cinco dimensões da ontologia. Além disso, não leva em consideração o tempo de uso, uma vez que não descreve que decisões o usuário pode tomar em relação ao sistema, que altere aspectos sobre o sistema e a interação.

O outro trabalho descrito foi o *Concurrent Task Trees* (CTT) [Paternò, 1999; Paternò et al., 2001; Paternò, 2003]. Este trabalho tem como características principais a representação gráfica dos modelos e o suporte a relacionamentos temporais entre tarefas. Como mostramos, o CTT contempla uma série de operadores temporais que podem ser aplicados a duas tarefa e ainda operadores unários que podem ser aplicados a tarefas individuais. As tarefas são categorizadas em quatro tipos e permitem ao projetista identificar “quem” é responsável por cada tarefa (o usuário, o sistema ou ambos.) A criação dos modelos é apoiada por uma ferramenta denominada CTTE [Mori et al., 2002]. Além de permitir a criação dos modelos de forma visual, a CTTE fornece ao projetista ainda a capacidade de simular as ações previstas segundo os operadores temporais previstos. Sob um olhar superficial o CTT se parece bastante com o SIGMa. Por isso, cabe aqui descrever as suas principais limitações que o diferenciam da nossa proposta.

Ao usar o CTT, o projetista necessariamente tem que descrever todas as tarefas possíveis e os relacionamentos temporais entre elas. A partir do simulador, o projetista pode ver como as relações temporais entre as tarefas definidas se desenvolvem. Assim, o projetista pode ver quando uma tarefa é executada, que outras tarefas ficam disponíveis para o usuário. No entanto, ele apenas gera esta simulação das relações entre as tarefas descritas nos operadores temporais. Não há como o projetista descrever mudanças que afetarão não apenas o momento em que as tarefas estarão disponíveis, mas em todos os lugares

Como veremos no próximo capítulo, o SIGMa tem uma proposta diferente. Ele nasce com o objetivo de gerar os cenários que não foram explicitamente especificados pelo projetista, mas que podem ser gerados pelos usuários em tempo de uso a partir

de uma combinação de ações que foram descritas pelo projetista. É claro que algumas definições devem ser feitas ao usar o SIGMa, mas o projetista define apenas as consequências que cada ação traz ao sistema como um todo. Ele não define todas as interações possíveis entre elas. Neste sentido, o SIGMa tem o potencial de chamar a atenção do projetista para cenários que não foram antevistos em tempo de projeto, possibilitando que ele reflita sobre as decisões tomadas. No CTT, uma mesma ação que tem os mesmos impactos no sistema deve ser modelada duas vezes (ou copiada). Uma alteração que eventualmente possa ser necessária deve ser feita em todas as cópias daquela ação, ou seja, a alteração em uma instância da ação não é propagada para as suas cópias. Finalmente, o CTT não prevê a “divisão” de um sistema colaborativo em diferentes contextos de aplicação. Por exemplo, um sistema de gerência de conferências passa por diversos contextos diferentes (fase de submissão, fase de atribuição de revisores, fase de revisão, fase de decisão de artigos aceitos e rejeitados, etc). Cada contexto do sistema envolve pessoas, ações e objetos diferentes. Como o CTT não contempla essa segmentação conceitual do sistema, os modelos que envolvem diferentes contextos podem ser difíceis de serem criados e ainda mais de serem simulados. Como veremos no próximo capítulo, os contextos pelos quais uma aplicação pode passar, os quais denominamos “espaços de tempo”, são o cerne da nossa linguagem de modelagem. Dessa forma, o projetista pode modelar a sua aplicação pensando nesses diferentes espaços de tempo, e pode acompanhar os impactos das ações previstas entre eles.

Por fim, o *Human-centered Assessment and Modeling to Support Task Engineering for Resilient Systems* (HAMSTERS) foi concebido para ser aderente ao CTT e, mesmo apresentando uma proposta para simplificar os modelos e propagar as alterações feitas em uma tarefa, ainda exige que o projetista especifique todos os cenários possíveis, e assim, embora trate de algumas limitações do CTT, não resolve as relacionadas a geração de cenários futuros que pretendemos tratar no SIGMa.

Capítulo 4

SIGMa: Modelo proposto

Neste capítulo, detalhamos o modelo que estamos propondo. Este modelo é composto por uma linguagem de modelagem denominada SIGMa-dl e pelo SIGN (ScenarIo GeNerator). Em seguida mostramos como funciona o algoritmo que implementa o SIGN e descrevemos o protótipo que o implementa.

Nossa pesquisa caminha numa direção diferente dos outros modelos presentes na literatura, como os apresentados no Capítulo 3. Nós consideramos o tempo como fundamental para a compreensão da evolução das ações disponibilizadas para os usuários e seus impactos. Ao analisar os possíveis cenários ao longo do tempo, o projetista pode refletir sobre o modelo feito e esta reflexão pode auxiliá-lo a tomar decisões sobre alterações que eventualmente devem ser feitas para evitar potenciais problemas. Para que isso seja possível, propomos o modelo descrito a seguir.

Como visto na subseção 3.2.1.1, o gerador de cenários já fazia parte do modelo proposto por Prates [1998]. Contudo, naquele trabalho ele foi pensando apenas conceitualmente e não chegou a ser desenvolvido. Nosso trabalho foi inspirado nesta ideia. O modelo que estamos propondo incorpora e implementa o gerador de cenários, daqui pra frente chamado de SIGN (ScenarIo GeNerator) mas a sua linguagem de modelagem própria não tem uma ligação direta com o MetaCom-G ou com suas derivações.

A Figura 4.1 ilustra o processo de uso do modelo proposto. Inicialmente o projetista faz uso da SIGMa-dl para modelar um sistema colaborativo de interesse. Em seguida, o modelo gerado por esta linguagem é então submetido ao SIGN, que gera os cenários correspondentes à modelagem feita. Dessa forma, o projetista pode analisar os cenários gerados e pode verificar se algum deles está criando situações potencialmente indesejadas. Neste caso, o projetista tem a opção de corrigir a modelagem e gerar novamente os cenários correspondentes. Esse ciclo termina quando o projetista se der por satisfeito sobre os cenários que pretende disponibilizar ao usuário final.

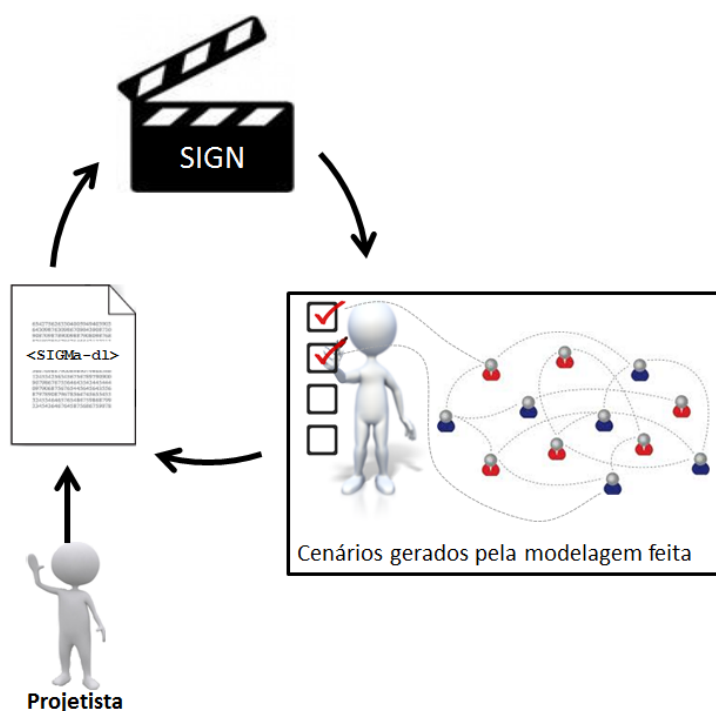


Figura 4.1. Modelo proposto.

As dimensões contempladas pela SIGMA-dl referem-se a dois aspectos do sistema de colaboração: sua estrutura (papéis, grupos, artefatos e a quem pertencem) e seu comportamento (definição dos tempos e das ações que podem ser executadas por cada papel ou grupo e suas consequências). Uma vez que o projetista tenha descrito o modelo conceitual do sistema, o SIGN gera então o conjunto de cenários que potencialmente podem se concretizar em tempo de uso, para que o projetista possa inspecionar.

Antes de apresentarmos mais detalhadamente a linguagem proposta, considere uma situação hipotética onde um projetista deseja fazer a modelagem de um sistema de gestão de conferência. Neste sistema, existem papéis diferentes (autores, coordenador de programa e revisores). Cada um desses papéis pode executar diferentes ações em momentos específicos de tempo. Neste caso o tempo não é o tempo cronológico, mas sim, o contexto (ou espaço de tempo) que descreve as possíveis ações em determinado momento. Por exemplo, um espaço de tempo começa quando a submissão é aberta. Neste contexto autores podem incluir, editar e excluir itens (no caso, artigo ou anexos). Neste momento, os revisores só podem definir os temas em que atuam. Quando o prazo de submissão termina há uma mudança de contexto (o espaço de tempo 1 termina e o espaço de tempo 2 começa). Neste espaço de tempo 2, as ações disponíveis para cada função são alteradas: autores não podem mais enviar ou editar uma submissão existente e o coordenador de programa pode atribuir revisores. Quando a atribuição

dos revisores é feita há outra mudança no contexto (o espaço de tempo 2 termina e o espaço de tempo 3 começa). No espaço de tempo 3, revisores terão acesso a submissões para revisão, e eles podem registrar seus comentários no sistema. Em seguida, inicia-se então o tempo de decisão (o espaço de tempo 3 termina e o espaço de tempo 4 começa), onde revisores podem discutir suas revisões e os coordenadores tomam a decisão sobre que artigos aceitar ou não. Finalmente, ao terminar o processo de revisão (o espaço de tempo 4 termina e o espaço de tempo 5 começa), os coordenadores notificam os autores e estes podem ver as revisões de seus artigos e os autores de artigos aceitos podem submeter as suas versões finais.

4.1 Linguagem de Modelagem Proposta - *SIGMa-dl*

A seguir descrevemos as dimensões previstas pela SIGMa-dl para a modelagem de um sistema colaborativo, ou em outras palavras os elementos lexicais da linguagem de modelagem [Pereira Junior & Prates, 2015]. Como discutido na seção anterior, estes elementos foram baseados na literatura da área de sistemas colaborativos e nos resultados do nosso estudo de caso. A linguagem é composta por nove elementos lexicais (ou dimensões): espaços de tempo, papéis, grupos, artefatos atômicos e compostos, ações, posses, mudanças e relações. Para facilitar o entendimento das dimensões propostas, para cada uma delas descrevemos a seguir sua sintaxe e fornecemos um exemplo simples de como usá-las. Sempre que possível, usaremos exemplos de um sistema de gerência de conferências, como o apresentado anteriormente. Vale ressaltar que todos os campos previstos na sintaxe de cada dimensão são **obrigatórios**, salvo quando indicado o contrário.

4.1.1 Espaços de tempo

A dimensão espaço de tempo contempla os contextos de execução do sistema que está sendo modelado. Como mostramos no exemplo do sistema de gerência de conferências, aplicações colaborativas podem passar por diversos contextos diferentes, os quais chamamos espaço de tempo. Diferente dos outros modelos, ao usar o SIGMa o projetista é capaz de modelar esses diferentes espaços de tempo. Vale ressaltar que o tempo considerado pelo SIGMa representa um período em que um determinado contexto da aplicação permanece o mesmo (e não o tempo cronológico).

Apesar de o projetista poder definir todos os possíveis valores para cada uma das dimensões previstas pela SIGMA-dl, algumas instâncias podem fazer sentido em um espaço de tempo e outras não. Uma ação pode existir em um tempo e pode não existir em outro. O mesmo vale para as outras dimensões da SIGMA-dl, explicadas a seguir. Assim, cada espaço de tempo é caracterizado por subconjuntos dos papéis, grupos, artefatos, ações, posses, mudanças e relações já definidos. Após definir o seu modelo em termos das dimensões que serão apresentadas nesta seção, o projetista deve indicar quais instâncias de cada dimensão são válidas para cada espaço de tempo¹. A definição de espaço de tempo é abstrata, ficando a cargo do projetista definir quando faz sentido criar um espaço de tempo, mas em geral recomenda-se a definição de um novo espaço de tempo quando há uma ou mais mudanças significativas nos papéis ou grupos, ou nas ações que eles podem executar. Quando o projetista define mais de um espaço de tempo na aplicação que está modelando, ele deve necessariamente informar qual ação irá causar a mudança de tempo, para garantir a consistência de seu modelo.

A sintaxe de definição de espaços de tempo é mostrada na Tabela 4.1 e um exemplo de uso na Tabela 4.2.

Tabela 4.1. Sintaxe de definição da dimensão Espaços de tempo.

| Espaços de Tempo | |
|---------------------|--|
| Parâmetro | Definição |
| Identificador único | Identificador único para um novo espaço de tempo sendo criado. |
| Descrição textual | Registro das decisões de projeto para o novo espaço de tempo no sistema. |

Tabela 4.2. Exemplos de definição de instância da dimensão Espaços de tempo.

| Exemplo de uso - Espaços de Tempo | |
|-----------------------------------|--|
| Identificador único | Descrição textual |
| Submissão | Etapa em que autores podem submeter seus artigos |
| Atribuição_de_Revisiones | Etapa em que revisores são selecionados para a revisão dos artigos |
| Revisão | Etapa em que revisores selecionados revisam e dão seu parecer quanto à aceitação ou não de um artigo |

¹Mais detalhes sobre essa caracterização dos espaços de tempo serão apresentados no fim dessa seção.

4.1.2 Papéis

Em um sistema colaborativo pessoas podem assumir diferentes funções durante o seu uso. Na nossa linguagem de modelagem estas funções são representadas pelos `papéis`. Embora seja frequente a ocorrência de hierarquia em sistemas colaborativos, nossa linguagem não prevê esta definição explícita. No nosso modelo, a hierarquia é tratada no protocolo social.

A sintaxe de definição de papéis é mostrada na Tabela 4.3 e um exemplo de uso na Tabela 4.4.

Tabela 4.3. Sintaxe de definição da dimensão `Papéis`.

| Papéis | |
|---------------------|--|
| Parâmetro | Definição |
| Identificador único | Identificador único para um novo papel sendo criado. |
| Descrição textual | Registro das decisões de projeto para o novo papel no sistema. |

Tabela 4.4. Exemplos de definição de instâncias da dimensão `Papéis`.

| Exemplo de uso - Papéis | |
|-------------------------|--|
| Identificador único | Descrição textual |
| Autor | Autor de artigos da conferência |
| Coordenador | Responsável por coordenar a conferência |
| Revisor | Pessoa responsável por revisar os artigos submetidos |

4.1.3 Grupos

Após determinar os papéis, opcionalmente o projetista pode definir diferentes grupos que poderão existir e atuar no contexto do sistema. O objetivo do grupo é permitir que o projetista defina comportamentos específicos para um subconjunto de papéis do sistema. Os grupos então são descritos pelos membros que os compõem.

Ao informar os papéis que estarão presentes em um grupo, o projetista deve definir o seu escopo, como visto na Tabela 4.5. Para o escopo há duas possibilidades: ou todos os indivíduos que assumirem aquele papel farão parte do grupo, ou apenas uma parte deles fará. Para fazer esta diferenciação, o projetista deve informar o escopo de cada papel a ser inserido no grupo. Os valores possíveis para o escopo são

“**SubGrupo**” - caso em que nem todos os indivíduos que assumirem o papel farão necessariamente parte do grupo, mas apenas parte deles - ou “**Todos**” - caso em que todos os indivíduos que assumirem o papel farão necessariamente parte do grupo.

Além de poder modelar apenas parte ou todo os indivíduos de um papel como integrantes de um grupo, é possível ainda que o projetista crie grupos com diferentes papéis (com seu devido escopo). Assim, é possível criar grupos homogêneos (com apenas um papel) ou heterogêneos (com diferentes papéis).

A sintaxe de definição de grupos é mostrada na Tabela 4.5 e um exemplo de uso na Tabela 4.6.

Tabela 4.5. Sintaxe de definição da dimensão Grupos.

| Grupos | |
|---------------------|--|
| Parâmetro | Definição |
| Identificador único | Identificador único para um novo grupo sendo criado. |
| Descrição textual | Registro das decisões de projeto para o novo grupo no sistema. |
| Lista de Papéis | (<papel>,<escopo>) |

Tabela 4.6. Exemplos de definição de instâncias da dimensão Grupos.

| Exemplo de uso - Grupos | | |
|-------------------------|--|--|
| Identificador único | Descrição textual | Lista de Papéis |
| Autores_Aceitos | Grupo que contém os autores com artigos aceitos | ((Autor, Subgrupo)) |
| Autores_Rejeitados | Grupo que contém os autores com artigos rejeitados | ((Autor, Subgrupo)) |
| Comite_Programa | Grupo que forma o comitê de programa | ((Coordenador,Todos), (Revisor, SubGrupo)) |

4.1.4 Artefatos

Ao utilizar um sistema colaborativo qualquer, normalmente papéis e/ou grupos atuam sobre diversos objetos, documentos, planilhas, etc. Nossa linguagem permite a modelagem destes elementos, os quais denominamos artefatos. Além de poder definir um artefato, em determinadas situações pode ser necessário definir atributos para um determinado artefato. Por isso, caracterizamos os artefatos como atômicos ou compostos. Os artefatos atômicos são indivisíveis, ou seja,

são elementos únicos que não possuem subelementos ou atributos. Já os artefatos compostos possuem características (ou atributos) que o integram. Eles são compostos por artefatos atômicos ou por outros artefatos compostos.

A sintaxe de definição de artefatos atômicos é mostrada na Tabela 4.7 e um exemplo de uso na Tabela 4.8.

Tabela 4.7. Sintaxe de definição da dimensão Artefatos Atômicos.

| Artefatos Atômicos | |
|---------------------|---|
| Parâmetro | Definição |
| Identificador único | Identificador único para um novo artefato atômico sendo criado. |
| Descrição textual | Registro das decisões de projeto para o novo artefato atômico no sistema. |

Tabela 4.8. Exemplos de definição de instâncias da dimensão Artefatos Atômicos.

| Exemplo de uso - Artefatos Atômicos | |
|-------------------------------------|---------------------------------|
| Identificador único | Descrição textual |
| Título_Artigo | Título do artigo submetido |
| Nome_Autor | Nome do autor do artigo |
| PDF | Arquivo PDF do artigo submetido |

Já para definir um artefato composto, o projetista deve informar, além do identificador único e da descrição textual, quais os outros artefatos (atômicos ou compostos) que integram o artefato que está sendo definido.

A sintaxe de definição de artefatos atômicos é mostrada na Tabela 4.9 e um exemplo de uso na Tabela 4.10

Tabela 4.9. Sintaxe de definição da dimensão Artefatos Compostos.

| Artefatos Compostos | |
|---------------------|--|
| Parâmetro | Definição |
| Identificador único | Identificador único para um novo artefato composto sendo criado. |
| Descrição textual | Registro das decisões de projeto para o novo artefato composto no sistema. |
| Lista de Artefatos | (<artefato >) |

Tabela 4.10. Exemplo de definição de uma instância da dimensão Artefatos Compostos.

| Exemplo de uso - Artefatos Compostos | | |
|--------------------------------------|---|----------------------------------|
| Identificador único | Descrição textual | Lista de Artefatos |
| Submissão | Submissão de um artigo em uma conferência | (Título_Artigo, Nome_Autor, PDF) |

4.1.5 Posses

Em todo sistema onde há papéis/grupos e artefatos, é comum existir uma relação de posse entre eles. Assim, é possível que o projetista modele a posse de determinados artefatos (atômicos ou compostos) por papéis e grupos. A definição da posse deve ser feita para cada papel/grupo e artefato que for possuído por eles. Cada artefato pode ser possuído por um ou mais papéis ou grupos.

Nenhum papel ou grupo detém a posse de um artefato, a menos que esta posse seja explicitamente modelada. Assim, se no contexto do sistema um papel ou grupo não detém a posse de nenhum artefato, basta que o projetista não faça nenhuma definição de posse para ele.

A sintaxe de definição de posses é mostrada na Tabela 4.11 e um exemplo de uso na Tabela 4.12.

Tabela 4.11. Sintaxe de definição da dimensão Posses.

| Posses | |
|--|--|
| Parâmetro | Definição |
| Identificador único | Identificador único para uma nova posse sendo criada. |
| Descrição textual | Registro das decisões de projeto para a nova posse no sistema. |
| Artefato (atômico ou composto) | <artefato> |
| Lista de quem terá a posse do artefato | (<papel>ou <grupo>) |

4.1.6 Ações

As ações disponíveis em um sistema colaborativo caracterizam as possibilidades de interação que o projetista irá permitir aos papéis, grupos ou mesmo ao sistema.

Tabela 4.12. Exemplo de definição de uma instância da dimensão *Posses*.

| Exemplo de uso - <i>Posses</i> | | | |
|--------------------------------|--------------------------------------|-----------|-----------------|
| Identificador único | Descrição textual | Artefato | Quem terá posse |
| Autor_Submissão | O autor tem a posse da sua submissão | Submissão | (Autor) |

A sintaxe de definição de ações é mostrada na Tabela 4.13 e um exemplo de uso na Tabela 4.14.

Tabela 4.13. Sintaxe de definição da dimensão *Ações*.

| Ações | |
|---------------------------------|---|
| Parâmetro | Definição |
| Identificador único | Identificador único para uma nova ação sendo criada. |
| Descrição textual | Registro das decisões de projeto para a nova ação no sistema. |
| Lista de quem pode fazer a ação | (<papel>, <grupo>ou <sistema>) |

Tabela 4.14. Exemplos de definição de instâncias da dimensão *Ações*.

| Exemplo de uso - <i>Ações</i> | | |
|-------------------------------|--|----------------------|
| Identificador único | Descrição textual | Quem (Agente) |
| Submete_Artigo | Ação de submeter um artigo | (Autor) |
| Acessa_Submissão | Ação de acessar uma submissão feita | (Autor, Coordenador) |
| Abrir_Conferencia | Ação de abrir oficialmente uma conferência | (Coordenador) |
| Encerra_Fase_Submissão | Ação automática de encerrar a fase de submissão de artigos de acordo com uma data pré-definida | (Sistema) |

Cabe aqui uma reflexão sobre a possibilidade de o sistema poder executar ações. Em nossos estudos, percebemos que o sistema em si por vezes executa ações durante o seu funcionamento. Estas ações, no entanto, não são independentes, ou seja, a SIGMa-dl considera que o sistema não toma a decisão de qual ação executar. Assim, as ações permitidas ao sistema, na nossa proposta, só são possíveis como consequência da ação de um papel ou grupo. Como exemplo, se observarmos um sistema de gerência de conferências, o próprio sistema encerra automaticamente o período de submissão. No

entanto, isso só é possível após o coordenador ou responsável pelo evento ter configurado a data e hora em que isso acontece.

Vale ressaltar que, a dimensão *ações* serve apenas para nomear e identificar as ações previstas para o sistema. No momento da definição de uma ação, o projetista não define os passos algorítmicos (dados de entrada e saída, linhas de comando, etc.) que uma ação terá dentro do sistema. Tampouco define quais os artefatos que serão impactados pela ação. A definição da ação é apenas conceitual. Serve somente para dar ao projetista a possibilidade de descrever as funcionalidades pensadas para o sistema que está sendo modelado. O impacto desta ação e mesmo os artefatos que serão impactados por ela são definidos numa fase posterior de modelagem, nas dimensões *relações* e *mudanças*, explicadas a seguir. Isso se justifica pois uma mesma ação pode trazer comportamentos diferentes ou mesmo pode afetar diferentes artefatos, dependendo do contexto do sistema.

4.1.7 Mudanças

Até este momento, as definições que podem ser feitas a partir das dimensões descritas são desconexas e não produzem nenhum efeito prático. Definir uma ação sem consequências não tem nenhuma utilidade, assim como definir que um papel detém a posse de um artefato, por si só, traz pouco ganho para o projetista. Assim é necessário ter uma dimensão onde o projetista pode descrever o que muda no sistema. Chamamos esta dimensão de *mudanças*.

Um sistema passa por diversas mudanças durante sua execução. Assim é essencial que o projetista consiga modelar as mudanças que acontecem no sistema ao longo do tempo. É ainda mais importante para que ele possa verificar se a mudança que ele previu está causando consequências indesejadas para o usuário final quando inspecionar os cenários gerados pelo SIGN. Uma mudança nada mais é que uma troca de contextos dentro do sistema. Nossa linguagem prevê a modelagem de quatro tipos diferentes de mudanças que podem ocorrer em um sistema colaborativo: mudanças de grupo ou papel, mudanças de espaço de tempo, mudanças de posse de um artefato e mudanças no conjunto de ações, conforme descrito a seguir:

4.1.7.1 Mudança no conjunto de ações

Indica as mudanças no conjunto de ações disponíveis ao longo do tempo. As ações disponíveis para cada papel ou grupo podem variar dependendo do contexto atual do sistema. Além disso, uma ação pode fazer sentido em um determinado contexto e não em outro. Assim, é necessário que o projetista especifique os conjuntos de ações disponíveis em cada contexto do sistema. Novamente ao modelar esta mudança no conjunto de ações, o projetista não especifica qual ação gera esta mudança. Ele apenas diz o que muda no sistema quando esta mudança que está sendo modelada acontecer.

A sintaxe de definição de mudanças no conjunto de ações é mostrada na Tabela 4.15 e um exemplo de uso na Tabela 4.16.

Tabela 4.15. Sintaxe de definição da dimensão Mudanças no conjunto de ações.

| Mudança no Conjunto das Ações | |
|---|---|
| Parâmetro | Definição |
| Identificador único | Identificador único para uma nova mudança no conjunto de ações sendo criada. |
| Descrição textual | Registro das decisões de projeto para a nova mudança no conjunto de ações no sistema. |
| Artefato (atômico ou composto) | <artefato> |
| Lista de ações que poderão ser desempenhadas por papéis ou grupos | (<papel>ou <grupo>, <ação>) |

Tabela 4.16. Exemplos de definição de instâncias da dimensão Mudanças no conjunto de ações.

| Exemplo de uso - Mudança no Conjunto das Ações | | | |
|--|--|-----------|---|
| Identificador único | Descrição textual | Artefato | Quem (Agentes) |
| Mud_Apos_Abrir_Conferencia | Estado do sistema após a abertura da conferência | Submissão | ((Autor, Faz_Submissão), (Coordenador, Configura_Fim_Submissão)) |
| Mud_Apos_Submissão | Estado do sistema após uma submissão | Submissão | ((Autor, Ver_Submissão), (Autor, Editar_Submissão), (Autor, Excluir_Submissão), (Coordenador, Ver_Submissão)) |

No exemplo da Tabela 4.16, quando a mudança “*Mud_Apos_Abrir_Conferencia*” acontecer, o autor poderá executar a ação “*Faz_Submissão*” e o coordenador poderá executar a ação “*Configura_Fim_Submissão*”. Vale ressaltar que não foi definida ainda neste momento a ação que irá gerar esta mudança. Este passo é feito na dimensão Relações, explicada à frente.

4.1.7.2 Mudança nos espaços de tempo

Como já dissemos, em um sistema colaborativo é comum que existam diferentes espaços de tempo, ou contextos de execução, onde determinados artefatos, papéis, grupos e suas ações estão disponíveis. Este tipo de mudança indica o término de um espaço de tempo (ou contexto) e o início de outro espaço de tempo. A sintaxe de definição de mudanças no espaço de tempo é mostrada na Tabela 4.17 e um exemplo de uso na Tabela 4.18.

Tabela 4.17. Sintaxe de definição da dimensão Mudanças nos espaços de tempo.

| Mudança nos Espaços de Tempo | |
|------------------------------|---|
| Parâmetro | Definição |
| Identificador único | Identificador único para uma nova mudança nos espaços de tempo sendo criada. |
| Descrição textual | Registro das decisões de projeto para a nova mudança nos espaços de tempo no sistema. |
| Espaço de tempo atual | <espaco_de_tempo1> |
| Novo espaço de tempo | <espaco_de_tempo2> |

Tabela 4.18. Exemplos de definição de instâncias da dimensão Mudanças nos espaços de tempo.

| Exemplo de uso - Mudança nos Espaços de Tempo | | | |
|---|--|-------------------------|-------------------------|
| Identificador único | Descrição textual | Espaço de Tempo Atual | Novo Espaço de Tempo |
| Mud_Termino_Submissão | Termina o tempo de submissão e inicia o tempo de atribuição de revisores | Submissão | Atribuição_de_Revisores |
| Mud_Termino_Atrib_Revisores | Termina o tempo de atribuição de revisores e inicia o tempo de revisão | Atribuição_de_Revisores | Revisão |

4.1.7.3 Mudança na posse dos artefatos

Indica as mudanças na posse dos artefatos. Ao longo do uso do sistema, um artefato pode ser cedido de um papel/grupo para outro. Assim, nosso modelo prevê um tipo de mudança para contemplar estes casos.

A sintaxe de definição de mudanças de posse é mostrada na Tabela 4.19 e um exemplo de uso na Tabela 4.20.

Tabela 4.19. Sintaxe de definição da dimensão Mudanças de posse.

| Mudança nas posses | |
|---------------------------|---|
| Parâmetro | Definição |
| Identificador único | Identificador único para uma nova mudança nas posses sendo criada. |
| Descrição textual | Registro das decisões de projeto para a nova mudança nas posses no sistema. |
| Quem irá ceder a posse | <papel>ou <grupo> |
| Qual artefato será cedido | <artefato> |
| Quem receberá a posse | <papel>ou <grupo> |

4.1.7.4 Mudança de papel ou grupo

Indica as mudanças que um papel ou grupo pode sofrer. Um papel pode deixar de existir em um determinado momento ou os indivíduos que tinham aquele papel podem assumir outro papel. O mesmo pode acontecer com os grupos.

A sintaxe de definição de mudanças de papel/grupo é mostrada na Tabela 4.21. Como o sistema de gerência de conferências não possui um exemplo de mudança de papel/grupo evidente, apresentamos um outro exemplo de um ambiente virtual de aprendizagem para ilustrar o uso desse tipo de mudança, na Tabela 4.22.

As mudanças em nosso modelo são desacopladas das ações que as geram. Isso se justifica pelo fato de que uma ação potencialmente pode gerar diversas mudanças. E uma mesma mudança potencialmente poderia ser causada por diversas ações

Tabela 4.20. Exemplo de definição de uma instância da dimensão Mudanças de posse.

| Exemplo de uso - Mudança nas posses | | | | |
|-------------------------------------|--|---------|-------------------------------|-------------|
| Identificador único | Descrição textual | Cedente | Artefato | Recebedor |
| Mud_Cede_Documento | Autor cede um documento para o coordenador | Autor | Contrato_de_Direitos_Autorais | Coordenador |

Tabela 4.21. Sintaxe de definição da dimensão Mudanças de papel/grupo.

| Mudança de papel/grupo | |
|------------------------|---|
| Parâmetro | Definição |
| Identificador único | Identificador único para uma nova mudança de papel/grupo sendo criada. |
| Descrição textual | Registro das decisões de projeto para a nova mudança de papel/grupo no sistema. |
| Papel/Grupo atual | <papel>ou <grupo> |
| Novo Papel/Grupo | <papel>ou <grupo> |

Tabela 4.22. Exemplo de definição de uma instância da dimensão Mudanças de papel/grupo em um ambiente virtual de aprendizagem.

| Exemplo de uso - Mudança de Papel/Grupo | | | |
|---|--|-------------------|------------------|
| Identificador único | Descrição textual | Papel/Grupo atual | Novo Papel/Grupo |
| Mud_Ex_Aluno | Após o término de um semestre, o aluno vira ex-aluno daquela disciplina no sistema | Aluno | Ex-aluno |

diferentes. Assim, ao definir uma mudança, o projetista não define a ação que a gera, mas apenas diz o que muda de acordo com o tipo de mudança. Em outro passo, o projetista deve então relacionar as ações com as mudanças que causam, como apresentado a seguir.

4.1.8 Relações

Até então, com as dimensões já descritas acima, não é possível que o projetista efetivamente modele o comportamento dos papéis ou grupos e nem as consequências de cada ação dentro do sistema. Isso é possível através da dimensão que denominamos relações. As relações fazem o elo entre uma ação e as mudanças que ela gera. Assim, para indicar quais as mudanças que serão disparadas quando uma ação for executada, o projetista deve definir as relações entre elas.

Além de uma ação poder gerar diversas mudanças no sistema, frequentemente elas estão sujeitas a condições diversas. Por exemplo, em um sistema acadêmico uma determinada ação (por exemplo lançar notas) só pode acontecer (e gerar suas mudanças) caso uma ou mais condições sejam satisfeitas (por exemplo a abertura do diário pela secretaria). Quando este é o caso a ser modelado, cada condição da qual dependem as mudanças deve ser descrita. Vale ressaltar que o modelo não estabelece

os tipos de condição que podem ser modeladas, até porque elas podem variar muito de um sistema para outro. Assim, esta condição não é processada pelo gerador de cenários, mas serve apenas como registro de que as mudanças atreladas a uma ação apenas acontecerão se esta condição for atendida. Como o SIGN não processa as condições descritas e como em determinadas situações a execução de uma ação independe de qualquer condição, este campo é opcional e só deve ser preenchido caso faça sentido.

Em seguida, o projetista deve indicar a ação que será envolvida na relação e o seu respectivo ator (papel, grupo ou sistema). A ação envolvida na relação, no entanto, pode ser **"dependente"** ou **"independente"**. A ação é independente quando não depende de outra ação para acontecer. E é dependente quando depende de uma ação anterior para acontecer. Por exemplo, a ação "Abrir_Conferência" realizada pelo coordenador é independente de qualquer outra, isto é, essa ação não depende de outra para ser realizada. Para modelar esta ação, o projetista deve definir o parâmetro "ação independente" **"Sim"**. Já a ação "Submeter_Artigo" realizada pelo autor só pode ser feita após a ação "Abrir_Conferência" realizada pelo coordenador. Assim, a ação "Autor.Submete_Artigo" é dependente da ação "Coordenador.Abrir_Conferência". Ao modelar este tipo de ação, o parâmetro de ação independente deve ser definido como **"Não"**.

Como já foi discutido na dimensão ações, papéis, grupos ou mesmo o sistema podem executar ações. O projetista deve definir a ação envolvida na relação e quem é responsável por executá-la.

Até então as ações e artefatos já definidos não possuem nenhuma ligação direta, ou seja, até então não foi possível modelar o fato de que uma ação acontece sobre um artefato. Isso também é possível ao definir uma relação. A ação envolvida na relação acontecerá sobre um artefato, e o projetista deve indicar qual deles será afetado por ela.

Depois de definir esses primeiros campos da relação, o projetista deve indicar as mudanças que acontecerão no sistema quando a ação envolvida na relação for executada. Vale ressaltar que a ação envolvida em uma relação pode causar várias mudanças (no conjunto de ações, de espaços de tempo, de papel ou grupo e de posse) no sistema. Cabe ao projetista indicar cada uma delas na definição de uma relação.

A sintaxe de definição de relações é mostrada na Tabela 4.23. Devido à quantidade de campos previstos pela sintaxe das relações, para facilitar a leitura invertemos os campos e os respectivos valores da Tabela 4.24, que mostra exemplos de duas relações. Na relação do "Exemplo1", a ação "Abrir_Conferencia", de-

Tabela 4.23. Sintaxe de definição da dimensão Relações.

| Relações | |
|--|---|
| Parâmetro | Definição |
| Identificador único | Identificador único para uma nova relação sendo criada. |
| Descrição textual | Registro das decisões de projeto para a nova relação no sistema. |
| Condição da Relação (<i>parâmetro opcional</i>) | condição que deve ser atendida para que a ação envolvida na relação possa ser executada. |
| Ação envolvida na relação | (<ação>,<papel>ou <ação>,<grupo>ou <ação>,<sistema>) |
| Ação da relação é independente? | Sim caso a ação não dependa de nenhuma outra para ser executada ou Não caso dependa. |
| Artefato afetado pela ação. | <artefato> |
| Lista de Mudanças | (<lista_mudanças_conjunto_acoes>,<lista_mudanças_espaco_tempo>,<lista_mudanças_posse>,<lista_mudanças_papel_grupo>) |

sempenhada pelo “Coordenador” e que envolve o artefato “Submissão” pode ser desempenhada independente de outras ações e, quando executada, gera a mudança “Mud_Apos_Abrir_Conferencia”. Da mesma forma, na relação do “Exemplo2”, a ação “Submete_Artigo”, desempenhada pelo “Autor” e que envolve o artefato “Submissão” é dependente de outras ações e, quando executada, gera a mudança “Mud_Apos_Submissão”. Ambas as relações não possuem condições, por isso a Tabela 4.24 mostra este campo como “Nula”.

Se voltarmos à Tabela 4.16, podemos ver que a mudança “Mud_Apos_Abrir_Conferencia” tem como consequências duas ações: “Autor.Faz_Submissão” e “Coordenador.Configura_Fim_Submissão”. Assim, quando a ação envolvida na relação do Exemplo1 for executada, irá permitir que essas ações sejam executadas no sistema. Uma das ações consequentes, a “Autor.Faz_Submissão” por acaso é a ação envolvida na relação do Exemplo2. Esta por sua vez, quando executada, gera a mudança “Mud_Apos_Submissão”, que tem suas próprias consequências. A Figura 4.2 mostra os cenários que seriam gerados pelo SIGN após processar as duas relações definidas no exemplo da Tabela 4.24. As ações que podem ser executadas pelo papel “Autor” estão pontilhadas para diferenciá-las as ações que podem ser executadas pelo papel “Coordenador”.

O exemplo ilustrado na Figura 4.2 é simples e tem o objetivo de mostrar o funcio-

Tabela 4.24. Exemplos de definição de instâncias da dimensão *Relações*. As duas relações de exemplo não possuem condições, por isso este campo foi definido para fins ilustrativos como “Nula”

| Exemplo de uso - Relações | | |
|---------------------------------|---|---|
| Parâmetro da Sintaxe | Exemplo1 | Exemplo2 |
| Identificador único | Rel_Abrir_Conferencia | Rel_Apos_Submissao |
| Descrição textual | Relação que envolve a ação de abrir a conferência | Relação que envolve a ação de submeter um artigo para a conferência |
| Condição da relação | Nula* | Nula* |
| Ação envolvida na relação | Coordenador. Abrir_Conferencia | Autor.Submete_Artigo |
| Ação da relação é independente? | Sim | Não |
| Artefato afetado pela ação | Submissão | Submissão |
| Lista de Mudanças | (Mud_Apos_Abrir_Conferencia) | (Mud_Apos_Submissão) |

namento do SIGN, detalhado na próxima subseção. No entanto, se considerarmos que na modelagem completa a ação “Autor.Faz_Submissao” poderia aparecer em outros cenários (como por exemplo após uma etapa de *rebuttal*), as consequências dessa ação seriam mostradas também para estes outros cenários. E o projetista não tem que fazer nenhuma outra definição para que isso aconteça. É neste sentido que dizemos que o projetista não precisa descrever todos os caminhos possíveis. Ao usar o CTT, por exemplo, o projetista teria que modelar as consequências da ação “Autor.Faz_Submissao” duas vezes, uma após a ação “Coordenador.Abrir_Conferencia” e outra para uma etapa posterior ao *rebuttal*. No SIGMa o projetista define as consequências de uma ação, e onde quer que ela apareça, as suas consequências serão mostradas no cenário.

4.1.9 Caracterização dos Espaços de tempo

Após a definição de todas as instâncias possíveis de cada dimensão proposta, o projetista deve indicar quais delas são válidas em cada espaço de tempo. No sistema de gerência de conferências, por exemplo, o papel do revisor pode não fazer sentido ou ser válido durante a etapa de submissões de artigos. Assim, mesmo já definido, esse papel não faz sentido naquele espaço de tempo. Apesar de não ser uma dimensão ex-

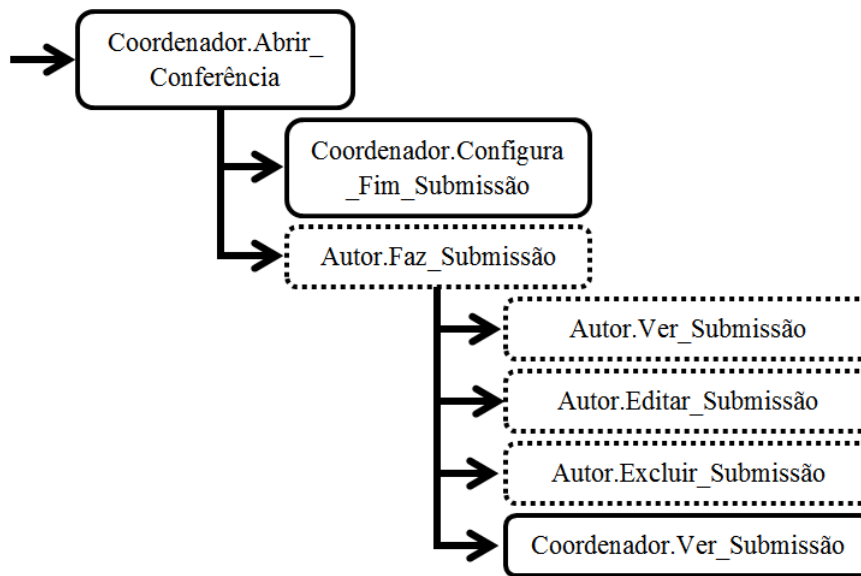


Figura 4.2. Cenários que seriam gerados pelo SIGN para as duas relações de exemplo mostradas na Figura 4.24.

plícita da linguagem de modelagem, a caracterização de cada espaço de tempo previsto é fundamental para a geração dos cenários pelo SIGN e para o melhor entendimento do modelo em si.

A sintaxe para a caracterização dos espaços de tempo é um par simples, que envolve a instância e a sua validade (*True* ou *False*) naquele espaço de tempo. A Tabela 4.25 mostra um trecho de uma caracterização de um espaço de tempo. Neste trecho de exemplo, o projetista está caracterizando o espaço de tempo “Submissão”, dizendo que os papéis “Autor” e “Coordenador” são válidos no espaço de tempo que está sendo modelado (“Submissão”) e que o papel “Revisor” não é válido. Diz ainda que os grupos “Autores_Aceitos” e “Autores_Rejeitados” não são válidos no espaço de tempo que está sendo modelado. Novamente ressaltamos que a Tabela 4.25 mostra um trecho da caracterização dos espaços de tempo, apenas para que o leitor entenda como é feita essa caracterização. O projetista deve caracterizar cada instância definida para cada espaço de tempo. A caracterização de cada espaço de tempo, ou seja, a definição da validade ou não de uma instância em um determinado espaço de tempo é considerada na geração dos cenários. No mesmo exemplo, ao gerar os cenários, se houver ações para os grupos “Autores_Aceitos” e “Autores_Rejeitados” envolvidas nas mudanças de alguma outra ação, serão ignoradas pelo SIGN, já que o projetista disse que esses papéis não são válidos para o espaço de tempo “Submissão”.

As dimensões propostas têm o intuito de permitir a modelagem de um sistema

Tabela 4.25. Exemplo de como caracterizar as instâncias válidas para cada dimensão em um espaço de tempo.

| Exemplo de caracterização dos Espaços de Tempo | |
|--|--|
| Espaço de Tempo | Submissão |
| Caracterização | <pre>//Papéis <Autor, True> <Coordenador, True> <Revisor, False> //Grupos <Autores_Aceitos, False> <Autores_Rejeitados, False> (...)</pre> |

colaborativo qualquer. Depois de modelar o sistema com base nestas dimensões, o projetista pode submetê-lo ao SIGN, descrito na subseção a seguir.

4.2 *SIGN - Scenario GeNerator*

Ao utilizar a linguagem de modelagem descrita acima, o projetista é capaz de criar um modelo da sua aplicação. Apesar de um modelo ter o potencial para auxiliá-lo a refletir sobre as decisões que toma durante o projeto, certamente antever os potenciais problemas que podem surgir do cruzamento de todos os eventos que podem acontecer durante a interação com o sistema pode ser um problema difícil de ser resolvido.

Por isso, nosso modelo é composto também pelo **SIGN**. Ele é responsável por gerar os cenários consistentes com o modelo feito pelo projetista. Assim, o projetista pode explorá-los livremente em busca de potenciais problemas, antes mesmo de qualquer fase de implementação. Assim, nosso objetivo é que o modelo proposto tenha o potencial de despertar reflexões na mente do projetista que poderiam não existir sem o seu uso. Vale ressaltar que, ao criar o modelo de sua aplicação utilizando a nossa linguagem, o projetista não descreve todas as possibilidades de interação possíveis, ou seja, toda a cadeia de consequências que cada ação causa no sistema. Ele apenas define, de forma isolada, as mudanças que cada ação traz ao sistema. O algoritmo do SIGN é responsável por estabelecer os caminhos possíveis e consistentes com o modelo gerado, a partir dessas definições isoladas. O algoritmo por trás do SIGN é detalhado a seguir.

O SIGN realiza uma análise sintática antes da geração dos cenários. Nesta fase são identificadas potenciais falhas como papéis ou grupos definidos mas sem ações relacionadas e espaços de tempo sem mudanças de tempo definidas. Essa verificação é

simplesmente complementar e serve para chamar atenção para pontos que o projetista possa ter esquecido de modelar.

Após a análise sintática, o algoritmo inicia a geração dos cenários. Para isso, ele utiliza basicamente um grafo direcionado para armazenar todos os caminhos possíveis entre as ações e suas consequências. Para criar este grafo, o algoritmo explora as relações definidas pelo projetista. Cada uma das mudanças geradas pela ação envolvida em uma relação são “ligadas” à ação que foi definida nesta relação. Estas ligações caracterizam as arestas do grafo. Para cada tipo de mudança é gerada uma aresta entre os nós correspondentes. Para ilustrar a construção do grafo, consideremos o exemplo de duas relações fictícias mostradas na Figura 4.3.

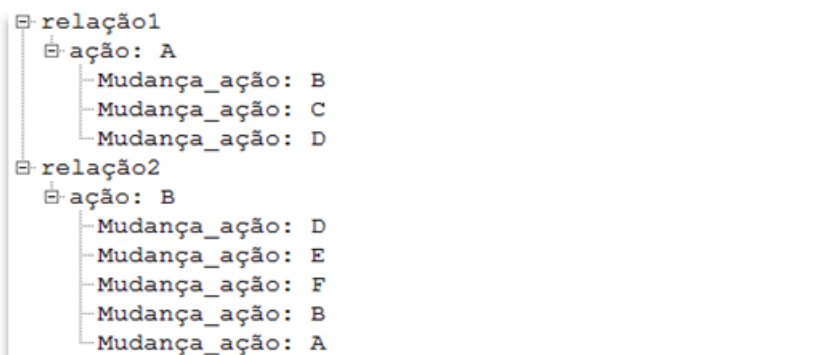


Figura 4.3. Exemplos de relações e suas consequências.

Esta figura ilustra duas relações (1 e 2) e cada uma delas tem a sua própria lista de mudanças. O algoritmo pega a ação definida na “relação1” (denominada “A”) e verifica se existe um nó no grafo relativo à essa ação. Caso não exista, é criado um nó no grafo com a ação “A”. Em seguida, o algoritmo passa a explorar as mudanças modeladas na relação. Para cada mudança encontrada, o algoritmo verifica se já existe um nó para a mudança. Caso não exista, este nó é criado. Em seguida, é criada uma aresta direcionada entre a ação definida na relação (“A”) e cada uma das mudanças. Quando há uma ação recursiva (como visto na relação 2, onde a ação “B” tem como consequência ela própria), o algoritmo cria uma aresta de *loop*. Este processo se repete até que todas as mudanças de todas as relações sejam exploradas. O grafo resultante para o exemplo da Figura 4.3 é mostrado na Figura 4.4. Vale ressaltar que o algoritmo explora os quatro tipos de mudanças possíveis (de espaços de tempo, no conjunto de ações, de papel/grupo e de posse) e gera uma aresta direcionada para cada nó “destino”.

Depois da criação do grafo, dá-se início ao processo de geração dos cenários. Neste passo, o algoritmo realiza uma busca em profundidade [Ziviani, 2004] no grafo, visitando todos os seus vértices e registrando a ordem em que os vértices são visita-

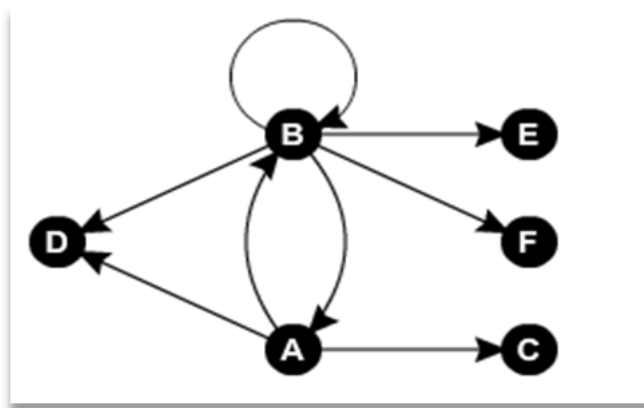


Figura 4.4. Grafo resultante das relações de exemplo definidas na Figura 4.3.

dos. Dessa forma, é possível determinar os caminhos possíveis de interação previstos. Vale ressaltar que esses caminhos podem ser infinitos (por exemplo, se observarmos novamente a Figura 4.4, as ações “A” e “B” geram um *loop* e este caminho pode ser executado indefinidamente). Assim, para que tenha uso prático, nosso algoritmo não gera todos os cenários possíveis de uma vez, mas o faz sob demanda, ou seja, à medida que o projetista sente a necessidade de aprofundar em um determinado cenário. Mais detalhes sobre a representação dos cenários serão mostrados logo em seguida, quando mostrarmos o protótipo que o implementa.

4.3 Protótipo implementado

Após definirmos as dimensões e o funcionamento do SIGN, foi necessário fazer a implementação de um protótipo que permitisse tanto a modelagem do sistema, quanto a exploração dos cenários². O protótipo foi implementado utilizando uma ferramenta de desenvolvimento rápido (Lazarus³). Vale ressaltar que não foi feito um projeto específico para a interface do protótipo. O único intuito da sua implementação foi permitir o uso real do modelo que estamos propondo. Assim, não avaliamos a interface do protótipo e tampouco ousamos dizer que ela é a mais adequada para representar o nosso modelo. Em passos futuros deste trabalho a interface do protótipo deve ser efetivamente projetada para que possa atender aos critérios de qualidade da área de Interação Humano-Computador.

Nesta seção mostramos o protótipo até então desenvolvido e descrevemos o seu funcionamento.

²O código-fonte do protótipo está disponível em <http://www.dcc.ufmg.br/~manuel/SIGN.rar>

³<http://www.lazarus-ide.org/>

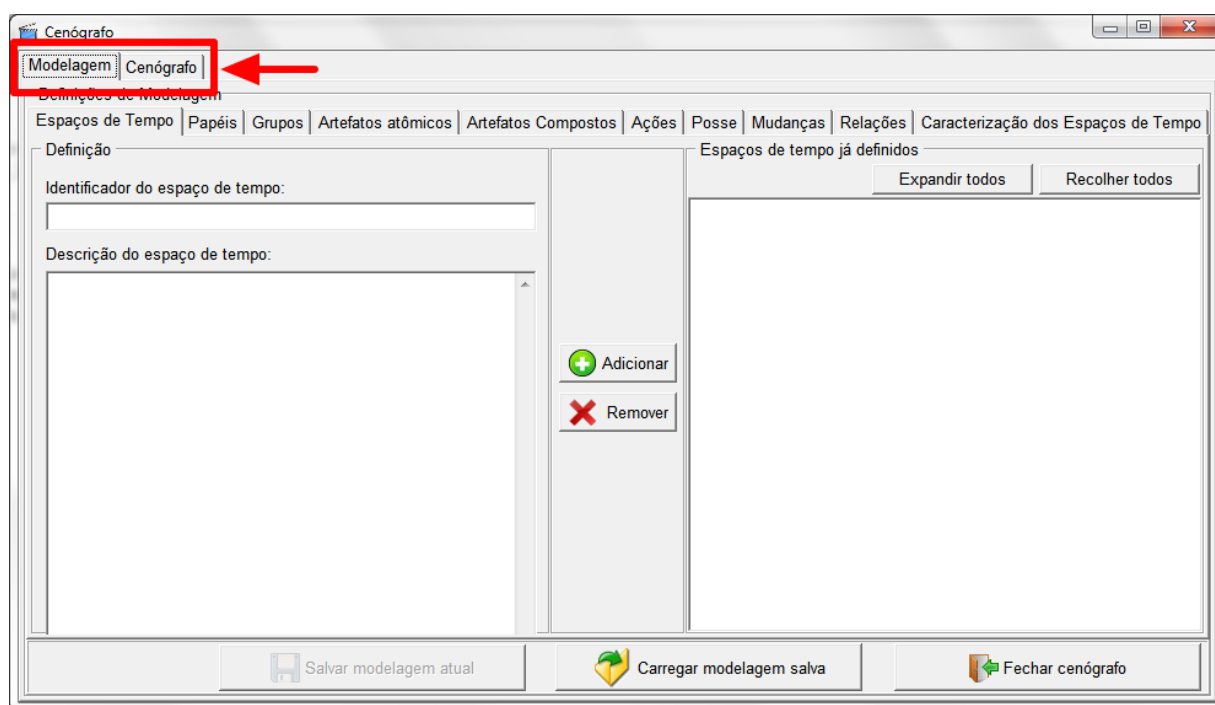


Figura 4.5. Imagem da tela do protótipo implementado, mostrando as duas principais áreas de interação.

O protótipo é composto por duas partes principais, destacadas na Figura 4.5. Na primeira parte, representada pela aba “Modelagem” o projetista pode editar visualmente seu modelo. Já na segunda parte (aba “SIGN”) o projetista pode interagir com o SIGN e explorar os cenários gerados.

Como visto na Figura 4.6, há abas específicas para cada dimensão de modelagem proposta e uma última, onde o projetista pode caracterizar cada espaço de tempo. Dentro da aba de cada dimensão, o projetista encontra os campos que são necessários para definir cada instância a ser modelada.

Após preencher cada um dos campos (área 1 da Figura 4.7), o projetista pode “Adicionar” esta nova instância ou pode “Remover” alguma instância existente (área 2 da Figura 4.7). Do lado direito da tela (área 3 da Figura 4.7), o projetista pode verificar as instâncias já definidas para cada dimensão. É possível ainda salvar o modelo atual ou carregar um modelo salvo anteriormente (área 4 da Figura 4.7).

Por exemplo, para modelar um espaço de tempo usando o protótipo, o projetista deve fornecer um identificador para o espaço de tempo que está modelando e uma descrição textual. Após inserir esses dados, o projetista pode adicionar o espaço de tempo que está criando ao modelo, como visto também na Figura 4.7.

As outras abas tem interface e funcionamento idênticos à essa aba de definição

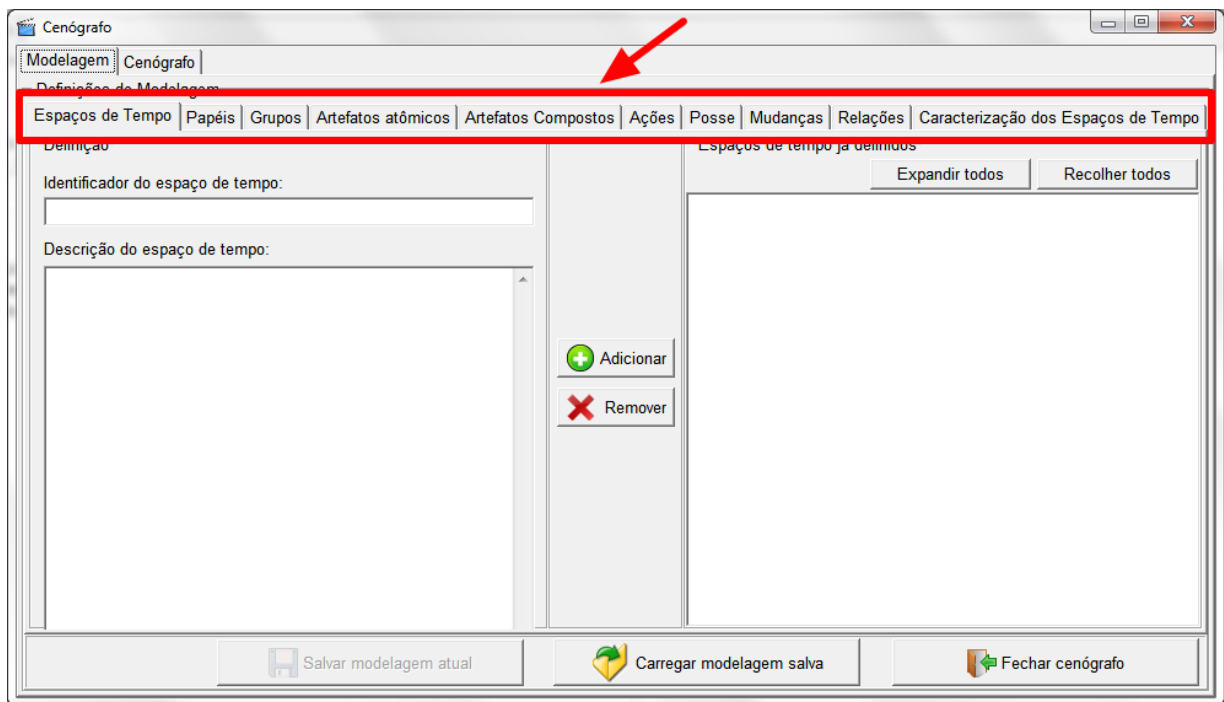


Figura 4.6. Abas que permitem a edição de cada dimensão prevista e a caracterização de cada espaço de tempo definido.

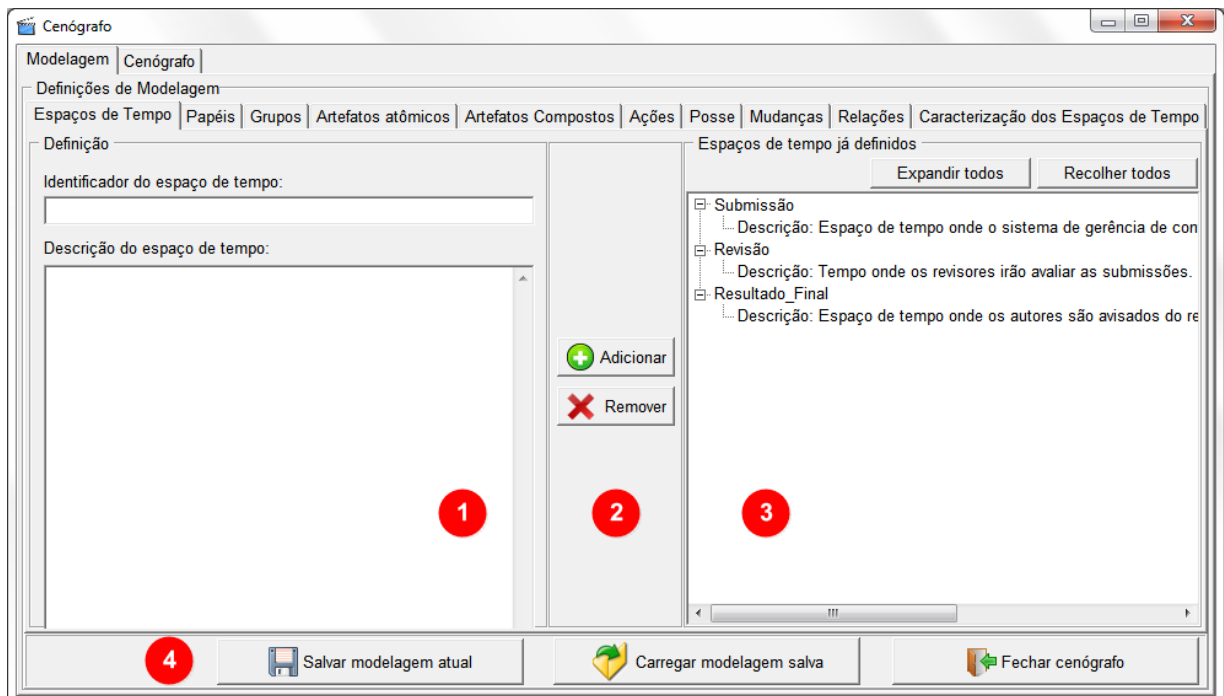


Figura 4.7. Estrutura de funcionamento do protótipo.

de espaços de tempo, por isso não são descritas neste texto. O projetista deve navegar

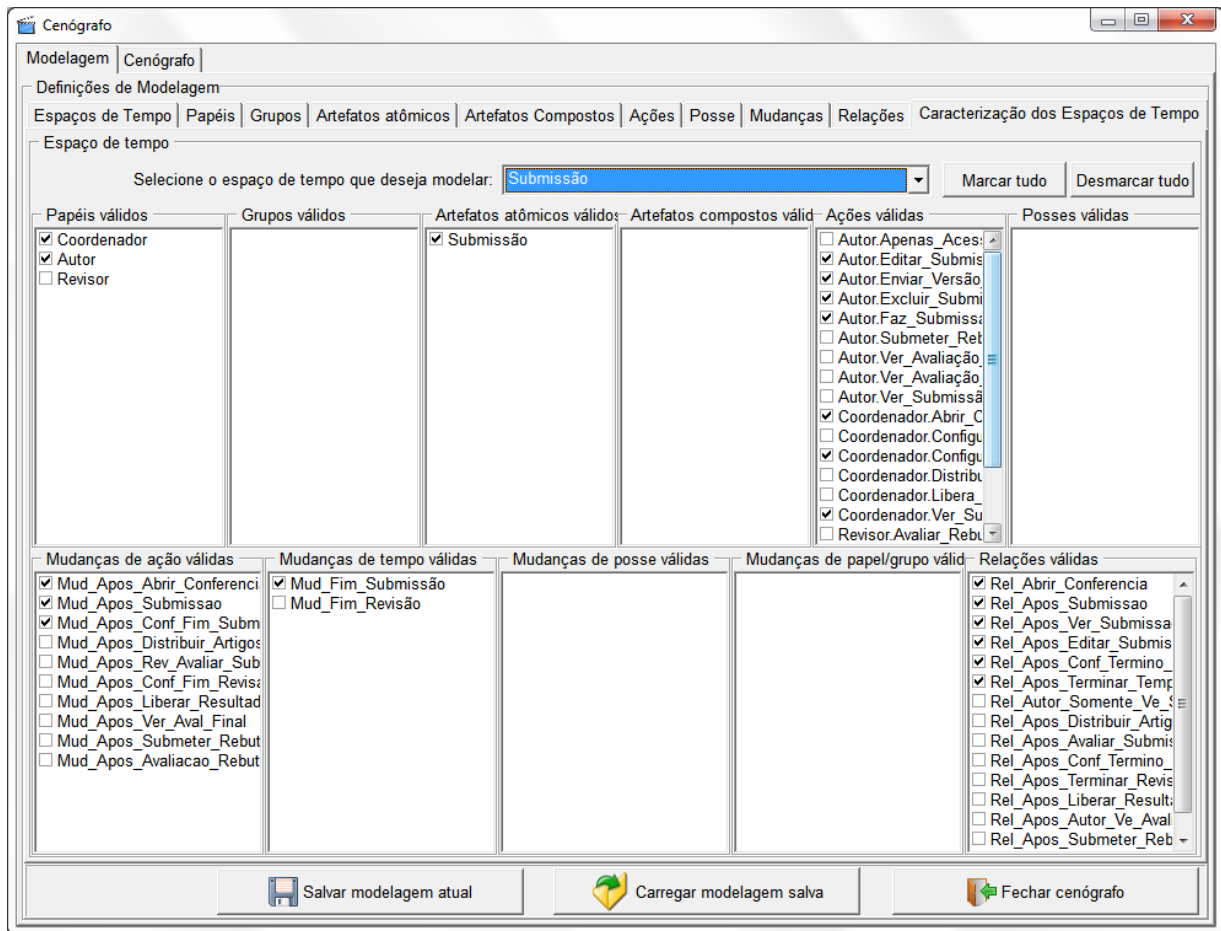


Figura 4.8. Aba de caracterização dos espaços de tempo.

pelas abas definindo os valores possíveis para cada uma das dimensões previstas que fazem sentido para o seu contexto. Vale ressaltar que, caso o projetista julgue que determinada dimensão não faz sentido para o seu contexto de projeto, basta ignorá-la, ou seja, não preencher nada na respectiva aba. Durante a geração dos cenários os potenciais impactos da dimensão que foi ignorada são desconsiderados.

Depois de inseridas todas as instâncias de cada dimensão, o projetista deve caracterizar cada espaço de tempo, ou seja, deve dizer quais instâncias definidas são válidas para cada espaço de tempo. Esta caracterização é possível através da última aba de modelagem. Para caracterizar um espaço de tempo, o projetista deve primeiro escolher o espaço de tempo que irá caracterizar e, sem seguida, selecionar as instâncias válidas para cada dimensão. Esta aba está ilustrada na Figura 4.8.

Depois de terminar seu modelo, usando a aba de “Modelagem”, o projetista pode então utilizar o SIGN. A Figura 4.9 mostra a interface do protótipo que permite ao projetista explorar livremente os cenários gerados.

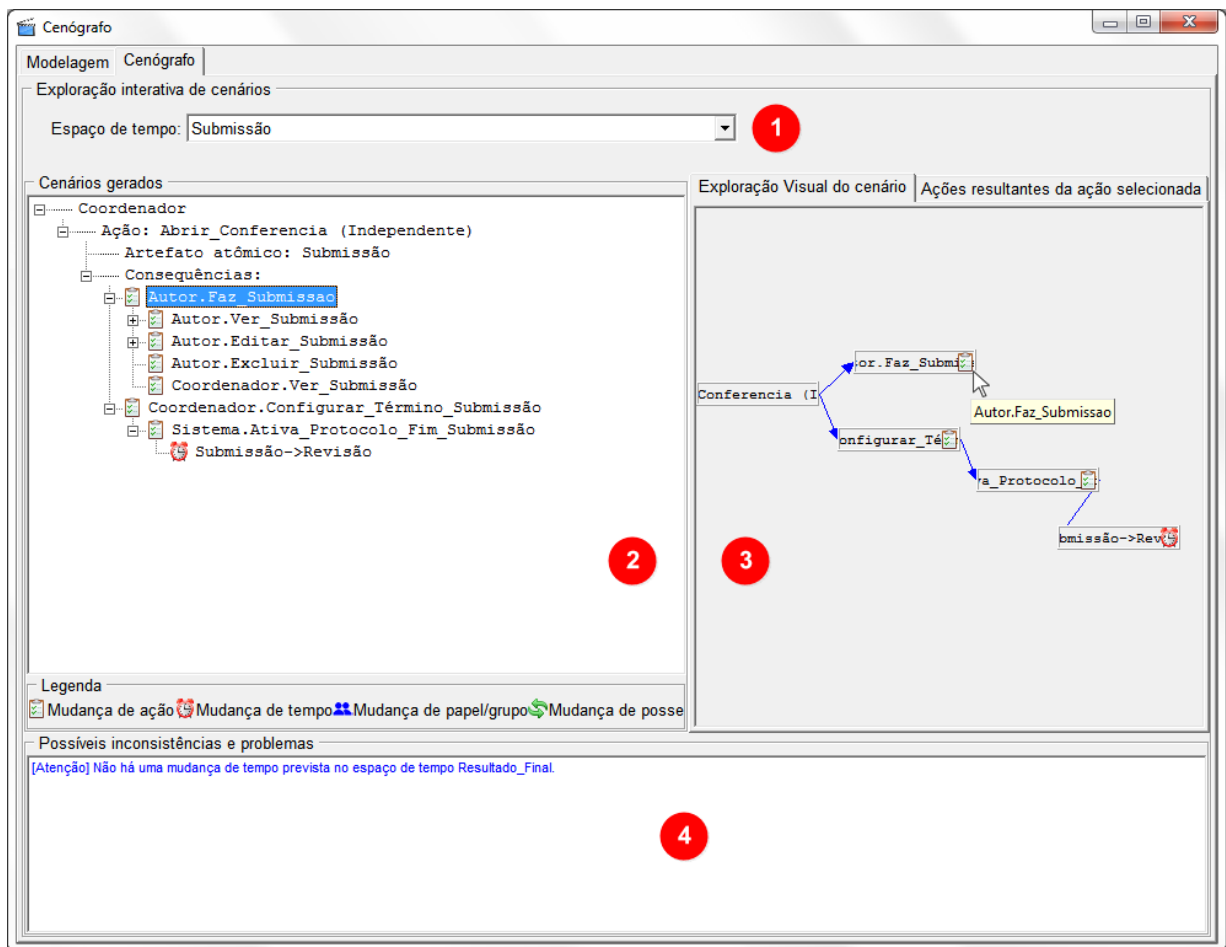


Figura 4.9. Imagem da tela do protótipo que permite a exploração livre dos cenários gerados destacando a visualização gráfica correspondente ao cenário explorado.

Na área 1 da Figura 4.9, o projetista pode selecionar o espaço de tempo que deseja explorar. Ao fazer isso, são gerados os cenários relativos às ações que foram modeladas (na respectiva relação) como “independentes” em forma de árvore, na área 2 da Figura 4.9. Como já dissemos, os cenários são gerados sob demanda. Assim, o SIGN mostra para o projetista os cenários relativos às ações independentes e os cenários que mostram as consequências dessas ações são gerados a medida que o projetista expande cada nó da árvore (que representa as ações independentes, no primeiro nível). Na área 2 da Figura 4.9 são mostrados os cenários relativos a cada espaço de tempo selecionado, de forma separada.

Como já discutido quando falamos do algoritmo, os cenários de um modelo podem ser infinitos. Assim, no protótipo, a geração dos cenários é gradual e interativa. Os cenários são gerados sob demanda, ou seja, são gerados conforme a necessidade e

interesse do projetista. O protótipo representa os cenários do espaço de tempo selecionado na área 1 da Figura 4.9 em forma de árvore, como visto na área 2 da mesma figura. As ações que possuem consequências são mostradas na árvore com o símbolo “+”. Ao clicar neste símbolo, o projetista pode explorar todas as consequências que são decorrentes daquela ação. A árvore mostra todos os quatro tipos de mudanças previstos e identifica cada uma delas através de um ícone diferente. Os ícones usados no SIGN podem ser vistos na Figura 4.10 e indicam (a) mudança no conjunto de ações, (b) mudança de tempo, (c) mudança nos papéis/grupos e (d) mudança de posse.

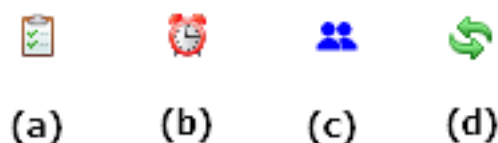


Figura 4.10. Ícones que caracterizam cada tipo de mudança.

Ao mesmo tempo que seleciona uma ação ou consequência na árvore de cenários (área 2 da Figura 4.9), o protótipo gera uma visualização gráfica (área 3 da Figura 4.9) do cenário que está sendo explorado, para permitir que o projetista veja o caminho que foi traçado até então ao longo dos vários espaços de tempo planejados para o sistema. Se observamos o exemplo da 4.9, a área 3 da mesma figura mostra o caminho que foi explorado pelo projetista ao interagir com a árvore de cenários (área 2 da Figura 4.9). Para cada nó da árvore explorado pelo projetista, é criada uma representação gráfica correspondente na área 3 da Figura 4.9, contendo o texto do nó selecionado. Como o texto pode ser grande para caber nesta representação, o projetista pode ver o texto correspondente ao passar o cursor do mouse sobre cada uma delas (como visto na Figura 4.9 para o caso da ação “Autor.Faz_Submissao”). Esta visualização pode ser interessante para o projetista pois através dela ele pode acompanhar o cenário entre os vários espaços de tempo possíveis, fato que não é possível na visualização em forma de árvore (área 2 da Figura 4.9). Assim, o projetista pode ter um entendimento completo da evolução dos cenários passando por todos os espaços de tempo possíveis.

A Figura 4.11 mostra um fragmento da interface do SIGN, mostrando a continuação da exploração do cenário pelo projetista, após ele ter selecionado o tempo “Revisão”. Nesta mesma figura podemos perceber a alteração de tempo “Submissão->Revisão” (simbolizada pelo ícone que indica uma mudança de tempo) e outras ações que foram inspecionadas pelo projetista (ao interagir com a árvore de cenários) após o início do novo espaço de tempo “Revisão”.

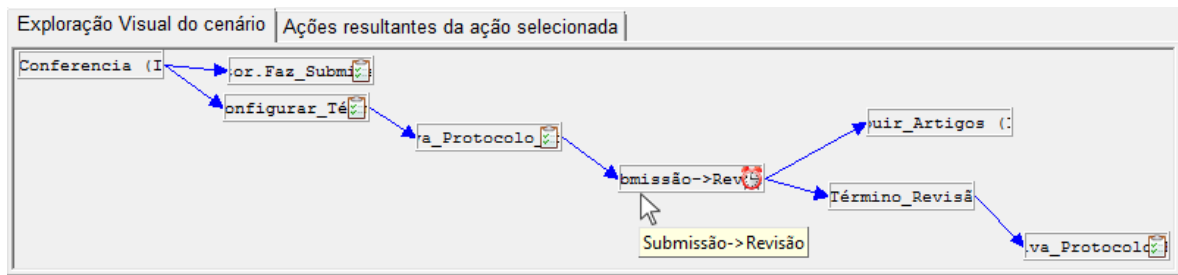


Figura 4.11. Fragmento da interface do SIGN mostrando a evolução de um cenário ao longo de dois tempos distintos.

Na área 4 da Figura 4.9 o protótipo mostra as potenciais inconsistências sintáticas do modelo identificadas pelo SIGN. As inconsistências sintáticas são listadas em azul e são apenas informativas.

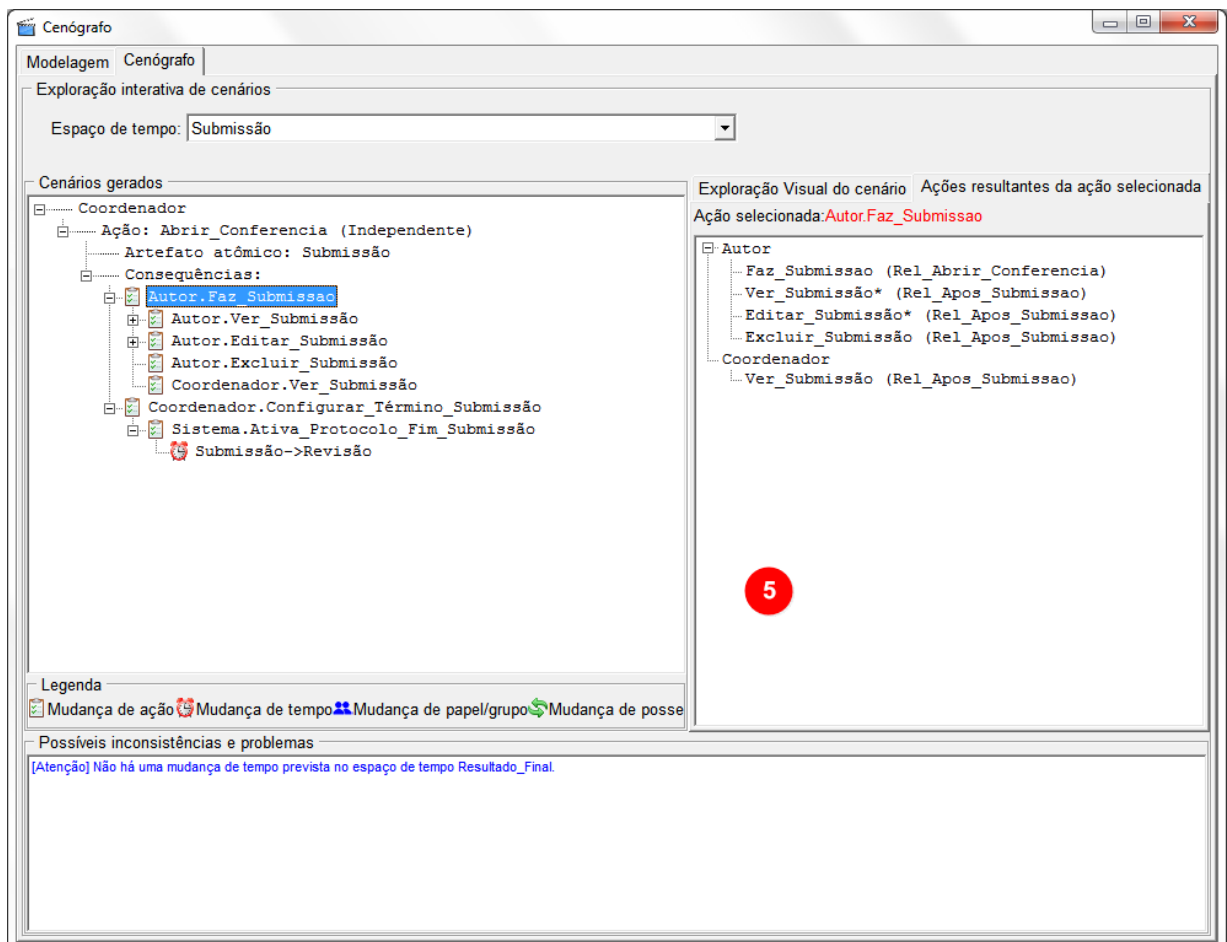


Figura 4.12. Imagem da tela do protótipo que permite a exploração livre dos cenários gerados destacando a listagem de ações possíveis subsequentes a partir de uma ação selecionada na árvore de cenários.

A natureza intrinsecamente infinita dos cenários gerados pode tornar a sua exploração na árvore da área 2 da Figura 4.9 uma tarefa árdua para o projetista. Pensando nisso, implementamos uma outra parte no SIGN que mostra ao projetista quais as consequências possíveis a partir de uma ação selecionada na árvore (Figura 4.12). Assim, ao selecionar uma ação na árvore de cenários, o SIGN informa todas as consequências possíveis a partir daquele ponto em diante do cenário. Se observarmos a Figura 4.12, ao clicar na ação “Autor.Faz_Submissao” na árvore de cenários, o SIGN mostra (área 5 da Figura 4.12) todas as ações que podem ser executadas a partir daquele ponto (em algum nível de profundidade) e ainda a relação que está permitindo que se alcance esta ação. Esta ferramenta permite ao projetista optar por fazer uma inspeção das consequências, em detrimento de ter que explorar cada passo do cenário. Novamente ao observar as consequências ele pode decidir se cada uma delas faz sentido para o seu contexto. O protótipo mostra ainda, entre parênteses, qual relação está permitindo aquela consequência. Caso o projetista opte por editar a relação, basta que ele clique sobre a consequência (na área 5 da Figura 4.12) que ele é levado automaticamente à relação correspondente, para que possa editá-la conforme sua necessidade. As ações que são recursivas, ou seja, que têm como consequência a si próprias, são listadas nesta área com um asterisco. O asterisco foi escolhido para indicar as ações recursivas pois é um termo comum usado na área de Computação quando se quer dizer que algo pode ser executado (ou representado) n vezes.

4.4 Considerações Finais

Neste capítulo apresentamos o modelo proposto neste trabalho, o SIGMa. O modelo prevê um ciclo iterativo, onde o projetista desenvolve seu modelo usando a SIGMa-dl e o submete ao SIGN para explorar os cenários consistentes com o seu modelo. Este ciclo se repete até que o projetista se dê por satisfeito com a modelagem feita. Ao interagir com o SIGN em cada iteração, o projetista pode experimentar e antecipar os diferentes cenários que pretende disponibilizar em sua aplicação colaborativa. Ao ter esta experiência, ele pode refletir sobre suas decisões e assim pode desenvolver um sistema que contenha menos cenários indesejáveis para os usuários finais. Assim, o modelo se torna uma nova ferramenta epistêmica que tem o papel de trazer novas reflexões ao projetista.

No capítulo seguinte, apresentamos as avaliações feitas para avaliar a expressividade da SIGMa-dl e a utilidade do SIGN.

Capítulo 5

Avaliação

O modelo apresentado no Capítulo 4 é composto pela SIGMa-dl, que permite ao projetista de uma aplicação colaborativa modelar um sistema de interesse de acordo com as dimensões previstas e pelo SIGN, que gera os cenários consistentes com a modelagem feita. Através de um protótipo, o projetista pode criar o seu modelo e explorar os cenários gerados pelo SIGN.

Uma vez proposto o modelo, realizamos uma avaliação qualitativa sobre o mesmo utilizando o *framework Cognitive Dimensions of Notations (CDN)* [Green, 1989]. O objetivo dessa avaliação foi identificar possíveis problemas e limitações do SIGMa e de levantar possíveis soluções para eles. Além disso, o CDN nos permitiu ter indicadores sobre o custo de aprendizado e uso da SIGMa-dl para o projetista. Avaliamos o SIGMa segundo cada dimensão prevista pelo CDN e também discutimos os impactos que os resultados de uma dimensão podem causar em outras, com o intuito de obter um entendimento mais completo dos pontos positivos e negativos que o SIGMa poderia conter.

Em seguida, com o objetivo de coletar indicadores qualitativos sobre a expressividade da SIGMa-dl e a utilidade e benefício do SIGN para o projetista, identificamos duas questões de pesquisa para investigação:

- **Questão de Avaliação 1 (QA1):** as dimensões previstas na SIGMa-dl são suficientes para modelar as mudanças que podem ocorrer no tempo ao longo do uso do sistema?
- **Questão de Avaliação 2 (QA2):** o SIGN gera cenários que levam o projetista a refletir sobre o modelo feito (assumindo seu papel de ferramenta epistêmica), possivelmente levando-o a alterar o modelo até que esteja satisfeito?

Para responder a estas perguntas, decidimos conduzir duas avaliações. Na primeira, executamos a reengenharia de sistemas colaborativos já existentes. O objetivo dessa avaliação foi verificar se a SIGMa-dl tinha expressividade suficiente para permitir a modelagem de sistemas existentes. Ao executar a reengenharia, pudemos analisar a capacidade da SIGMa-dl de representar decisões reais que foram tomadas pelo projetista e implementadas, gerando assim indicadores relativos à QA1. Assim, foram selecionados dois sistemas distintos para análise - Facebook (funções relativas à postagem de fotos) e Google Inactive Account Manager. Feita a modelagem, pudemos também analisar o comportamento do SIGN na geração dos cenários, endereçando assim a QA2. Na segunda avaliação, executamos testes com potenciais projetistas de sistemas colaborativos. Essa avaliação também tinha o objetivo de gerar indicadores sobre as duas questões de avaliação, porém levando em consideração a percepção de potenciais projetistas de sistemas colaborativos reais no uso da SIGMa-dl e SIGN. As tarefas solicitadas aos participantes envolvia tanto a análise de uma modelagem pronta, quanto a modelagem de um pequeno sistema colaborativo. Em ambos os casos foi solicitado aos participantes que explorassem os cenários gerados.

Para cada passo da metodologia prevista, descrevemos abaixo a sua aplicação e os resultados observados.

5.1 Avaliação usando o *framework* Cognitive Dimensions of Notations

Nesta avaliação usamos o *framework Cognitive Dimensions of Notations* (CDN)[Green, 1989]. Por isso, antes de descrever a avaliação em si e os resultados obtidos, na próxima subseção fazemos uma breve apresentação do CDN e suas principais características, para que se possa entender a avaliação do modelo proposto usando este *framework*.

5.1.1 O *framework Cognitive Dimensions of Notations*

Para Green [1989], as diferentes formas de interação com o computador podem ser descritas pela inter-relação entre notação e ambiente. Segundo o autor, não há como dissociar um do outro na interação humano-computador. Na falta de um ferramental que trate dessa relação e que permita avaliar as diferentes dimensões nela envolvidas, surge o *framework Cognitive Dimensions of Notations* (CDN).

O *framework* CDN apresenta um vocabulário comum que permite a discussão de vários aspectos cognitivos de uma notação, seja de artefatos interativos (processadores

de texto, ambientes de desenvolvimento de software, etc) ou de artefatos não interativos (como gráficos, tabelas, linguagens de programação, etc). Segundo Green [1989], os métodos de avaliação tradicionais de IHC focam principalmente nos artefatos interativos e não se aplicam bem a artefatos não interativos, como faz o CDN. O *framework* foi concebido para permitir o uso de projetistas não especialistas em IHC e, portanto, permite uma discussão mais geral em detrimento de uma análise detalhada da notação e pode ser utilizado tanto na avaliação formativa (durante o desenvolvimento do sistema) quanto na somativa (após o produto pronto, com o intuito de identificar melhorias).

Além de contemplar catorze dimensões, o CDN possibilita ainda uma reflexão sobre os problemas que se pode ter na combinação dessas dimensões, ou seja, ao se realizar um esforço para melhorar uma das dimensões, pode-se estar gerando um efeito contrário em outra. Abaixo apresentamos as catorze dimensões que atualmente compõem o *framework*, uma breve descrição e as principais relações entre elas [Green, 1989; Green & Blackwell, 1998; Blackwell & Green, 2003].

1. **Abstração:** indica o nível de abstração disponível para uso da notação. Indica se é possível ou obrigatório a criação de novas abstrações ou se deve-se usar apenas as abstrações já fornecidas pela notação. Exemplos de abstrações são a criação de macros e definição de estruturas de dados. Um alto nível de abstração pode tornar um sistema ou notação mais difíceis de aprender.
2. **Dependências ocultas:** indica a presença de ligações não explícitas entre componentes de uma notação. Por exemplo, uma planilha eletrônica pode conter várias células dependentes uma da outra e o usuário pode não perceber esta dependência.
3. **Comprometimento prematuro:** indica que existem restrições na ordem de execução de coisas. Em outras palavras, é a necessidade de se tomar uma decisão sem ter o conjunto completo de informações para tal. Por exemplo, definir todos os papéis de um sistema sem ter feito o levantamento de requisitos, ou ter que declarar todas as variáveis que irá precisar antes de pensar no código.
4. **Notação secundária:** indica a capacidade de registrar informações que não são antecipadas pela notação. Um exemplo claro de notação secundária é a possibilidade de inserir comentários em um código-fonte.
5. **Viscosidade:** indica a resistência a mudanças, ou seja, o esforço necessário para realizar uma mudança. A viscosidade pode ser diminuída se forem oferecidas abstrações adequadas (por exemplo a criação de uma macro). A viscosidade pode ser

subdividida em viscosidade de repetição: aquela em que uma mesma ação deve ser repetida várias vezes (por exemplo, ao mudar o nome de uma variável em um código fonte, deve-se atualizar todos os locais onde ela é usada) e viscosidade dominó: aquela em que uma mudança implica em outra para restaurar a consistência (por exemplo, para remover uma tupla de uma tabela de um banco de dados que serve como chave primária para outra tabela, primeiro deve-se excluir as tuplas que dependem dessa).

6. **Visibilidade:** indica a capacidade de ver componentes. Quando informações são encapsuladas, por exemplo, a visibilidade é diminuída.
7. **Proximidade de mapeamento:** relativo ao quão similar uma solução é do problema. Por exemplo, uma ferramenta que permite o desenho de circuitos eletrônicos tem uma alta proximidade de mapeamento, enquanto que um código-fonte tem uma baixa proximidade de mapeamento.
8. **Consistência:** relacionada a similaridade dos padrões em uma notação, facilitando a inferência sobre o seu uso. Em outras palavras, indica o quão bem uma parte da notação é “adivinhada” com sucesso após se aprender uma parte. Um exemplo disso é o comando de copiar (CTRL+C) que é padronizado em vários sistemas. Se não há consistência, a facilidade de uso da notação pode ser prejudicada.
9. **Difusão:** relacionada à verbosidade da linguagem, ou seja, o trabalho necessário para atingir um objetivo. Por exemplo, uma notação pode ser prolixa em excesso. Quando o nível de difusão é alto, o uso da notação pode ser mais difícil e cansativo.
10. **Operações mentais difíceis:** relativo à carga cognitiva necessária para o uso da notação. Por exemplo, uma linguagem pode usar elementos complexos para representar conceitos simples. Ou pode exigir raciocínios que demandam uma carga cognitiva excessiva.
11. **Propensão a erros:** indica se a notação facilita a ocorrência de erros no seu uso. Por exemplo, janelas de aviso (*warnings*) como as que excluem dados definitivamente, com o botão de exceção como padrão (o botão de confirmação da exclusão, por exemplo), facilitando assim que o usuário exclua seus dados definitivamente por engano.
12. **Provisoriedade:** indica a possibilidade de criar esboços ou rascunhos, ou seja, de usar a notação sem ter uma ideia completa do problema, de experimentar diferentes possibilidades. A provisoriedade diminui o comprometimento prematuro.

13. **Avaliação progressiva:** possibilidade de se avaliar o progresso do trabalho a qualquer momento. Por exemplo, é possível testar uma parte de um código-fonte antes de implementar todo o sistema.
14. **Expressão de papéis:** relacionado à facilidade de se inferir o propósito de um componente da notação e as suas relações com outros. Por exemplo, ao analisar um circuito eletrônico, pode-se identificar os pequenos componentes de forma separada e entender a sua relação com os outros.

As relações de dependência entre as dimensões da CDN são bastante complexas [Green & Blackwell, 1998]. Alguns desses “*trade-offs*” são mostrados em [Green & Blackwell, 1998] e [Blackwell & Green, 2003]. Por exemplo, a notação secundária pode aumentar a viscosidade, que por sua vez pode aumentar o comprometimento prematuro. Abstrações podem diminuir a viscosidade, mas em contrapartida podem aumentar a visibilidade, as dependências ocultas e o comprometimento prematuro. E uma alta visibilidade pode aumentar as dependências ocultas.

Ao avaliar um sistema ou notação usando o CDN, o projetista pode refletir sobre diferentes aspectos. Nesta reflexão, problemas podem ser identificados e melhorias podem ser propostas. Além disso, o projetista pode refletir sobre a combinação das dimensões e avaliar se as melhorias que poderiam ser propostas para uma determinada dimensão não trariam perdas em outra.

5.1.2 Resultados

Apresentamos abaixo os resultados que obtivemos durante a avaliação do SIGMa para cada uma das dimensões previstas pelo *framework* CDN.

- **Abstração:** a SIGMa-dl não permite que nenhuma abstração seja feita. Neste sentido, não podem ser criadas novas dimensões ou mesmo meta-definições abstratas sobre as quais seria possível criar definições mais especializadas. Assim, cada instância de cada dimensão, mesmo que parecida com outra, deve ser definida inteiramente, não sendo possível reaproveitar partes do modelo. A falta de abstrações pode aumentar a verbosidade da SIGMa-dl. Se observarmos o SIGN, o nível de abstração é também é baixo. O projetista não pode criar representações diferentes daquela implementada pela árvore de cenários e deve usá-la sempre na exploração dos cenários gerados.
- **Dependências Ocultas:** a linguagem de modelagem em si não possui dependências ocultas, pois apresenta claramente todas as ligações entre os elementos da

linguagem. No entanto, estas dependências aparecem durante a geração dos cenários. Como já dissemos, não é trivial enxergar todas as consequências de todas as ações em um sistema colaborativo ao longo do tempo. Ao usar o SIGMa, projetistas explicitam as relações e as respectivas mudanças. No entanto, pode ser difícil perceber ao longo do tempo os diferentes cenários que podem ser criados a partir da combinação de mudanças. Neste sentido, o cenário apresenta situações cujos processos de geração podem não estar representados explicitamente. Assim, o projetista pode não compreender o motivo de um determinado cenário ter sido gerado pelo SIGN. Este fato, no entanto, é minimizado no protótipo quando o projetista explora a funcionalidade que mostra a ele qual relação está permitindo a geração daquele cenário.

- **Comprometimento prematuro:** ao observar este quesito da CDN, a linguagem de modelagem proposta apresenta pouco comprometimento prematuro, visto que o projetista pode modelar apenas os detalhes que julga importantes para a situação sob análise.

O SIGN também não exige qualquer tipo de comprometimento prematuro, já que os cenários gerados são baseados no modelo que já foi feito, então o esforço de modelagem já foi feito quando o projetista interage com o SIGN.

- **Notação secundária:** todas as dimensões da linguagem de modelagem possuem o campo de descrição textual, que é obrigatório. Neste campo o projetista deve registrar suas decisões de projeto. Assim, o quesito notação secundária é contemplado pela SIGMa-dl.

Já o SIGN não tem esta capacidade, ou seja, o projetista não pode acrescentar nenhuma informação durante a exploração dos cenários. Neste ponto pudemos identificar um ponto que deve ser melhorado. À medida que explora os cenários, o SIGN deveria permitir que o projetista marcasse os cenários já inspecionados e que inclusive registrasse anotações e observações feitas sobre eles.

- **Viscosidade:** em relação à viscosidade, se analisarmos apenas a SIGMa-dl, percebemos que há um certo nível de viscosidade, visto que, quando há uma alteração em uma instância de uma dimensão (por exemplo no identificador de um papel), ela deve ser feita também em todos os lugares em que esta instância é usada. Porém, como a modelagem é apoiada pelo protótipo, esta viscosidade é dissolvida, pois o protótipo propaga as alterações em uma instância para todos os casos em que ela é usada.

- **Visibilidade:** ao refletir sobre este quesito da CDN, percebemos que nossa linguagem não apresenta problemas de visibilidade, ainda mais quando apoiada pela modelagem visual implementada no protótipo. Porém os cenários gerados pelo SIGN podem ser difíceis de acessar (por exemplo, pode haver um cenário problemático em um nível profundo da árvore de cenários). De toda forma, para tentar contornar o problema da visibilidade da árvore de cenários, o protótipo conta com a funcionalidade que mostra ao projetista quais as ações são possíveis a partir de uma ação selecionada na árvore de cenários. Assim, mesmo apresentando o problema de visibilidade, o SIGN possui essa ferramenta que tem o potencial de diminuí-lo.
- **Proximidade de mapeamento:** a maioria das dimensões previstas pela SIGMa-dl é natural em qualquer sistema colaborativo. Papéis, grupos, ações, artefatos e posses são conceitos muito próximos da realidade. Assim, para estas dimensões, o nível de proximidade é muito alto. No entanto, a dimensão *relações* é uma exceção a esta regra. Ao analisar essa dimensão, podemos notar que é difícil estabelecer uma ligação direta entre o seu significado e um sistema colaborativo, ou mesmo entre as consequências que ela exercem sobre a geração dos cenários. Também ao observar a dimensão *espaços de tempo*, apesar de ser comum um sistema colaborativo passar por diferentes contextos, o termo *espaço de tempo* pode não ser próximo da realidade dos projetistas. Assim, algumas dimensões da linguagem são muito próximas da realidade, enquanto outras não são.

A representação dos cenários pelo protótipo não se aproxima da visão no mundo real de um cenário. Porém, ao apresentar os cenários em forma de árvore, o SIGN se aproxima de representações que são bastante familiares aos projetistas de sistemas colaborativos, uma vez que são muito usadas em computação. Exemplos dessas representações são diagramas UML ou mesmo modelos CTT.

- **Consistência:** ao observarmos as dimensões da SIGMa-dl, podemos observar que elas têm consistência. Todas as dimensões iniciam com um campo de identificador e logo em seguida de descrição textual. A modelagem visual apoiada pelo protótipo também melhora a consistência, pois todas as telas de definição de cada dimensão funcionam da mesma forma. Assim, ao compreender como se define uma instância de uma dimensão, o projetista possivelmente terá um bom entendimento de como fazer o mesmo com outras dimensões. Ao observarmos os cenários gerados também observamos uma boa consistência, pois, ao compreen-

der o funcionamento de um nó da árvore, o projetista é capaz de inferir o seu funcionamento geral (que pode ter diversos níveis de profundidade, que se pode expandir ou retrair um determinado nó, etc.).

- **Difusão:** em relação à difusão, a SIGMa-dl, no seu estágio atual, apresenta alguns problemas. Apesar de ser apoiada pelo protótipo, a modelagem ainda requer um certo esforço para ser realizada. A SIGMa-dl poderia ser melhorada se fossem incorporados meios para o aproveitamento de partes da modelagem. Por exemplo, pode-se acrescentar à linguagem a capacidade de criação de meta-definições. E com base nestas, instâncias seriam criadas, dizendo apenas o que muda em relação à meta-definição. Se for necessário, por exemplo, modelar duas relações que possuem as mesmas características, com a exceção de uma, atualmente é necessário criar as duas relações por completo. Se houvesse uma meta-definição desta relação, seria mais fácil dizer apenas o que muda em uma e outra. Além disso, permitir a cópia de instâncias já definidas também poderia ajudar na difusão. Neste caso o projetista poderia copiar uma definição já pronta e fazer as modificações necessárias na cópia. Além disso, a definição das relações envolve muitos parâmetros. No entanto, as relações foram pensadas dessa forma para permitirem que uma única ação gere diferentes mudanças no sistema (o que de fato é possível).

Em relação ao SIGN, o trabalho de explorar todos os cenários possíveis pode ser grande. Porém ele é dotado de funcionalidades que têm o objetivo de diminuir este trabalho, como a identificação de problemas sintáticos e a listagem de ações possíveis a partir de um nó selecionado na árvore de cenários.

- **Operações mentais difíceis:** analisando as dimensões previstas pela linguagem de modelagem, podemos perceber que não há operações mentais difíceis, novamente com exceção das dimensões mudanças e relações. Essas dimensões exigem uma alta carga cognitiva para que tenham um uso eficaz. Como essas dimensões não têm uma boa proximidade de mapeamento, ou seja, é um conceito novo introduzido pela SIGMa-dl, aprender o seu conceito pode ser difícil. Nas relações especificamente, além de ter que definir diversos parâmetros, o projetista deve ainda informar quais mudanças são geradas por ela, podendo exigir uma alta carga cognitiva por parte do projetista.

Ao interagir com o SIGN, o projetista não precisa empregar uma alta carga cognitiva, pois os cenários são apresentados em forma de árvore e esta é uma representação comum no vocabulário de um projetista de sistemas computacionais.

Projetistas estão acostumados com representações de hierarquia e fluxo, como diagramas UML e diagramas entidade-relacionamento. Acreditamos que o projetista não terá problemas em explorar os cenários ao interagir com a estrutura de árvore.

- **Propensão a erros:** novamente ao analisar este quesito, a expectativa é que a maior propensão a erros esteja nas dimensões espaços de tempo, mudanças e relações, já que as outras tem uma alta proximidade de mapeamento. Em relação às outras dimensões, podemos notar que a propensão a erros é muito baixa, possivelmente devido à sua proximidade da realidade.

Já no SIGN a propensão a erros é menor, pois o projetista apenas interage com os cenários que já foram gerados. Assim, podem haver erros de interpretação dos cenários gerados, mas novamente estes podem ser minimizados se o projetista observar a funcionalidade que indica qual foi a relação que tornou aquele cenário possível.

- **Provisoriedade:** o nível de provisoriedade da SIGMa-dl é alto. Como já foi dito, ao modelar uma pequena parte da aplicação (um espaço de tempo, um papel ou grupo, uma ação para ele, um artefato, uma mudança no conjunto de ações onde a ação teria como consequência ela própria e uma relação considerando essa ação) o projetista já tem condições de interagir com o SIGN, podendo experimentar os cenários a partir dela. Assim, nosso modelo permite que o projetista faça diversos experimentos de modelagem e novas instâncias de cada dimensão podem ser criadas aos poucos. E esses experimentos auxiliam na criação de uma ideia mais geral e completa do sistema. Em outras palavras, como ferramenta epistêmica, nosso modelo permite que o projetista reflita sobre suas decisões de projeto e dessa forma o auxilia a buscar uma boa solução para um problema.
- **Avaliação progressiva:** se observarmos a capacidade de avaliação progressiva da linguagem, podemos perceber que ela é evidente e se torna ainda mais efetiva quando se leva em conta o SIGN. O projetista pode verificar os cenários resultantes de uma mudança no modelo assim que termina de fazê-la, bastando para isso recorrer ao SIGN. A iteração modelo-SIGN prevista pelo SIGMa permite que o projetista avalie constantemente as consequências das decisões de projeto que está tomando ao desenvolver o modelo de uma aplicação. É necessário, no entanto, definir um espaço de tempo, um papel ou grupo, uma ação para ele, um artefato, uma mudança no conjunto de ações (onde a ação teria como consequên-

cia ela própria) e uma relação considerando essa ação para que o projetista tenha condições de interagir com o SIGN.

- **Expressão de papéis:** como já foi observado em outras dimensões, como na proximidade de mapeamento, as dimensões previstas pela linguagem são bastante naturais para o projetista, uma vez que dizem respeito a características comuns do dia a dia do projetista, como papéis, grupos, ações, etc. No caso deste quesito da CDN, até mesmo a dimensão *mudanças* pode ter uma boa expressividade, pois chama a atenção do projetista para as características que mudam em um sistema. Porém, novamente a dimensão *relações* pode não ser tão intuitiva para o projetista. O rótulo dado a esta dimensão pode não informar imediatamente o seu propósito. As relações são o elo de ligação entre as ações e mudanças definidas. Ao observar o rótulo “relações” o projetista de sistemas colaborativos pode confundi-la com as relações possíveis entre os papéis e/ou grupos no sistema (por exemplo, pode haver uma relação de amizade entre dois usuários, ou uma relação de chefia entre um chefe e seus subordinados).

Em relação ao SIGN, a representação dos cenários apresenta uma boa expressividade. Ao visualizar a árvore de cenários o projetista pode perceber facilmente a hierarquia e interligação entre ações e pode identificar quem pode fazer a ação e qual artefato ela está manipulando.

Como visto, a análise do nosso modelo usando o CDN nos permitiu refletir sobre pontos positivos e negativos do modelo SIGMa. Basicamente os principais problemas identificados foram sobre as dimensões *mudanças* e *relações*. Estas duas dimensões realmente não são triviais para o projetista e exigem certa experiência com a linguagem para que possam ser usadas com eficácia. Apesar das limitações observadas para essas dimensões, elas permitem a modelagem de operações complexas com pouco esforço. Como já dissemos, ao usar o SIGMa o projetista não precisa especificar todos os caminhos possíveis de interação, ou seja, todos os cenários possíveis. Ele deve descrever apenas quais mudanças uma ação traz ao sistema (através de uma relação) e o SIGN se encarrega de gerar automaticamente todas as combinações possíveis em relação ao impacto de cada ação modelada. Como já dissemos no Capítulo 4, onde apresentamos em detalhes cada dimensão da SIGMa-dl, é importante desacoplar as ações das mudanças que ela causa. Isso pois uma mesma ação pode causar várias mudanças diferentes no sistema. E uma mesma mudança pode ser gerada por diversas ações diferentes. Esse é o papel da dimensão *relações*. Se não fosse assim, o projetista teria que definir

todos os caminhos de interação possíveis no sistema e, neste caso, o uso do gerador de cenários não se justificaria, pois ele já teria em mente todos os cenários possíveis.

Deve-se investigar sim se existem mudanças a serem feitas nas dimensões mudanças e relações a fim de diminuir seu custo, sem no entanto impactar no seu poder de representação e na geração dos cenários futuros. Por exemplo, deve-se investigar se é possível alterá-las para facilitar a visibilidade e proximidade de mapeamento e para melhorar a expressão de papéis sem no entanto aumentar a viscosidade da linguagem, o comprometimento prematuro e as dependências ocultas. Dar maior poder de abstração para a linguagem pode aumentar a carga cognitiva necessária para usá-la (operações mentais difíceis).

Como vimos, o SIGMa apresenta um baixo nível de abstração. Apesar de ser interessante a possibilidade da inserção de mecanismos de abstração, tais como a possibilidade de criar meta-definições de cada dimensão, e a partir dessa definir instâncias especializadas, a carga cognitiva para usá-las poderia também aumentar. Por outro lado, seria interessante investigar se a inserção de um mecanismo de abstração na representação dos cenários traria benefícios para o entendimento do projetista. Outros tipos de visualização dos cenários poderiam ser geradas, por exemplo uma descrição em linguagem natural ou algum outro tipo de representação gráfica que pudesse enriquecer o entendimento do projetista. Neste sentido é necessário investigar se é possível gerar novas representações, quais seriam essas novas representações e quais os ganhos que cada uma delas traria para o entendimento do projetista acerca dos cenários gerados.

5.2 Reengenharia de sistemas existentes

A primeira decisão tomada para a realização desse passo da avaliação foi a seleção dos sistemas colaborativos que passariam pelo processo de reengenharia. Foram selecionados dois sistemas reais de diferentes contextos para permitir a análise de diferentes aspectos a serem considerados na modelagem. Assim, os sistemas selecionados foram: (a) Facebook (rede social); e (b) Google Inactive Account Manager (configuração sobre futuro dos dados, uma vez que a conta Google do usuário se torne inativa).

A escolha do Facebook foi motivada pelos resultados de um trabalho anterior [Pereira Junior et al., 2014]. Neste trabalho identificamos que os usuários não compreendiam as consequências de suas ações ao lidar com a marcação de pessoas em fotos postadas no Facebook. O problema acontece quando um usuário “A” posta uma foto apenas para seus amigos, mas um desses amigos, o usuário “B”, marca um outro usuário “C”. O usuário “C”, mesmo não fazendo parte dos amigos do usuário “A”, passa a ter

acesso à foto. Isso pode trazer consequências indesejáveis e desagradáveis para o usuário “A” já que, ao postar a foto, imaginava que somente os usuários que pertencem ao seu grupo de amigos teriam acesso à foto. Assim, como é um problema real enfrentado pelos usuários do Facebook, seria relevante que o projetista pudesse identificá-lo em tempo de projeto. Assim, a reengenharia do Facebook buscou verificar se o SIGMA poderia modelar as ações disponíveis aos usuários, e se a partir desta modelagem o SIGN geraria (dentre os cenários possíveis) o cenário problema relativo a marcação de fotos.

O segundo sistema colaborativo escolhido foi Google Inactive Account Manager (Google IAM), que está disponível à qualquer usuário Google, dentro das suas configurações de sua conta. Neste sistema, o usuário pode configurar o que acontece com os seus dados relativos aos diversos produtos desta empresa (Gmail, Blogger, Drive, etc.) quando a conta se torna inativa (i.e. o usuário passa um determinado tempo sem usar nenhum produto Google). O usuário do sistema pode configurar o tempo que o caracteriza como inativo, ou seja, o tempo (em meses) que ele fica sem acessar sua conta. Após definir esse critério de inatividade, o usuário pode adicionar “contatos confiáveis”. Ao fazer isso, é possível escolher a quais dados cada contato terá acesso quando a inatividade for atingida (por exemplo, o usuário pode configurar que uma pessoa da sua família poderá fazer o download de seus dados do Google+ e do seu Gmail e que o seu chefe poderá fazer o download dos seus contatos do Google Contacts). Este sistema foi escolhido pois embora o usuário tome decisões e realize ações de configuração no presente, o impacto destas ações é sempre no futuro. Assim, o projetista deve antecipar os cenários que pretende disponibilizar ao usuário final para ter certeza de que cenários indesejáveis não possam resultar das decisões tomadas pelo usuário.

Uma vez selecionados os sistemas, conduzimos a reengenharia dos mesmos. No caso do Facebook, a reengenharia focou nas funcionalidades relativas à postagem e marcação de fotos, com inspiração nos resultados de um trabalho anterior [Pereira Junior et al., 2014]. Já no caso do Google IAM, a reengenharia contemplou todo o sistema. A reengenharia dos dois sistemas foi conduzida pelo autor dessa tese, com revisão por sua orientadora. Para que os modelos dos dois sistemas selecionados pudessem ser os mais fiéis possível em relação aos sistemas reais, fizemos a inspeção minuciosa desses sistemas e registramos todas as regras de funcionamento (e conseqüentemente todos os cenários possíveis) existentes em cada sistema. Para a parte relativa à fotos do Facebook, levantamos cerca de 30 cenários possíveis, enquanto que para o Google IAM, identificamos cerca de 8 cenários possíveis. Após efetuarmos a reengenharia, confrontamos os cenários gerados pelo SIGN com aqueles que levantamos durante a nossa

Tabela 5.1. Principais aspectos da modelagem do Facebook usando a SIGMa-dl.

| Dimensão | Instâncias definidas |
|------------------|---|
| Espaço de Tempo | Único |
| Papel | Postador |
| Grupos | Amigos_do_postador Amigos_dos_amigos_do_postador |
| Ações | Postador.Publicar_para_amigos Postador.Marcas_Foto Amigos_do_postador.Marcas_Foto Amigos_do_amigo_do_postador.Marcas_Foto Amigos_do_amigo_do_postador.Curtir_Foto |
| Mudanças de Ação | <ul style="list-style-type: none"> ▶ Chg_Act1: Postagem_Amigos <ul style="list-style-type: none"> > Amigos_do_Postador.Marcas_Foto ▶ Chg_Act2: Amigo_Marcou_Foto <ul style="list-style-type: none"> > Amigos_do_Amigo_do_Postador.Marcas_Foto > Amigos_do_Amigo_do_Postador.Curtir_Foto |
| Relações | <ul style="list-style-type: none"> ▶ Rel1: Postador.Publicar_Amigos <ul style="list-style-type: none"> > Chg_Act1: Postagem_Amigos ▶ Rel2: Amigos_do_Postador.Marcas_Foto <ul style="list-style-type: none"> > Chg_Act2: Amigo_Marcou_Foto |

inspeção dos sistemas.

5.2.1 Resultados

Para que o leitor possa compreender os resultados mostrados ao longo dessa subseção, mostramos um fragmento da modelagem que foi feita durante a reengenharia do Facebook e do Google IAM e em seguida apresentamos os resultados obtidos a partir dessa modelagem.

Como já dissemos anteriormente, a escolha do Facebook foi motivada pelas dificuldades constatadas que os usuários têm de entender completamente as possíveis consequências de suas decisões em relação à publicação de fotos no Facebook [Pereira Junior et al., 2014]. Assim, o escopo para a reengenharia do Facebook incluiu a parte de publicação e possíveis ações sobre fotos. A Tabela 5.1 mostra as principais dimensões usadas e as principais instâncias modeladas, com base nas funcionalidades relativas a fotos do Facebook.

Definimos apenas um espaço de tempo, pois não há diferentes contextos de uso que se alterem ao longo do tempo. Definimos também o papel “Postador”, que representa o usuário que irá postar uma foto no Facebook e os grupos “Amigos do Postador”

(que representa os amigos do postador da foto) e “Amigos dos amigos do Postador” (que representa o grupo de amigos dos amigos do postador da foto). Foi definido um artefato atômico denominado “Foto”, que será o artefato manipulado pelas ações modeladas. As ações de cada agente foram modeladas com base nas funcionalidades do Facebook disponíveis aos usuários em julho de 2015. O postador pode postar uma foto para amigos (Postador.Publicar_para_amigos) e pode também marcar seus amigos nas fotos (Postador.Marcar_Foto). Os amigos do postador podem marcar pessoas que o postador não conhece, mas que são amigos do amigo do postador. Finalmente, os amigos do amigo do postador podem marcar outros, mesmo que o postador não conheça. As mudanças modeladas foram do tipo mudanças de ação. A mudança denominada “Postagem_Amigos” permite que os amigos do postador executem a ação “Marcar Foto”. Já a mudança “Amigo_Marcou_Foto” permite que os amigos do amigo do postador também executem a ação “Marcar Foto”. Foram definidas ainda duas relações principais. Na relação “Rel1”, quando o Postador executar a ação Publicar_Amigos, a mudança “Chg_Act1” acontecerá no sistema, ou seja, os amigos do postador poderão marcar a foto. Já quando o grupo “Amigos_do_Postador” executar a ação “Marcar_Foto” acontecerá a mudança “Chg_Act2” no sistema, indicando que amigos de amigos poderão marcar ou curtir a foto.

A reengenharia do modelo do Facebook nos permitiu ter indicadores sobre QA1, que endereça a expressividade da SIGMa-dl. Foi possível representar as ações possíveis e as mudanças que resultariam destas ações. Uma questão que surgiu foi em como modelar a ação do usuário como postador e o grupo de amigos ou amigos de amigos que poderiam então agir sobre a foto. Afinal, o Facebook não tem diferentes papéis aos usuários. Porém a partir da perspectiva de um usuário, os demais usuários assumem diferentes papéis a partir de suas relações com este usuário. Assim, a decisão foi modelar como papel as diferentes relações que um usuário pode assumir com outro, e foi criado o papel de “Postador” e os grupos “Amigos do postador” e “Amigos dos amigos do postador”.

Ao confrontar os cenários gerados pelo SIGN com os que levantamos na inspeção do sistema real (cerca de 30 cenários possíveis), foi interessante notar que o SIGN foi capaz de gerar todos os cenários que levantamos e que representou o funcionamento geral das funcionalidades relativas à postagem de fotos no Facebook. Apesar de não estarmos realizando um reprojeto do Facebook, mas sim a sua reengenharia, a interação com o SIGN foi essencial para a geração do modelo do Facebook. Diversas vezes não encontramos cenários esperados dentre os gerados pelo SIGN e, ao voltar à modelagem, pudemos constatar os erros e então corrigi-los para que os cenários esperados pudessem ser gerados. Assim, mesmo executando a modelagem de um sistema onde os cenários já

eram esperados, pudemos observar indícios da utilidade do SIGN. No entanto, a ideia é que o projetista não tenha que conhecer os cenários, e sim que a partir da modelagem gere cenários que não tinha antevisto que poderiam ser possíveis.

O outro sistema selecionado para a execução da reengenharia foi o Google IAM. Como já dissemos, este sistema foi escolhido pois permite que o usuário faça configurações importantes sobre o seu legado digital. Após se tornar realmente inativo (por exemplo em caso de morte), não há como desfazer essas configurações. Assim, é necessário que o projetista tenha um conhecimento exato dos cenários que está disponibilizando ao usuário final para evitar problemas. A seguir apresentamos as principais decisões de projeto e os resultados obtidos.

No Google IAM, há dois espaços de tempo muito bem caracterizados: o momento em que o usuário está ativo, ou seja, está fazendo uso de sua conta sem que o tempo que configurou como de inatividade seja atingido e o tempo em que se torna inativo, ou seja, quando deixa de acessar sua conta pelo período que configurou como sendo característico de inatividade. Este sistema difere do anterior pois apresenta mais de um espaço de tempo. Assim, poderíamos avaliar como a SIGMa-dl lida com diferentes espaços de tempo.

Na Tabela 5.2 mostramos os principais pontos modelados usando as dimensões previstas na SIGMa-dl. Foram modelados dois espaços de tempo: “ativo” e “inativo”. No espaço de tempo “ativo”, o papel “usuário” pode executar as ações “define_criterio_inatividade”, “nomeia_contatos_confiveis”, “define_apenas_aviso” e “define_aviso_e_download”. O papel “usuário” representa o usuário que está configurando o que acontece com o seu legado digital enquanto que o papel “contato confiável”, representa a pessoa que receberá o legado digital do “usuário” quando ele se tornar inativo. O artefato tratado neste caso é o próprio legado digital do usuário, ou seja, seus dados gravados em diferentes sistemas da Google. Além das ações já mencionadas do papel “usuário”, modelamos ainda as ações realizadas automaticamente pelo sistema, como “Sistema.Verifica_Inatividade_Aviso”, que representa a situação em que o sistema identifica a inatividade do “usuário” e apenas avisa o “contato confiável” sobre essa situação e “Sistema.Verifica_Inatividade_AvisoDownload”, que representa a situação em que, além de avisar o “contato confiável”, o sistema permite ainda que ele faça o download dos dados do “usuário” que se tornou inativo.

Em seguida, modelamos as mudanças que podem acontecer neste sistema. A mudança de ação “Chg_act1”, quando acontece, permite que as seguintes ações sejam executadas: “Usuario.Define_Criterio_Inatividade” “Usuario.Nomeia_Contatos_Confiaveis”, “Sistema.Verifica_Inatividade_Aviso” e “Sistema.Verifica_Inatividade_AvisoDownload”. Já a mudança de ação

Tabela 5.2. Principais aspectos da modelagem do Google IAM usando a SIGMAdl.

| Dimensão | Instâncias definidas |
|------------------|---|
| Espaços de Tempo | Ativo Inativo |
| Papel | Usuario Contato_Confiavel |
| Artefato Atômico | Legado_Digital |
| Ações | Usuario.Define_Criterio_Inatividade Usuario.Nomeia_Contatos_Confiaveis Usuario.Define_Apenas_Aviso Usuario.Define_Aviso_e_Download Sistema.Verifica_Inatividade_Aviso Sistema.Verifica_Inatividade_AvisoDownload Sistema.Avisa_Contato_Confiavel Contato_Confiavel.Fazer_Download_Legado |
| Mudanças de Ação | <ul style="list-style-type: none"> ▶ Chg_Act1: Definir_Inatividade <ul style="list-style-type: none"> ➢ Usuario.Nomeia_Contatos_Confiaveis ➢ Usuario.Define_Criterio_Inatividade ➢ Sistema.Verifica_Inatividade_Aviso ➢ Sistema.Verifica_Inatividade_AvisoDownload ▶ Chg_Act2: Contatos_Nomeados <ul style="list-style-type: none"> ➢ Usuario.Define_Apenas_Aviso ➢ Usuario.Define_Aviso_e_Download ▶ Chg_Act3: Inatividade_Atingida_Aviso <ul style="list-style-type: none"> ➢ Sistema.Avisa_Contato_Confiavel ▶ Chg_Act4: Inatividade_Atingida_Download <ul style="list-style-type: none"> ➢ Sistema.Avisa_Contato_Confiavel ➢ Contato_Confiavel.Fazer_Download_Legado |
| Mudança de Tempo | <ul style="list-style-type: none"> ▶ Chg_Time1: Inatividade <ul style="list-style-type: none"> ➢ Ativo → Inativo |
| Relações | <ul style="list-style-type: none"> ▶ Rel1: Usuario.Define_Criterio_Inatividade <ul style="list-style-type: none"> ➢ Chg_Act1: Definir_Inatividade ▶ Rel2: Usuario.Nomeia_Contatos_Confiaveis <ul style="list-style-type: none"> ➢ Chg_Act2: Contatos_Nomeados ▶ Rel3: Sistema.Verifica_Inatividade_Aviso <ul style="list-style-type: none"> ➢ Chg_Time1: Inatividade ➢ Chg_Act3: Inatividade_Atingida_Aviso ▶ Rel4: Sistema.Verifica_Inatividade_AvisoDownload <ul style="list-style-type: none"> ➢ Chg_Time1: Inatividade ➢ Chg_Act4: Inatividade_Atingida_Download |

“Chg_act2”, quando acontece, permite que sejam executadas as ações: “Usuario.Define_Apenas_Aviso” (representando o caso em que o usuário quer definir que o contato confiável seja apenas avisado da sua inatividade) e “Usuario.Define_Aviso_e_Download” (representando o caso em que o usuário quer definir que o contato confiável, além de ser avisado da sua inatividade, possa fazer o download de seus dados). Quando a mudança “Chg_Act3” acontece, ela permite que a ação “Sistema.Avisa_Contato_Confiavel” (representando o caso em que o sistema apenas avisa que o usuário ficou inativo) seja executada. Finalmente, quando a mudança “Chg_Act4” acontece, ela permite que sejam executadas as ações: “Sistema.Avisa_Contato_Confiavel” e “Contato_Confiavel. Fazer_Download_Legado” (representando o caso em que o “contato confiável” pode fazer o download dos dados do “usuário” que se tornou inativo). Como já dissemos, este sistema apresenta ainda uma mudança de tempo, a qual denominamos “Chg_Time1”. Quando essa ação acontece, o espaço de tempo “ativo” é encerrado e inicia-se um novo espaço de tempo, o “inativo”.

Finalmente, modelamos as relações que definem o comportamento do sistema. Na primeira relação, denominada “Rel1”, quando o usuário executa a ação “Define_Criterio_Inatividade”, a mudança de ação “Chg_Act1” é disparada. Já na relação denominada “Rel2”, quando o usuário executa a ação “Nomeia_Contatos_Confiaveis”, é gerada a mudança “Chg_Act2”. Na relação “Rel3”, quando o sistema verifica que o usuário está inativo e que deve apenas informar o contato confiável (“Sistema.Verifica_Inatividade_Aviso”), são disparadas duas mudanças. Primeiro a mudança de tempo (Chg_Time1), que é caracterizada pelo término do espaço de tempo “ativo” e pelo início do espaço de tempo “inativo”. A segunda mudança causada é nas ações (Chg_Act3). Finalmente, na “Rel4”, quando o sistema identifica a inatividade do usuário tem-se a mudança de tempo (Chg_Time1) e uma mudança nas ações (Chg_Act4). Para as relações “Rel3” e “Rel4”, modelamos ainda uma pré-condição que é o critério de inatividade do “usuário” (em meses). Vale lembrar que essa condição não é avaliada pelo SIGN e tem caráter apenas informativo. De toda forma, ao ser registrada, a condição permite que o projetista entenda que a ação envolvida na relação só será executada quando ela for satisfeita.

Vale ressaltar que esta modelagem ilustrou duas características da linguagem. Primeiro mostrou como é possível duas ações (“Sistema.Verifica_Inatividade_Aviso” e “Verifica_Inatividade_AvisoDownload”) gerarem uma mesma mudança (Chg_Time1). Segundo, a capacidade de representar infinitos cenários a partir de uma única definição. Se observamos as consequências que a mudança “Chg_Act1”, prevista na relação “Rel1” traz, podemos observar que a ação “Usuario.Define_Criterio_Inatividade” faz parte

das consequências dessa mudança. Mas esta ação é a própria ação definida na relação “Rel1”. Assim, a ação “Usuário. Define_Criterio_Inatividade” é infinita, ou seja, dá origem à ela mesma quantas vezes for necessário. Essa decisão de modelagem foi tomada pois o “usuário”, enquanto ativo, pode alterar qualquer configuração a qualquer tempo.

Após realizarmos a modelagem desse sistema, pudemos analisar as duas questões de avaliação. Primeiro, em relação à expressividade da linguagem, pudemos notar que as dimensões previstas pela linguagem foram suficientes para modelar este sistema e o seu funcionamento geral pôde ser representado. A SIGMa-dl permitiu a modelagem de um cenário potencialmente infinito (aquele que envolve a ação de “Usuário.Define_Criterio_Inatividade”), uma vez que, na modelagem, foi definido que esta ação tem como consequência ela própria. Dessa forma, não foi necessário especificar todos os caminhos possíveis de interação, mas apenas uma relação (“Rel1”) que dá origem aos cenários potencialmente infinitos. Além disso, novamente fizemos uso de uma abstração para representar uma situação do mundo real (o caso do tempo “ativo”) e a SIGMa-dl foi capaz de contemplar essa abstração. Dessa forma, colhemos indícios para endereçar a QA1.

Novamente confrontamos os cenários gerados pelo SIGN com aqueles que levantamos durante a inspeção do Google IAM. Assim como no caso da reengenharia do Facebook, novamente o ciclo de uso previsto pelo modelo foi percorrido várias vezes, ou seja, todas as vezes em que o SIGN não apresentava os cenários esperados, voltamos à modelagem para verificar o que estava acontecendo. Este ciclo se repetiu até ficarmos satisfeitos com o modelo e cenários gerados. Por exemplo, ao verificarmos que a ação “Usuario.Define_Criterio_Inatividade” do usuário não gerava nenhuma consequência no sistema, pudemos refletir sobre a necessidade desta ação ser executada inúmeras vezes, e então voltamos ao modelo para fazer a correção necessária. Neste estudo pudemos ainda observar como o SIGN se comportou em relação à mudança de tempo. Ao selecionar a ação “Sistema.Verifica_Inatividade_AvisoDownload”, o SIGN foi capaz de indicar a mudança de tempo que essa ação gera. Ao selecionar o novo tempo (“inativo”) gerado no SIGN, pudemos observar as ações disponíveis (Fazer_Download_Legado) ao “Contato_Confiável”. E pudemos verificar também que o “Usuário” não podia mais executar nenhuma das ações que podia executar no tempo “ativo”. Novamente conhecíamos os cenários esperados, mas a ideia é que cenários imprevistos pelo projetista sejam gerados pelo SIGN.

5.3 Avaliação com Projetistas Reais

Após efetuarmos a reengenharia de sistemas colaborativos existentes, realizamos testes com projetistas reais de sistemas colaborativos, e potenciais usuários da SIGMa-dl para verificar suas impressões sobre a expressividade da linguagem e sobre a utilidade do SIGN, endereçando as duas questões de avaliação.

Conduzimos um teste piloto para avaliar o material gerado para o teste e as tarefas planejadas. Como o objetivo do teste piloto era avaliar o material em si, e não propriamente o SIGMa, selecionamos apenas um participante. Ele é analista de sistemas de uma autarquia do governo federal brasileiro. Ele participa ativamente do desenvolvimento de sistemas colaborativos reais e usa constantemente diversos sistemas colaborativos, como redes sociais, sistemas acadêmicos e de biblioteca, dentre outros. Os resultados do teste piloto levaram a pequenas correções de grafia, forma e concórdância no material produzido. Com o material corrigido, conduzimos o teste com seis projetistas de sistemas.

O primeiro passo do teste foi selecionar potenciais projetistas de sistemas colaborativos reais. Foram convidados seis participantes graduados em Ciência da Computação que atuam no desenvolvimento de sistemas colaborativos, seja desenvolvendo-os em aulas que ministram em cursos de Ciência da Computação ou atuando diretamente no mercado de trabalho. A escolha desses participantes foi proposital, pois era necessário que os participantes conhecessem aspectos de sistemas colaborativos para avaliar o SIGMa. Apesar de o SIGMa ser baseado na Engenharia Semiótica, não era necessário que os participantes tivessem qualquer conhecimento sobre esta teoria.

A avaliação com potenciais projetistas foi conduzida em uma única sessão no dia 28 de dezembro de 2015 e durou aproximadamente 5 horas. A avaliação consistiu das seguintes etapas: **(1)** identificação do perfil dos participantes e apresentação do objetivo da avaliação; **(2)** apresentação do SIGMa; **(3)** execução e **(4)** registro das percepções dos usuários através de questionário e grupo focal.

Na **Etapa 1**, apresentamos aos participantes o objetivo da avaliação e colhemos as assinaturas do termo de consentimento livre e esclarecido (Apêndice A). Em seguida todos os participantes responderam a um questionário pré-teste (Apêndice B). Esse questionário teve como objetivo levantar o perfil dos participantes e sua experiência no desenvolvimento e uso de sistemas colaborativos em geral. A etapa 1 levou cerca de 30 minutos para ser concluída.

Na **Etapa 2**, os participantes ouviram uma explicação sobre a SIGMa-dl e sobre o SIGN (Apêndice C). Primeiro foram apresentadas as dimensões previstas pela SIGMa-dl. A apresentação durou cerca de 30 minutos e os participantes puderam ti-

rar suas dúvidas em relação às dimensões apresentadas. Em seguida, os participantes receberam um treinamento sobre a operação do protótipo implementado, já que ele seria utilizado nos testes. Durante este treinamento, que durou cerca de 30 minutos, mostramos aos participantes um modelo pronto de um sistema de gerência da conferências, semelhante àquele utilizado como exemplo no início do Capítulo 4. Todos os participantes conheciam o funcionamento geral de sistemas de gerência de conferências existentes. Fizemos alterações no modelo de exemplo e mostramos as consequências dessas alterações no SIGN. Por exemplo, criamos uma nova relação no modelo e o SIGN gerou um novo cenário, que antes não existia. Em seguida, os participantes puderam explorar livremente a modelagem apresentada e puderam tirar todas as suas dúvidas em relação ao modelo e ao protótipo. Embora o protótipo tenha sido utilizado na avaliação, seu único propósito foi o de facilitar a etapa de modelagem, bem como implementar o algoritmo do SIGN. Assim, a interface do protótipo em si não era o foco da nossa avaliação. A etapa 2 levou cerca de 60 minutos para ser concluída, incluindo a explicação do SIGMa e a exploração da modelagem do sistema de gerência de conferências pelos participantes.

Demos início então à **Etapa 3** (execução do teste). O objetivo da etapa 3 foi gerar indicadores sobre as duas questões de avaliação, segundo a ótica dos projetistas reais e ainda contrastar as percepções de projetistas reais com aquelas obtidas por nós na reengenharia de sistemas colaborativos existentes. A execução da avaliação com os projetistas reais foi dividida em duas partes: (3.1) inspeção de uma modelagem pronta e (3.2) modelagem de um sistema novo (Apêndice D). O teste foi realizado em duplas com o intuito de permitir o debate entre os pares sobre a modelagem e os cenários gerados. Após a organização das duplas, entregamos os materiais necessários para a execução da avaliação. As duplas receberam um guia de consulta contendo as dimensões da SIGMa-dl e a sua definição. Além disso, foram disponibilizados computadores com o protótipo e o modelo que deveria ser inspecionado. O modelo representava uma modelagem parcial do Facebook, contemplando as funcionalidades relativas à postagem e marcação de fotos, e apresentava o potencial problema de um usuário marcado poder fazer marcações em uma foto sem que o postador tivesse ciência. No teste apresentamos o sistema como uma “rede social genérica”, com o objetivo de minimizar o efeito da experiência de uso real do Facebook pelos usuários. Isso foi importante para que os participantes pudessem explorar o modelo sem fazer nenhum pré-julgamento do seu funcionamento. Após a entrega do material, demos início ao teste. A etapa de inspeção (3.1) consistiu de duas tarefas:

- **Tarefa 3.1.1:** nesta tarefa cada dupla explorou apenas a modelagem dispo-

nibilizada (sem interagir com o SIGN) para verificar se eles compreendiam o funcionamento do sistema modelado e se conseguiam identificar algum potencial problema no modelo. Chamamos de potencial problema aquele que pode ter algum impacto negativo para o usuário final, como por exemplo o caso do impacto negativo que um usuário pode ter caso outro usuário não autorizado tenha acesso às suas fotos. O objetivo desta parte do teste foi verificar o entendimento dos projetistas sobre a modelagem feita usando as dimensões da SIGMa-dl, ou seja, verificar a adequação das dimensões para a modelagem do sistema em questão, endereçando a QA1. Após a inspeção, a dupla deveria descrever detalhadamente e em texto livre o funcionamento esperado do sistema, de acordo com o modelo apresentado. Os participantes deveriam registrar ainda os pontos em que tiveram opiniões divergentes entre si e se chegaram a um consenso ou os casos em que não concordavam com o modelo apresentado, devendo neste caso apontar as mudanças que fariam. Os participantes foram instruídos a não fazerem perguntas sobre o modelo que estavam inspecionando, justamente para que a análise feita por eles não fosse enviesada por nós. No entanto, perguntas sobre o significado das dimensões (sem nenhuma ligação com o sistema a ser modelado) e sobre a operação do protótipo poderiam ser respondidas durante o teste.

- **Tarefa 3.1.2:** nesta tarefa solicitamos aos participantes que continuassem analisando o mesmo modelo da tarefa 3.1.1, mas agora usando o SIGN. Nesta tarefa as duplas interagiram livremente com o SIGN e com a modelagem. Durante a tarefa, as duplas deveriam fazer o mesmo registro que fizeram na tarefa 3.1.1, ou seja, deveriam descrever o funcionamento esperado do sistema observando os cenários gerados pelo SIGN e deveriam registrar as discordâncias que tiveram entre si e/ou sobre o modelo apresentado. As duplas foram orientadas a fazer alterações no modelo caso achassem necessário. Além disso, ao terminar as tarefas, deveriam responder algumas perguntas abertas. Essas perguntas tinham o objetivo de verificar se os participantes foram capazes de identificar cenários problemáticos no modelo feito, se o SIGN causou novas reflexões sobre o modelo, se essas reflexões levaram à mudanças no modelo e por fim, se o SIGN gerou algum cenário incorreto, ou seja, se o algoritmo implementado estava gerando cenários inconsistentes com o modelo feito. Novamente foram respondidas apenas dúvidas em relação à interação com o protótipo e às dimensões da SIGMa-dl. O objetivo desta tarefa foi observar o contraste entre o entendimento dos participantes sobre o funcionamento do sistema, com base no modelo apresentado, antes e depois do uso do SIGN, com o objetivo de colher indícios sobre a QA2.

A primeira parte da avaliação (3.1), onde os participantes exploraram um modelo pronto, durou cerca de 30 minutos.

Após inspecionarem um modelo pronto, solicitamos aos participantes que executassem a modelagem de um sistema a partir do “zero”, ou seja, eles deveriam tomar todas as decisões de como modelar um sistema usando as dimensões previstas pela linguagem que estamos propondo (parte 3.2). As duplas foram mantidas para a realização dessa parte da avaliação. O objetivo desta etapa foi observar a viabilidade de uso do modelo proposto para a modelagem de um sistema. O objetivo desta etapa era coletar indicadores sobre a expressividade da linguagem, ou seja, se os participantes conseguiam representar as decisões desejadas sobre o sistema usando as dimensões previstas pela SIGMa-dl, e coletar indicadores sobre a utilidade do SIGN como ferramenta epistêmica, ou seja, se a inspeção dos cenários gerados apontava aspectos que não tinham sido considerados ou percebidos e que causaram novas reflexões sobre as decisões e sobre a solução proposta pelos participantes. Uma especificação do sistema foi entregue aos participantes. Essa especificação continha o funcionamento de um sistema de edição de documentos com configurações de um sistema de legado digital similar ao Google Inactive Account Manager (Apêndice E), conhecido pelos participantes. Este sistema foi escolhido por ser menor, já que um sistema grande poderia demorar muito para ser modelado. Pela especificação, um usuário deveria poder criar, ver, editar e excluir documentos. Além disso, pode configurar o que acontece com seus documentos quando ele se tornar inativo. Ao configurar o seu critério de inatividade (quantos meses sem usar o sistema por exemplo), o usuário deve determinar quem receberá acesso aos seus arquivos e o que a pessoa que recebeu o acesso poderá fazer sobre ele (apenas ler, ler e editar ou ler, editar e excluir). Com base nesta especificação, solicitamos que as duplas executassem duas tarefas:

- **Tarefa 3.2.1:** os participantes deveriam fazer a modelagem do sistema proposto.
- **Tarefa 3.2.2:** os participantes deveriam inspecionar os cenários relativos ao seu modelo gerados pelo SIGN.

Diferente do teste feito na primeira parte (3.1), neste os participantes poderiam navegar livremente entre as tarefas 3.2.1 e 3.2.2, justamente para reproduzirmos uma situação real de modelagem, onde o projetista pode verificar os efeitos de sua modelagem à medida que progride na sua especificação. Os participantes foram instruídos a registrar todas as decisões de projeto usando os campos de descrição textual da própria SIGMa-dl e também os formulários fornecidos. Eles deveriam registrar principalmente as mudanças que fizessem em seus modelos quando perceberam algo inesperado durante

a interação com o SIGN. Os modelos criados por cada dupla foram copiados ao final do teste e os formulários escritos foram recolhidos para análise posterior. A segunda parte da avaliação (3.2) durou cerca de 90 minutos para ser executada.

Finalmente, na **Etapa 4**, registramos as impressões dos participantes através de um questionário pós-teste (Apêndice F) e da realização de um grupo focal (Apêndice G). O questionário era composto por questões abertas que buscavam levantar a percepção dos participantes sobre a expressividade da SIGMa-dl e sobre a utilidade do SIGN, endereçando novamente as questões de avaliação. Os questionários foram respondidos por cada participante individualmente. Optamos por realizar adicionalmente o grupo focal para possibilitar a discussão e confronto de ideias entre os participantes. Toda a discussão gerada pelo grupo focal foi registrada em áudio e posteriormente transcrita para análise. A aplicação do questionário durou cerca de 30 minutos e a realização do grupo focal cerca de 60 minutos.

5.3.1 Resultados

Essa parte da avaliação do SIGMa contou com a participação de seis participantes que desenvolvem sistemas colaborativos em ambiente acadêmico ou profissional. Todos os seis participantes possuem graduação em Ciência da Computação. Quatro participantes são professores de uma instituição de ensino pública brasileira, denominados P1 a P4. Estes compuseram o grupo de profissionais do meio acadêmico, sendo que dois deles já participaram do desenvolvimento de sistemas colaborativos reais no mercado de trabalho, antes de se tornarem professores. Todos os quatro professores participantes são mestres, sendo que um deles está terminando seu doutorado em Ciência da Computação. Todos já lecionaram disciplinas específicas de modelagem e desenvolvimento de software em geral, tais como Engenharia de Software, Programação Orientada a Objetos, Banco de Dados, dentre outras, em cursos técnicos e superiores de Computação. Todos fazem uso constante de sistemas colaborativos no seu cotidiano, tais como sistemas acadêmicos e de biblioteca, Moodle, redes sociais online, dentre outros. Nenhum dos participantes teve contato anterior com o modelo proposto neste trabalho de doutorado e também não possuem conhecimento sobre Engenharia Semiótica. Os outros dois participantes do teste atuam no mercado de trabalho. Um deles, denominado P5, é Gerente de Projetos em uma empresa de desenvolvimento de software. Gerencia uma equipe de 14 pessoas que desenvolvem principalmente sistemas web colaborativos. Além de gerenciar uma equipe de desenvolvimento de sistemas colaborativos no seu dia a dia, faz uso constante de sistemas colaborativos em geral, incluindo redes sociais online, Moodle e sistemas acadêmicos. O sexto participante, denominado P6,

é Técnico em Informática de uma instituição do governo federal brasileiro e participa ativamente no desenvolvimento de sistemas colaborativos. Um dos sistemas onde atuou diretamente é inclusive utilizado por várias autarquias federais do Brasil.

Conforme explicado, as tarefas foram feitas em duplas, definidas pelos próprios participantes. A primeira dupla foi formada por P1 e P4, a segunda dupla foi formada por P2 e P5 e a terceira dupla foi formada por P3 e P6. As duplas P2-P5 e P3-P6 foram compostas por um participante do meio acadêmico e um do meio profissional, enquanto que a dupla P1-P4 foi composta por dois participantes do meio acadêmico. As duplas foram formadas pelos próprios participantes de forma independente, sem a nossa intervenção.

Nesta seção apresentamos os resultados observados na condução da avaliação, buscando responder QA1 e QA2. Os dados empíricos coletados durante a realização dessas tarefas foram obtidos da seguinte maneira: primeiramente, analisamos as descrições de funcionamento e todas as anotações feitas (por exemplo quando discordaram e quais foram as discordâncias) pelos participantes durante a realização das tarefas 3.1.1 e 3.1.2; em seguida, analisamos os modelos gerados por cada dupla e as anotações feitas nos formulários fornecidos (discordâncias e consensos), obtidos na realização das tarefas 3.2.1 e 3.2.2 e, por fim, analisamos as respostas de questionários com perguntas abertas e discursos de todos os participantes durante a realização do grupo focal. Para enriquecer o texto, inserimos os comentários de cada dupla/participante quando oportuno. Todas as respostas estão entre aspas e em itálico e foram fielmente transcritas.

5.3.1.1 Inspeção de uma modelagem pré-definida

Para colher os indicadores referentes aos resultados mostrados nesta subseção, analisamos as descrições do funcionamento do sistema geradas pelos participantes a partir da sua análise da modelagem parcial do Facebook (tarefas 3.1.1 e 3.1.2) e também as respostas dos questionários após a realização das tarefas.

Ao analisar as descrições feitas por cada dupla, relativas à tarefa 3.1.1, identificamos as funcionalidades descritas e verificamos a sua adequação quanto ao resultado esperado. Nessa análise pudemos observar que as descrições das três duplas foram consistentes entre si. Todas as descrições apresentaram os mesmos papéis e as mesmas ações para cada um deles. Foi interessante notar que todas as duplas descreveram apenas os papéis previstos e as ações de cada papel. Eles não registraram as consequências (o fluxo de ações) que cada ação de cada usuário traz ao sistema. Por exemplo, todas as duplas listaram que os “Amigos_dos_Amigos_do_postador” podem realizar a ação “Marcar_Foto”, mas não registraram que essa ação só pode acontecer após

um “Amigo_do_Postador” ter executado a ação “Marcar_Foto”. No entanto, pudemos perceber que os participantes tiveram um entendimento parcial do funcionamento geral do sistema, já que nenhuma das duplas conseguiu visualizar o cenário problemático da marcação de fotos. Além disso, os participantes registraram discordâncias em relação ao seu entendimento e ao modelo. Por exemplo, a dupla P3-P6 registrou que eles não concordaram sobre as consequências da ação “Publicar_para_Amigos” do “Postador”, mas que depois chegaram a um consenso de que essa ação permitiria que os amigos do postador executassem ações sobre a foto. Durante a análise, pudemos perceber também indícios sobre a dificuldade que as duplas tiveram ao analisar um modelo pronto sem o uso do SIGN. Em suas próprias palavras, a dupla formada por P1 e P4 registrou que *“o trabalho de compreender um modelo pronto e procurar por potenciais problemas no modelo é mecânico e tedioso”*. Esta percepção já era esperada pois os participantes nunca tinham tido contato com o SIGMa antes. Em seguida analisamos as descrições feitas pelas duplas na tarefa 3.1.2.

Na tarefa 3.1.2 solicitamos aos participantes que continuassem analisando o mesmo modelo, mas agora usando o SIGN. Os participantes deveriam explorar todos os cenários gerados. Ao analisar as descrições dos cenários feitas pelas duplas, pudemos constatar que após interagir com o SIGN a descrição dos cenários resultantes ficou mais completa, se comparada às descrições feitas na tarefa 3.1.1. As descrições feitas ao interagirem com os cenários ficaram muito mais detalhadas. Além disso, as descrições dessa vez foram feitas em forma de fluxo de ações, ou seja, as descrições contemplaram a sequência de ações modeladas (ou seja, as consequências de cada ação). Por exemplo, todas as duplas incluíram em sua descrição o caso em que a um “amigo do amigo do postador” poderia executar ações sobre ela (curtir, comentar, etc.) somente após um “Amigo_do_Postador” ter executado a ação “Marcar_Foto”. Incluíram ainda uma observação sobre a ação “Marcar Fotos” do Postador da foto ser potencialmente infinita (o modelo apresentado aos participantes possuía essa característica). Como dissemos, o funcionamento do sistema descrito durante a realização da tarefa 3.1.1 foi estático, ou seja, se limitou a apresentar apenas os papéis e as respectivas ações. Já a descrição do funcionamento do sistema durante a realização da tarefa 3.1.2 ficou muito mais detalhada, incluindo não só os papéis e ações previstos, mas também o fluxo entre essas ações. Durante a realização da tarefa 3.1.2 não houve discordância entre os participantes. Ao confrontar os resultados das descrições feitas nas tarefas 3.1.1 e 3.1.2, pudemos observar que os participantes compreenderam parcialmente o modelo feito usando a SIGMa-dl durante a tarefa 3.1.1, e durante a tarefa 3.1.2 esse entendimento foi melhorado, com o apoio do SIGN. Como os participantes da avaliação conseguiram compreender o modelo feito com as dimensões previstas na SIGMa-dl, na tarefa 3.1.1,

observamos indicadores positivos sobre a sua expressividade. Além disso não houve registro de discordâncias em nenhuma dupla. Esses indícios da expressividade nos permitiram endereçar a QA1. Como as descrições ficaram mais detalhadas e mais próximas do esperado com o uso do SIGN, na tarefa 3.1.2, pudemos observar que o SIGN traz ganhos para o entendimento do projetista, gerando indícios para endereçarmos a QA2.

Em seguida, avaliamos as respostas que as duplas forneceram a um questionário que continha perguntas abertas em relação ao entendimento que tiveram do modelo antes e depois da interação com o SIGN. Nesse questionário, todas as duplas responderam que acreditaram que seu entendimento do sistema melhorou após a interação com o SIGN e que, ao explorar os cenários, conseguiram identificar o cenário potencialmente problemático da marcação de fotos. A dupla formada por P2-P5 registrou que, ao perceber o problema da marcação de fotos, voltaram ao modelo e removeram a relação que dava origem ao problema pois entenderam que esse cenário realmente era indesejável para o usuário. O trecho registrado foi o seguinte *“alteramos o modelo proposto e removemos a relação que estava causando o problema de marcação, pois esse cenário pode trazer consequências indesejadas para o postador da foto”*. As outras duas duplas não chegaram a efetivamente fazer a alteração no modelo para corrigir o problema, mas registraram que puderam visualizá-lo e que em um contexto real de desenvolvimento, teriam corrigido a modelagem para remover o problema da marcação de fotos, por entenderam que esse cenário é indesejável. Por exemplo, a dupla P3-P6 registrou: *“podemos verificar o problema da marcação de fotos após a interação com o SIGN. O modelo deveria ser corrigido para remover esse cenário”*.

5.3.1.2 Modelagem de um sistema a partir do zero

Após analisarem um modelo já pronto, pedimos aos participantes que executassem a modelagem de um sistema de legado digital, semelhante ao Google Inactive Account Manager a partir do “zero”, ou seja, eles deveriam tomar todas as decisões de como modelar esse sistema usando as dimensões previstas pela SIGMa-dl. Durante a realização das tarefas 3.2.1 e 3.2.2, participamos como observadores, registrando o processo de uso do SIGMa. Nesta observação, pudemos observar e registrar que as duplas sempre recorriam ao SIGN para ver o efeito da modelagem que estavam fazendo e que, sempre que não obtinham o resultado esperado, voltavam ao modelo para fazer as alterações que julgavam ser necessárias. O objetivo desta observação foi verificar a forma de uso do modelo proposto sem intervir no processo de uso do modelo. Esse fato foi comprovado ao analisar os registros feitos pelas duplas. A dupla P1-P4 registrou que *“o uso do*

SIGN foi fundamental para a modelagem. Sempre que fazíamos alguma alteração no modelo, recorriamos ao SIGN para ver os efeitos (dessa alteração)". A dupla formada por P2-P5 registrou as mudanças que fizeram no modelo: *"fizemos várias alterações no modelo após interagir com o SIGN, principalmente nas relações"*.

Além de poder observar o processo de uso do modelo, pudemos verificar ainda nos registros feitos pelos participantes as divergências que tiveram em relação à modelagem. Por exemplo, a dupla P2-P5 registrou *"não concordamos inicialmente sobre as relações que deveriam ser criadas. Porém ao verificar os cenários gerados pelo SIGN chegamos a um consenso sobre as relações que deveríamos criar"*. Já a dupla P3-P6 registrou que *"inicialmente discordamos sobre as mudanças e relações que deveríamos criar, mas fomos modelando aos poucos e verificando as consequências usando o SIGN, o que nos permitiu chegar a um consenso"*.

Os modelos criados por cada dupla foram copiados ao final do teste para análise posterior. Nesta análise, pudemos constatar que duas duplas fizeram um modelo muito parecido. Apesar de semelhantes, a dupla formada por P2 e P5 criou um modelo mais simplificado que a dupla formada por P3 e P6, que fez um modelo bastante detalhado da aplicação. A dupla P2-P5 optou por não representar algumas ações recorrentes (como por exemplo, a ação editar um documento que é potencialmente infinita). Eles registraram que *"essas ações são menos problemáticas para o usuário, então decidimos não fazer a modelagem"*. Já a dupla P3-P6 modelou todas as consequências previstas. Foi interessante notar que as dimensões usadas e não usadas pelas duplas P2-P5 e P3-P6 foram idênticas (por exemplo, as duas duplas definiram dois papéis e não utilizaram a dimensão "Grupos", usaram a dimensão "Posse" e modelaram a "Mudança de Posse" quando o usuário se torna inativo). O modelo gerado por P2-P5 e por P3-P6 foi bem próximo do modelo que nós tínhamos gerado para este sistema. Já a dupla formada por P1 e P4 criou um modelo semelhante ao das outras duplas, porém utilizou a dimensão "Grupos" para categorizar os usuários que poderiam "ler", "ler e editar" ou "ler, editar e excluir", não utilizou a dimensão "Posse" e por conseguinte não utilizaram a "Mudança de Posse". Ao analisarmos a modelagem, a dupla criou um papel que nomearam "OutroUsuario", e definiram os grupos com base neste papel. Inclusive não criaram nenhuma ação para o papel em si (pois o papel foi definido unicamente para a composição dos grupos que foram criados). O restante da modelagem seguiu semelhante às das outras duplas e os cenários gerados foram consistentes com os das outras duas duplas (com exceção da mudança de posse que não foi modelada e, é claro, das ações serem executadas pelos grupos e não por um papel isolado). Esta análise forneceu indícios interessantes sobre a expressividade da linguagem de modelagem, como a possibilidade de modelar o sistema por completo e a similaridade entre os modelos gerados. Apesar

da modelagem feita pela dupla P1-P4 ter sido diferente das modelagens feitas pelas duplas P2-P5 e P3-P6, ambas representaram a situação sob análise, ou seja, os modelos das três duplas retrataram os cenários válidos antes e após a inatividade de um usuário. Embora não tenham percebido nenhum cenário indesejável, os participantes analisaram os cenários e conferiram que estavam dentro do previsto. Os modelos criados por todas as duplas continham todos os aspectos de modelagem esperados (como os papéis ou grupos previstos e suas ações). No entanto, houve dificuldade no uso das dimensões mudanças e relações, o que sugere que deve-se investigar se estão adequadas e se modificações devem ser feitas. O uso dessas dimensões, no entanto, foi facilitado com o uso conjunto do SIGN, já que os participantes puderam experimentar os impactos de cada decisão de modelagem que realizavam.

Terminada a etapa de execução, conduzimos então a parte final da avaliação, que tinha como objetivo levantar as impressões e percepções dos participantes sobre a SIGMa-dl e o SIGN. Na subseção seguinte apresentamos os resultados obtidos.

5.3.1.3 Percepções dos participantes sobre o entendimento da SIGMa-dl e utilidade do SIGN

Na seção final da avaliação, buscamos indicadores para endereçar as questões de avaliação 1 e 2. Os participantes responderam inicialmente a um questionário com questões abertas. O objetivo do questionário era registrar o entendimento individual sobre a expressividade da SIGMa-dl e sobre a utilidade do SIGN. Como a resposta aos questionários foi individual, não permitindo o confronto de ideias entre os participantes, optamos por realizar ainda um grupo focal, onde buscamos identificar os pontos conflitantes no entendimento dos participantes. O questionário teve um caráter mais abrangente com diversas perguntas abertas, enquanto que o grupo focal continha temas mais restritos, endereçando pontos específicos que poderiam trazer discordâncias entre os participantes. Os pontos tratados como tema do grupo focal foram elencados a partir das percepções que tivemos durante a etapa de reengenharia de sistemas colaborativos existentes. Listamos abaixo os principais resultados obtidos na análise dos questionários respondidos. Novamente as falas de cada participante são realçadas entre aspas e em itálico e foram fielmente transcritas.

As perguntas 1 e 2 foram direcionadas à SIGMa-dl e buscaram levantar a percepção dos participantes sobre a expressividade da SIGMa-dl, endereçando a QA1. A pergunta 1 indagava aos participantes se as dimensões previstas pela SIGMa-dl foram adequadas e suficientes para a modelagem do sistema previsto nas tarefas 3.2.1 e 3.2.2 da execução da avaliação. O objetivo desta pergunta foi identificar o entendimento

dos participantes sobre as dimensões previstas pela linguagem. Assim, estávamos particularmente interessados em verificar se os participantes identificaram algum ponto que não pode ser representado pela SIGMa-dl, influenciando negativamente na sua expressividade (QA1). Todos os participantes responderam que as dimensões foram suficientes e adequadas para a modelagem do sistema. P5 complementou dizendo que *“para cada detalhe de modelagem que foi pensado, havia uma dimensão que permitia modelá-la”*. Já P1, disse *“consegui criar o modelo usando as dimensões previstas, embora as dimensões mudanças e relações sejam confusas num primeiro momento”*. P3 disse que *“o mapeamento entre a situação real e as dimensões previstas não é imediato, principalmente em relação às dimensões mudanças e relações, mas foi possível modelar o sistema com as dimensões previstas”*.

Em seguida, na pergunta 2, os participantes responderam se sentiram alguma dificuldade em modelar o sistema previsto de acordo com as dimensões previstas pela linguagem de modelagem. Em suas palavras, P6 relatou que *“as mudanças e relações se mostraram um pouco confusas em um primeiro momento. Mas ao interagir com o SIGN as dúvidas foram sanadas”*. Já P3, disse que *“a dificuldade se apresentou pela pouca experiência com a ferramenta”*. Todos os participantes apontaram em seus relatos que as dimensões mudanças e relações são difíceis de compreender.

As questões seguintes do questionário eram focadas no SIGN e tiveram o objetivo de ter uma apreciação da opinião dos participantes sobre a sua utilidade. Os participantes foram instruídos a considerar a sua experiência nas tarefas 3.1.1, 3.1.2, 3.2.1 e 3.2.2 para responder as questões seguintes.

A pergunta 3 questionava os participantes sobre o entendimento dos cenários gerados pelo SIGN. Todos os participantes relataram ter entendido todos os cenários gerados, com exceção de P2, que disse que *“uma pequena confusão entre as relações/mudanças e ações geraram feedbacks confusos”*.

Analisando as respostas dessa questão, obtivemos indícios da percepção dos participantes sobre a utilidade do SIGN. P2 relatou ter tido dificuldades de entendimento sobre as dimensões mudanças e relações. Novamente, como P6 já havia relatado na pergunta 2, o esforço para compreender o significado das dimensões “Relações” e “Mudanças” é grande.

A pergunta 4 indagava aos participantes se identificaram cenários potencialmente indesejados ao interagir com o SIGN. O objetivo dessa questão foi verificar se o SIGN foi capaz de permitir a descoberta de cenários potencialmente problemáticos no modelo durante a sua inspeção. Os participantes deveriam registrar se encontraram cenários problemáticos e quais foram. Apenas P3 respondeu que não tinha encontrado cenários problemáticos. Após o término do questionário, perguntamos a P3 o motivo de ter

não ter encontrado cenários problemáticos e ele relatou que *“ao responder a questão, lembrei apenas das tarefas 3.2.1 e 3.2.2, onde não vi nenhuma potencial inconsistência. Porém, se tivesse considerado as tarefas 3.1.1 e 3.1.2, teria percebido os cenários potencialmente problemáticos durante a inspeção”*. Todos os outros participantes relataram perceber a presença de cenários potencialmente indesejados na tarefa de inspeção de um modelo pronto.

Para aqueles que responderam ter encontrado cenários problemáticos na pergunta 4 (todos com exceção de P3, como já comentado acima), fizemos ainda outras perguntas relacionadas, cujos resultados são mostrados abaixo. Perguntamos aos participantes se a informação gerada pelo SIGN na identificação de cenários potencialmente problemáticos tinha sido útil, se o SIGN tinha mudado o seu pensamento sobre o modelo inspecionado ou que estava sendo desenvolvido e se eles fizeram alterações no modelo ao perceber que haviam cenários indesejados ou incorretos. Todos os participantes argumentaram que a inspeção dos cenários proporcionada pelo SIGN traz informações que os auxiliaram a analisar os potenciais problemas contidos na modelagem. Em seu comentário, P5 relatou *“vi que faltava[sic] configurações e realizei o ajuste necessário em uma relação para que as mudanças causadas pela ação configurar inatividade pudessem aparecer no cenário”*, se referindo às tarefas 3.2.1 e 3.2.2. Ao analisar a fala de P5, pudemos notar que a inspeção dos cenários gerados pelo SIGN mudou o pensamento de P5 sobre o modelo que estava construindo, mudando assim o seu pensamento original. P2 argumenta que *“pudemos repensar a modelagem feita com base nas dimensões e revê-la, principalmente no sistema da rede social, onde removemos a relação que dava origem ao cenário problemático de marcação de fotos. No sistema de legado digital, tivemos que apagar algumas ações e mudanças criadas de forma errônea”*. Já P6 relatou *“ao explorar os cenários, notamos que após a ação se tornar inativo, o usuário continuava podendo executar ações sobre seu documento. Essa situação não faz sentido, então removemos a relação que estava causando esse erro”* (se referindo às tarefas 3.2.1 e 3.2.2). P3 disse *“alteramos o modelo do legado digital após verificarmos que o espaço de tempo inativo não tinha nenhum cenário”*. P4 registra *“ao identificar o erro durante a inspeção de um cenário, procurei corrigir a inconsistência e isso levou a uma modelagem mais simples”*. Finalmente, P1 disse que *“visualizamos alguns loops e propomos[sic] algumas simplificações na modelagem”*. P1 completou dizendo que *“a visualização dos cenários gerados permitiu excluir algumas relações, gerando uma solução mais simples”*. No entanto, P1 não deixou claro o que nos cenários, gerou mudanças em seu pensamento.

Diferente da pergunta anterior, na pergunta 5 questionamos os participantes se o SIGN foi capaz de gerar algum cenário que eles não tinham previsto durante a execução

das tarefas. O objetivo desta questão foi identificar o potencial do SIGN de mostrar ao projetista situações imprevistas na modelagem. P2 relatou que *“ao analisar apenas o modelo da rede social não consegui encontrar nenhum problema para o usuário final, mas ao interagir com o SIGN o problema da marcação de fotos ficou evidente.”* P5, que formou dupla com P2, também fez referência a identificação do problema da marcação de fotos no modelo da rede social e registrou que *“após termos identificado o cenário problemático, voltamos ao modelo e removemos a relação que estava dando origem a ele”*. P1 e P4 também disseram ter percebido o cenário da marcação de fotos ao inspecionar o modelo da rede social. Em alusão à modelagem do sistema legado, P5 disse que *“ao interagir com o SIGN e inspecionar os cenários, ficou claro que faltava mapear algumas mudanças e relações”*. Já o participante P3 voltou a fazer referência à modelagem do sistema de legado digital, onde disse *“pela pouca experiência com a ferramenta, no SIGN, foi observado que no espaço de tempo inativo, não colocamos um nó independente na árvore. Ao perceber esse erro, voltamos ao modelo e fizemos a correção necessária na relação”*. P6, que fez dupla com P3 relatou o mesmo caso, dizendo *“ao ver que não havia nenhum cenário no tempo inativo, voltamos ao modelo e fizemos a correção necessária na relação”*.

Por fim, pedimos que os participantes registrassem os pontos positivos e negativos do SIGMa. Sobre os pontos positivos, P1 registrou que *“as dimensões previstas ajudam na abstração do problema proposto e o SIGN mostra de forma visual as modelagens e cenários”*. P2 destaca como ponto positivo do modelo ser *“baseado nas ações do sistema”*. P3 apresenta como ponto positivo *“a ferramenta auxilia na modelagem de sistemas colaborativos no que tange a inconsistências de permissões em ações envolvendo terceiros”*. P4 diz que, como ponto positivo, o modelo *“oferece um meio/linguagem para descrever sistemas colaborativos”*. P5, que atua no meio profissional, registra que o *“modelo (é) consistente e robusto”*. P6, que assim como P5 também atua no meio profissional, diz que *“(o modelo) torna o custo de manutenção de software futuro menos custoso, afinal evita a implementação de cenários indevidos ou não desejáveis já no momento de modelagem”*. Outro ponto registrado por P6, em suas palavras, foi *“garante agilidade no processo de modelagem”*.

Em geral os comentários sobre os pontos positivos do modelo abordaram as dimensões previstas pela SIGMa-dl. P1 ressalta que elas, por si só, já auxiliam na reflexão do projetista sobre o modelo que está desenvolvendo. É interessante notar que esta afirmação de P1 contrasta com uma outra própria opinião desse participante, ao concluir a tarefa 3.1.1, em que registrou que *“o trabalho de compreender um modelo pronto e procurar por potenciais problemas no modelo é mecânico e tedioso”*. Essa diferença possivelmente pode ser atribuída ao fato de que naquela tarefa os participantes

inspecionaram um modelo já pronto, ou seja, não tiveram que raciocinar sobre um problema com as dimensões em mente. Já na tarefa seguinte, os participantes tiveram que modelar um sistema pensando nas dimensões previstas, fato este que pode justificar a mudança de opinião de P1 ao apontar os pontos positivos do modelo. P2 aponta como interessante a modelagem com base nas ações. Na resposta de P3 podemos destacar dois pontos interessantes. Primeiro ele diz que a ferramenta auxilia na modelagem de sistemas colaborativos. Depois ele fala de inconsistências envolvendo terceiros, fato este que ele poderia perceber com a ajuda do SIGN. Novamente encontramos evidências da utilidade do nosso modelo. P4 fala da vantagem de se ter uma linguagem de modelagem para sistemas colaborativos. Os participantes que atuam no mercado de trabalho (P5 e P6) apontaram pontos positivos interessantes do modelo. Em seus comentários, P5 e P6 apontaram que o modelo pode diminuir os custos de manutenção do sistema, já que evita a implementação de cenários indesejáveis. Esta afirmação, vinda de profissionais do mercado com boa experiência e com conhecimento de outros modelos, fortalece os indícios da expressividade da nossa linguagem de modelagem e da utilidade do SIGN. Outro ponto registrado por P6 foi a agilidade no processo de modelagem.

Sobre os pontos negativos, P1 aponta que *“a modelagem se torna repetitiva ao modelar ações e relações”*. P2 relata que *“a time line no protótipo é confusa”*. P3 registrou que *“é difícil compreender o conceito das relações e como se deve modelar as mudanças”*. Já P4 diz *“a linguagem poderia oferecer meios para descrever os sistemas de forma mais amigável, talvez utilizando orientação a objetos. Por exemplo, os elementos da linguagem (ações, relações, mudanças, etc) poderiam utilizar mecanismos de herança para facilitar a escrita e descrição do modelo”*. P5 diz *“achei um pouco confuso o conceito mudança/relação”*. Finalmente, P6 registra que *“o modelo é muito textual, o que de certa forma torna o processo muito detalhista e minucioso. Talvez o uso de métodos mais visuais permitam um melhor aproveitamento do modelo”*. Todos os participantes registraram ainda a dificuldade em compreender as dimensões mudanças e relações como ponto negativo do modelo.

Analisando os pontos negativos levantados, observamos que a sobrecarga de trabalho que a modelagem exige pode ser um entrave na aceitação do modelo, como nas falas de P1 e P4. P2 comenta que a time line ao usar o protótipo é confusa. A observação de P5 já é diretamente relacionada à linguagem de modelagem. Embora este participante tenha tido dificuldades em compreender o significado das dimensões mudanças e relações e possivelmente de como modelar um sistema usando-as, esta dificuldade já era esperada por ser o primeiro contato com a linguagem e por não terem um mapeamento direto com o mundo real. Termos como “papéis”, “grupos” e “ações” são comuns em todas as propostas para a modelagem de sistemas colaborativos, mas o

mesmo não acontece para as mudanças e relações. Essas duas dimensões, no entanto, realmente não são intuitivas para o projetista e exigem uma experiência razoável para possam ser usadas de forma correta na modelagem do sistema. Por fim, P6 registra que a modelagem deveria ser feita de forma visual, como em uma ferramenta CASE (e não textual). É certo que representações visuais podem trazer ganhos para o entendimento de modelos e esta é uma característica que vale a pena ser investigada em desenvolvimentos futuros do modelo.

Após terminarem de responder os questionários individuais, conduzimos um grupo focal com o objetivo de verificar as divergências e concordâncias entre as percepções de cada participante. Outro objetivo era fomentar a troca de ideias e identificar algum ponto que não tivesse sido registrado nos questionários. A discussão do grupo focal foi registrada em áudio, com o consentimento de todos os participantes. Para enriquecer a leitura, apresentamos abaixo a transcrição fiel das falas de cada participante em itálico e entre aspas sempre que uma opinião se destaca (quando apenas um participante opina ou quando um participante discorda do outro).

Iniciamos o grupo focal pedindo que os participantes falassem e discutissem sobre as dificuldades que tiveram ao modelar o sistema de legado digital. Os participantes disseram que tiveram sim dificuldades, mas que elas se deram principalmente pela falta de experiência com a linguagem. O participante P6 relata que *“num primeiro momento tem uma dificuldade, mas ao fazer a primeira vez ficou mais fácil”*, fato este que foi ratificado por todos os outros participantes. Em seguida, P4 levantou a dificuldade em compreender o significado das dimensões “Mudanças” e “Relações”. Todos os participantes concordaram, mas relataram também que após o uso da linguagem de modelagem conseguiram entender o significado dessas dimensões e que daí em diante *“ficou mais fácil”*, fala esta de P4.

Em seguida, pedimos que os participantes falassem e discutissem abertamente sobre os cenários que foram gerados pelo SIGN durante a realização das tarefas. As discussões espontaneamente foram para o lado da identificação de cenários problemáticos nas modelagens. P2 disse que, depois de ter identificado o possível problema da marcação de fotos ao inspecionar a modelagem parcial do Facebook, *“voltou e arrumou e a inconsistência sumiu”*. Quando perguntados se a inspeção dos cenários os fizeram refletir sobre o modelo, todos os participantes disseram que alternaram diversas vezes entre a modelagem e o SIGN para fazer alterações e correções e que esta interação os fez refletir sobre as decisões que haviam tomado durante a modelagem. P5 disse inclusive que *“eu tinha desenhado no papel mas ela (ação) não estava lá (no cenário), e aí eu vi que eu precisava alterar o modelo”*.

Prosseguimos pedindo que os participantes discutissem sobre a adequação do

SIGMa ao desenvolvimento de sistemas colaborativos reais. P4 diz *“eu acho que ele (o modelo) pode permitir uma forma de você especificar formalmente uma situação colaborativa em que outras ferramentas não são tão simples assim de conseguir isso, mesmo em UML, que tem muitos documentos mas a maioria deles o pessoal não usa muito”*. P3 diz que *“especificamente para softwares colaborativos ele (o modelo) auxilia bastante, mas pra alguns outros tipos de software ela também pode ajudar, inclusive na área de computação inteligente”*. Um ponto que consideramos interessante levantado por P4, quando diz *“na UML o projetista tem que ter a visão toda (do modelo) e tem que fazer todas as ligações do grafo (em alusão aos possíveis caminhos que um cenário pode assumir). Na sua ferramenta o projetista pensa só no nó e a ferramenta (SIGN) depois vai costurar tudo”*. E ele continua *“talvez mostrar esse grafo “costurado” seja mais interessante”*. Porém o próprio participante reconhece que a visualização gráfica dos cenários (usando grafos) é um problema a parte. Seguiu-se então uma discussão sobre este tema e os participantes concluíram que, mesmo mostrado como um grafo, o projetista necessariamente teria que querer explorar todos os “nós” gerados e que isso poderia ser trabalhoso.

Finalmente, solicitamos aos participantes que fizessem comentários gerais englobando pontos positivos e negativos do modelo ou mesmo do protótipo com o objetivo de verificar se as opiniões eram convergentes ou divergentes. P5 salientou que o protótipo deveria *“emitir a documentação do sistema, com árvore expandida, papéis, etc”* com o intuito de fornecer ao projetista um documento formal de modelagem. P5 ressaltou ainda que seria interessante que o modelo contemplasse a ordem das ações (se uma é executada antes da outra, se são executadas em paralelo, etc). P4 interveio dizendo que *“esses são detalhes de implementação e que complicariam o modelo enquanto ferramenta epistêmica”*. Essa divergência entre P4 e P5 foi observada pelos demais participantes e após a fala de P4, todos os demais concordaram que acrescentar detalhes de implementação no modelo não era desejável.

Em geral as respostas do grupo focal foram consistentes com aquelas fornecidas no questionário. No entanto, pudemos perceber detalhes que não haviam sido relatados no questionário, o que enriquece o nosso entendimento sobre a opinião dos participantes, como por exemplo as alusões à linguagem UML. Além disso pudemos confirmar que os participantes frequentemente alternaram entre a modelagem e o SIGN buscando corrigir falhas de modelagem e que a inspeção dos cenários gerados causou novas reflexões sobre o modelo que estavam criando.

5.4 Revisitando as Questões de Avaliação 1 e 2

Após a apresentação das avaliações conduzidas e dos resultados observados, discutimos os indicadores gerados para endereçar as questões de avaliação QA1 e QA2, colocadas no início deste Capítulo.

Para responder a QA1 (*as dimensões previstas na SIGMa-dl são suficientes para modelar as mudanças que podem ocorrer no tempo ao longo do uso do sistema?*) levamos em consideração principalmente os resultados da reengenharia de sistemas existentes e da avaliação com projetistas reais. Na reengenharia dos dois sistemas escolhidos, que são usados em diferentes contextos, pudemos descrever todas as decisões de projeto necessárias usando as dimensões da SIGMa-dl. Além disso, ao analisar os resultados da avaliação com projetistas reais, notamos que eles relatam a capacidade da SIGMa-dl em representar as situações que foram apresentadas na avaliação.

Ao analisar a SIGMa-dl segundo o *framework* CDN, pudemos observar indicadores sobre o seu custo de aprendizado. Pudemos notar que as dimensões mudanças e relações podem não ser naturais para o projetista, causando assim dificuldade no seu entendimento. Esse fato foi confirmado na avaliação com projetistas, onde por diversas vezes eles relatam dificuldades no entendimento dessas dimensões.

Ao refletirmos sobre a QA2 (*o SIGN gera cenários que levam o projetista a refletir sobre o modelo feito (assumindo seu papel de ferramenta epistêmica), possivelmente levando-o a alterar o modelo até que esteja satisfeito?*), levamos em consideração os resultados da reengenharia de sistemas existentes, mas principalmente os resultados dos testes com projetistas reais. Antever todos os cenários possíveis em uma aplicação colaborativa não é uma tarefa fácil para o projetista. Ao analisar os resultados da avaliação com projetistas, pudemos perceber que a inspeção dos cenários gerados pelo SIGN efetivamente foi capaz de alterar o pensamento inicial dos projetistas sobre o sistema, fazendo com que eles voltassem ao modelo para realizar a mudança necessária, em alguns casos.

Os resultados da reengenharia de sistemas existentes e da avaliação com projetistas reais, apresentados neste capítulo, são preliminares, mas mostram indicadores positivos sobre a expressividade da SIGMa-dl e o potencial do SIGN em fomentar novas ideias na mente do projetista. No entanto, novas avaliações envolvendo modelos maiores e outros domínios, serão importantes para consolidar os resultados obtidos até o momento.

Capítulo 6

Conclusões

Em nossos estudos iniciais, percebemos que os usuários envolvidos em um sistema colaborativo onde as ações de um determinado usuário têm impacto em outros estão sujeitos a experimentar problemas de ordem social. Peguemos o exemplo do Facebook, que analisamos ao longo de toda a pesquisa. Um determinado usuário pode postar uma foto esperando que apenas seus amigos a vejam, fazendo as devidas configurações. No entanto, as ações destes amigos alheias à sua vontade podem fazer com que a foto deixe de ser restrita a este grupo e passe a ser vista por pessoas indesejadas, sem o conhecimento do usuário postador. Neste caso, o sistema permitiu que o usuário experimentasse um problema social (ele pode ter passado por constrangimento, por exemplo, já que pessoas indesejadas tiveram acesso à sua foto). De fato, como mostramos no nosso estudo de caso do Facebook, descrito na Seção 2.1, esta foi uma situação problemática real enfrentada por seus usuários na época do estudo. Como observado naquela ocasião, boa parte dos usuários não sabiam que isso poderia ocorrer. Ao utilizar o nosso modelo, o projetista dessa aplicação poderia ter antevisto este problema e então ter refletido sobre a melhor maneira de lidar com ele no sistema. Este problema, no entanto, não é específico do Facebook, mas pode estar presente nos sistemas em que diferentes pessoas interagem através de um sistema computacional colaborativo. Isso se dá pelo fato de que as ações de um usuário podem depender das ações de outros usuários e principalmente podem trazer consequências para o sistema como um todo.

Os diversos trabalhos propostos na literatura foram fonte de inspiração para a nossa proposta, a começar pelo trabalho de Prates [1998]. Já naquela época a autora previu a necessidade e os benefícios que a exploração de cenários futuros poderia trazer para a atividade de projeto. Apesar de o gerador de cenários (cenógrafo) ter sido definido apenas conceitualmente naquele trabalho, ele foi a principal fonte de inspiração desta pesquisa.

Ao explorarmos a literatura, encontramos modelos e *frameworks* propostos para a modelagem de sistemas colaborativos, como o GTA [Veer et al., 1996] ou HAMSTERS Martinie et al. [2011]. No entanto, nenhum deles permite que o projetista experimente (ou simule) os cenários que pretende disponibilizar aos usuários finais. Como já apresentamos no Capítulo 3, a única exceção à essa regra é o CTT [Paternò, 1999; Paternò et al., 2001]. Ainda naquele capítulo, fizemos uma discussão extensa das diferenças do CTT para o SIGMa. Sumarizamos a seguir as principais diferenças discutidas naquele capítulo. A primeira diferença do CTT em relação ao SIGMa diz respeito à descrição das tarefas. No CTT, o projetista deve necessariamente descrever todas as tarefas previstas para o sistema e todos os relacionamentos entre elas. Se isso é necessário, para usar o CTT o projetista terá que conhecer todos os cenários possíveis antes de criar o seu modelo. Neste sentido, o CTT se torna apenas uma forma de materializar de forma gráfica, o entendimento que o projetista já possuía do sistema. É evidente que essa representação tem o seu valor para sistemas grandes, com muitas tarefas, mas de fato o CTT não gera novos cenários a partir de decisões que possam ser tomadas pelos usuários. Já o SIGMa permite que os cenários sejam gerados a partir de uma descrição isolada das consequências de cada ação. Assim, o projetista não precisa definir (e antever) todos os cenários que podem ser gerados através das combinações das diversas ações de cada usuário (papel ou grupo) no sistema. Assim, o SIGMa tem o potencial de chamar a atenção do projetista para cenários que não foram antevistos em tempo de projeto, possibilitando que ele reflita sobre as decisões que tomou na definição das consequências de cada ação. A segunda diferença é relacionada ao trabalho que o projetista tem em definir ações iguais, ou que têm o mesmo impacto no sistema. No CTT cada uma das instâncias de uma ação deve ser feita de forma isolada. E quando uma alteração é necessária, ela não se propaga para as “cópias”. Além de aumentar a possibilidade de erros, isso traz trabalho extra ao projetista e pode gerar inconsistências no modelo. No SIGMa, o projetista define uma ação e as consequências que ela traz ao sistema uma única vez. Em qualquer lugar que aquela ação apareça, em qualquer cenário, ela terá o mesmo comportamento. Caso seja necessário alterar esse comportamento, essa alteração é feita uma única vez e, ao gerar os cenários, o SIGMa automaticamente considera a alteração em cada instância da ação. A terceira principal diferença diz respeito aos espaços de tempo (contextos) pelos quais uma aplicação colaborativa pode passar. Como apresentamos, o CTT não prevê a modelagem do sistema em termos dos diferentes contextos de aplicação. Por não ter essa característica, os modelos CTT que envolvem diferentes contextos podem ser difíceis de serem criados e ainda mais de serem simulados. Vale ressaltar novamente que o no CTT o termo “temporal” diz respeito aos relacionamentos que podem existir entre tarefas (concorrência,

habilitação, etc) e não aos diferentes contextos pelos quais uma aplicação colaborativa pode passar.

Na seção seguinte apresentamos e discutimos as principais constatações que fizemos sobre o SIGMa, bem como suas limitações, e em seguida apresentamos algumas considerações sobre as avaliações realizadas. Encerramos o capítulo apresentando as contribuições da nossa pesquisa de doutorado, assim como com trabalhos futuros que podem dar continuidade a esta pesquisa.

6.1 Constatações e limites do SIGMa

Para verificar se o conjunto por nós definido era adequado ou suficiente para modelar um sistema colaborativo, conduzimos três avaliações iniciais detalhadas no capítulo anterior. Nessas avaliações, encontramos indicadores de que o conjunto de dimensões definido foi suficiente para modelar os sistemas sob análise. Apesar dos indicadores positivos sobre a expressividade da SIGMa-dl que pudemos observar a partir de nossas avaliações, observamos também outros indicadores que esta expressividade pode gerar um custo para o usuário, tanto na análise que realizamos do SIGMa usando o *framework* CDN quanto na avaliação com projetistas reais. Pudemos verificar que as dimensões mudanças e relações não são de fácil entendimento e que não têm um mapeamento direto para os problemas reais em que sistemas colaborativos são aplicados ou mesmo para outros modelos. Apesar dessa limitação, essas duas dimensões são preponderantes para a posterior geração de cenários pelo SIGN. Em última análise são essas dimensões que permitem ao projetista descrever o comportamento do sistema, em termos das consequências das ações de cada ator (papel ou grupo) envolvido no sistema. Assim, é necessário investigar se haveria forma melhor de representar essas dimensões. Neste sentido, um caminho interessante de ser investigado é a criação de uma ferramenta de modelagem integrada, onde as definições de cada instância pudessem ser mais intuitivas e que, no momento da definição de cada relação, o projetista fosse informado de quantos e quais cenários estão sendo criados pela definição que foi feita. Ao fazer isso o projetista poderia ter a visualização das consequências da relação e o seu entendimento poderia ser melhorado.

Para que fosse possível avaliar o SIGMa implementamos as funcionalidades de modelagem e geração de cenários através de um protótipo funcional. O protótipo foi construído com o objetivo de permitir um uso prático do SIGMa. Porém, foi criado sem o rigor dos critérios de qualidade das áreas de IHC e Engenharia de Software. Neste sentido, é necessário investigar a melhor forma de permitir a modelagem do sistema

usando as dimensões da SIGMa-dl através de uma ferramenta computacional. Um dos participantes durante o grupo focal chegou a relatar que a modelagem do sistema deveria ser mais visual, ou seja, deveria se aproximar das modelagens proporcionadas por ferramentas CASE. Outro fator que deve ser investigado em relação ao protótipo é a representação dos cenários gerados. A visualização dos cenários em forma de árvore, como implementada no protótipo, pode não ser a mais adequada. Assim, é necessário investigar se é possível representar os cenários de uma outra forma mais indicada para que o projetista possa inspecionar. Uma das possibilidades neste sentido seria criar descrições textuais dos cenários gerados para que o projetista pudesse inspecionar, complementando assim o seu entendimento do funcionamento do sistema. Outra possibilidade é investigar se a representação dos cenários em forma de grafo traria benefícios maiores que a representação em árvore. Neste caso devem ser verificados os trabalhos que lidam com a visualização de grafos. Finalmente, ao identificar a forma mais adequada de representar os cenários, idealmente ela deveria ser implementada em uma ferramenta de modelagem integrada, que permitisse ao projetista ver em tempo real os cenários criados (quantos e quais) a partir das definições que está fazendo.

Por mais que os cenários sejam gerados de forma satisfatória para inspeção dos projetistas, ainda será primordial que estes inspecionem cada cenário gerado, na busca de potenciais problemas para os usuários finais. Considerando que os cenários que podem ser gerados em uma aplicação colaborativa são potencialmente infinitos, se torna muito importante encontrar uma forma de categorizar ou priorizar os cenários gerados (através de filtros e da marcação de cenários que já foram inspecionados), no sentido de diminuir o número de cenários que devem ser inspecionados pelo projetista. Nesta direção implementamos no protótipo como prova de conceito um componente que faz uma análise semântica dos cenários gerados. Para analisar criamos apenas a regra sobre o caso em que um usuário pode estar tendo acesso à dados de terceiros sem o seu conhecimento, como por exemplo no caso da marcação de fotos no Facebook (onde um usuário posta a foto para seus amigos, mas um desses amigos marca um terceiro que não faz parte do grupo de amigos, sem que o usuário que postou a foto saiba). Como resultado, na reengenharia do Facebook observamos que foi possível identificar esses cenários e gerar um aviso para o projetista na área do protótipo na qual são mostrados os avisos sobre inconsistências sintáticas. Como dissemos, implementamos essa regra para ver se era possível identificar cenários que o projetista deve inspecionar. Porém, é necessário investigar se seria possível gerar um conjunto de heurísticas que representassem regras semânticas de como caracterizar e priorizar os cenários, para que este componente semântico fosse útil. Por exemplo, o projetista poderia criar uma regra para que o SIGN mostrasse apenas os cenários que possuem ações potencialmente

infinitas ou que mostrasse apenas os cenários em que um determinado papel possui ações disponíveis. Em uma implementação futura do SIGN, deve-se investigar quais tipos de regras podem ser definidas, como serão agregadas ao SIGN (através de uma base de conhecimentos, por exemplo), e como serão implementadas em uma futura ferramenta computacional.

Voltando à Engenharia Semiótica, o SIGMa é voltado para sistemas customizáveis, ou seja, sistemas onde as decisões de um usuário podem impactar no sistema. Contudo, é importante realizar novos estudos para o campo de aplicações extensíveis, que não é abordado pelo SIGMa.

6.2 Considerações sobre as avaliações realizadas

O conjunto de domínios considerados por nossas avaliações foi relativamente pequeno, dadas as diversas áreas de aplicação de sistemas colaborativos. Apesar de termos colhido bons indicadores sobre a expressividade da SIGMa-dl e da utilidade do SIGN, é necessário que o SIGMa seja avaliado em outros domínios de aplicação. Nossas avaliações iniciais mostraram que a SIGMa-dl possui expressividade suficiente para modelar os sistemas analisados (tais como o Facebook ou o *Google IAM*), mas é necessário fazer uma investigação mais aprofundada da aplicação do SIGMa em sistemas colaborativos de outros domínios (que não aqueles tratados nas nossas avaliações) para verificar a real expressividade da SIGMa-dl. Estas avaliações poderiam seguir o mesmo modelo seguido neste trabalho, onde realizamos a reengenharia de sistemas. Neste caso, o sistema que passará pelo processo de reengenharia deve ser antes inspecionado e os cenários correspondentes (ou o seu fluxo de funcionamento) devem ser estabelecidos, para que possam ser comparados aos gerados pelo SIGMa. Idealmente neste caso devem ser inspecionados e modelados sistemas maiores, para investigar o SIGMa nestes casos. Seria interessante ainda realizar a reengenharia de um mesmo sistema colaborativo existente, usando diferentes modelos (SIGMa, CTT e UML - indicada pelos participantes) para fazer uma análise mais detalhada dos custos benefícios e diferenças entre elas.

Nessa tese apresentamos uma avaliação inicial do SIGMa com projetistas reais. Nessas avaliações coletamos indicadores positivos sobre a expressividade da SIGMa-dl e a utilidade do SIGN em fomentar novas ideias na mente do projetista enquanto exerce a sua atividade intelectual de projeto. Seria importante e interessante realizar novas avaliações com projetistas, porém envolvendo um sistema mais complexo para observar o comportamento do SIGMa nesse contexto de modelagem.

No teste que envolveu projetistas reais, eles não modelaram um sistema em que realmente estivessem trabalhando, o que pode gerar um compromisso menor com o modelo, pois o mesmo não seria mais usado. Idealmente deveria ser conduzida uma avaliação em contexto real, como discutimos a seguir.

Por fim, nossa avaliação com projetistas reais foi realizada em ambiente controlado, quando o ideal é que ela fosse conduzida em um ambiente real de desenvolvimento. Avaliações como as que conduzimos se justificam por terem um menor custo de aplicação e por isso são desejáveis antes de uma avaliação de escala maior. Contudo, para que o SIGMa pudesse ser avaliado em um desenvolvimento real, seria necessária a participação de uma corporação interessada em conduzir tal avaliação, o que por si só pode ser um grande desafio. Se isso for possível, pode-se obter indicadores da aplicação real do SIGMa. Serão levantadas questões sobre a expressividade da SIGMa-dl para o sistema sendo desenvolvido e sobre a real utilidade que a antecipação dos cenários futuros traz ao projetista de um sistema colaborativo. A partir dessas observações, novas melhorias e refinamentos podem ser propostos para o SIGMa, aumentando assim o seu potencial em apoiar o projeto de sistemas colaborativos.

Um ponto que deve ser discutido em qualquer modelo é a relação entre seu custo e benefício. Apesar de não termos endereçado diretamente essa relação em nossas avaliações, pudemos notar indícios isolados de cada um desses fatores. Em relação ao custo, todos os projetistas relataram ter tido dificuldades em compreender as dimensões mudanças e relações da SIGMa-dl. Percebemos que há um custo para o correto aprendizado dessas dimensões. Outro ponto que indica a percepção do custo de modelagem são as indicações de melhorias para a SIGMa-dl (por exemplo quando um dos projetistas citou que seria bom que a linguagem pudesse ter características de orientação a objetos, como herança). Por outro lado, observamos diversos indícios dos benefícios trazidos pela inspeção dos cenários gerados pelo SIGN. Todas as duplas relataram que a inspeção dos cenários mudou o seu pensamento em relação ao modelo que estavam inspecionando/criando. Apesar dos indicadores isolados, é necessário estudar a relação entre custo e benefício trazidos pelo SIGMa. A reengenharia de um mesmo sistema colaborativo utilizando diferentes modelos, como já dissemos acima, também poderia trazer indicadores para avaliar a relação entre o custo e o benefício do SIGMa.

6.3 Contribuições do trabalho

O projeto de sistemas colaborativos impôs aos projetistas novos desafios que não existiam no projeto de sistemas monousuário. Dentre esses desafios está o de compreender

todas as possibilidades de interação entre os atores envolvidos no sistema. Frequentemente as ações de um usuário impactam direta ou indiretamente nas ações que outros usuários podem executar dentro do sistema. Prever todas as combinações possíveis entre as ações dos usuários e suas consequências pode ser uma tarefa árdua para o projetista. Diversos modelos foram propostos para o projeto de sistemas colaborativos [Barbosa et al., 2007; Forbrig et al., 2014; Martinie et al., 2011; Mori et al., 2002; Paternò et al., 2001; Paternò, 2003]. Em sua maioria, os modelos existentes focam na modelagem de tarefas e nenhum deles trata da antecipação de cenários futuros de um sistema colaborativo ao longo do tempo. Tampouco chamam a atenção do projetista para este fato.

Ao conhecer os impactos das ações de um usuário em outros, o projetista pode refletir sobre a aplicação colaborativa que está criando sob uma nova perspectiva. O modelo SIGMa proposto neste artigo dá ao projetista essa capacidade. Ele permite que o projetista antevja cenários e os examine para ver se apresentam potenciais problemas que estarão presentes na sua aplicação final, mas em tempo de projeto. E ao antever esses problemas, o projetista pode se decidir por manter a modelagem, deixando que os problemas sejam experimentados e resolvidos pelos usuários no protocolo social, ou pode alterar a modelagem para eliminar ou modificar os cenários potencialmente problemáticos.

O SIGMa surge como uma nova ferramenta epistêmica, ou seja, o modelo não tem o objetivo de gerar uma solução para o problema, mas sim de causar novas reflexões sobre ele. Nossas avaliações mostraram o potencial do SIGMa em fomentar novos pensamentos na mente do projetista sobre o problema que estava modelando. Nesse sentido, nossa pesquisa contribui diretamente para a área de Engenharia Semiótica, acrescentando uma nova ferramenta epistêmica ao rol de ferramentas existentes. Nenhuma das ferramentas já propostas focam na antecipação de cenários de uma aplicação, como a nossa proposta.

Nossa pesquisa contribui ainda diretamente para a área de Sistemas Colaborativos e IHC. O SIGMa pode ser usado por projetistas de sistemas colaborativos, auxiliando-os na antecipação de cenários problemáticos. Identificar potenciais problemas em tempo de projeto pode ser benéfico, pois alterações no projeto tendem a ter um impacto financeiro muito menor, se comparadas com alterações feitas já nas fases posteriores do desenvolvimento de um software [Pressman & Maxim, 2014]. Além disso, a identificação de cenários que possam trazer situações de interação indesejáveis para o usuário, melhora a qualidade da interação e experiência dos usuários com o sistema.

Nossa proposta é um primeiro passo na direção de auxiliar o projetista de sistemas colaborativos na antecipação dos cenários que pretende disponibilizar aos seus

usuários finais. No entanto, é necessário refiná-la para que possa ter uma melhor aceitação por parte dos projetistas. Como identificado na avaliação segundo o CDN e nas avaliações com projetistas reais, foram identificadas dificuldades na compreensão das dimensões relações e mudanças, previstas na SIGMa-dl. Por isso é necessário investigar se melhorias podem ser feitas nesta direção e, em caso positivo, de quais tipos para que essa barreira possa ser vencida.

Além disso, seria importante realizar uma avaliação mais abrangente do SIGMa. Mas para que isso seja possível, é necessário rever o protótipo para criar um sistema de modelagem prático e robusto, projetado segundo os critérios de qualidade de IHC, para que a modelagem e exploração dos cenários seja feita de forma mais simples e natural para o projetista, potencializando assim o uso do modelo que estamos propondo. Idealmente, essa avaliação deverá ser conduzida em um projeto real de um novo sistema colaborativo. Porém encontrar uma empresa interessada em colaborar com essa avaliação pode ser um desafio.

Nosso trabalho traz uma abordagem original em relação a geração e apoio à exploração de cenários futuros de uma aplicação colaborativa. A partir do nosso trabalho, outras pesquisas podem propor soluções alternativas ou melhorias do modelo proposto nesta tese de doutorado e novas ideias podem surgir visando apoiar o projetista na antecipação dos cenários futuros de uma aplicação.

6.4 Trabalhos Futuros

Na seção 6.2, discutimos os pontos que merecem investigação posterior, porém relacionados às avaliações que foram conduzidas neste trabalho de doutorado. No entanto, apresentamos aqui outras possibilidades de avanço que podem trazer contribuições relevantes para este trabalho.

Neste trabalho não tratamos a temporalidade dos espaços tempo, i.e. as possibilidades de ocorrência de cada um ao longo do tempo. Consideramos que cada espaço de tempo acontece unicamente no tempo. Porém, é possível que dois ou mais espaços de tempo ocorram de forma concorrente, ou que tenham alguma interseção no tempo. É extremamente relevante verificar qual o impacto que relações temporais, como por exemplo as definidas por Allen [1983], trazem para o modelo aqui proposto. A partir dessa análise, podem ser feitas novas proposições de alterações no modelo, permitindo assim que sejam contemplados sistemas diversos.

Outro ponto que merece atenção é a hierarquia entre papéis e grupos possíveis no sistema. Neste trabalho de doutorado a hierarquia é representada pelas ações que

podem ser executadas por cada papel/grupo e por isso fica implícita no modelo. Seria interessante verificar qual o impacto que a possibilidade de definição explícita de hierarquia traria ao modelo. A hierarquia tem potencial para trazer melhorias, por exemplo, para a definição de regras de geração e classificação de cenários.

O SIGMa considera o sistema como um agente que não toma decisões, mas que tem um comportamento que é disparado pela ação de algum papel/grupo. Seria interessante investigar se o SIGMa teria ganhos ao permitir a modelagem de sistemas que tomam decisões, como sistemas adaptativos ou de recomendação. Ao permitir a modelagem de sistemas dessa forma, o SIGMa se aproximaria mais do dia a dia do projetista. Com essa mudança, o projetista poderia modelar, por exemplo, um sistema de *feed* de notícias, que toma decisões sobre o que mostrar ao usuário, sem a sua interferência.

Outra característica do SIGMa é que a relação entre artefato e um papel/grupo é representada simplificada pela dimensão de posse. No entanto, há outras relações possíveis (e.g. Smith & Erwin [2005]) que podem trazer ganhos para a geração de cenários e para a criação de regras semânticas para filtro e identificação de problemas em cenários gerados.

Finalmente, seria interessante investigar se os conceitos de *Behavior Driven Development* (BDD) [North, 2006] podem ser úteis para a proposição de mudanças no modelo, sobretudo nas dimensões apontadas como difíceis de entender (mudanças e relações) do SIGMa.

Referências Bibliográficas

- Ackerman, M. S. (2000). The intellectual challenge of cscw: the gap between social requirements and technical feasibility. *Hum.-Comput. Interact.*, 15(2):179--203.
- Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832--843.
- Barbosa, C. M. d. A. (2002). MetaCom-G*: Especificação da comunicação entre membros de um grupo. Dissertação de mestrado, Departamento de Informática, PUC-Rio.
- Barbosa, C. M. d. A. (2006). *Manas: uma ferramenta epistêmica de apoio ao projeto da comunicação em sistemas colaborativos*. Tese de doutorado, Departamento de Informática, PUC-Rio.
- Barbosa, C. M. d. A.; Prates, R. O. & de Souza, C. S. (2007). Identifying potential social impact of collaborative systems at design time. Em *Proceedings of the 11th IFIP TC 13 international conference on Human-computer interaction, INTERACT'07*, pp. 31--44, Berlin, Heidelberg. Springer-Verlag.
- Barbosa, S. & Silva, B. (2010). *Interação Humano-Computador*. Elsevier.
- Blackwell, A. & Green, T. (2003). Notational systems - the cognitive dimensions of notations framework. Em Carroll, J., editor, *HCI Models, theories, and frameworks: Towards a multidisciplinary science*, capítulo 5. Elsevier Science.
- da Silva, R. F. (2009). Manastool: Uma ferramenta computacional para apoio ao projeto da comunicação entre usuários em sistemas colaborativos. Dissertação de mestrado, Universidade Federal de Minas Gerais.
- da Silva, R. F.; Prates, R. O. & Salles, V. J. (2010). Smart: A computational tool for supporting the identification at design time of the social impact of collaborative systems. *Sistemas Colaborativos, Simpósio Brasileiro de*, 0:39--46.

- de Souza, C. S. (2005). *The Semiotic Engineering of Human-Computer Interaction (Acting with Technology)*. The MIT Press.
- de Souza, C. S.; Leitão, C. F.; Prates, R. O.; Amélia Bim, S. & da Silva, E. J. (2010). Can inspection methods generate valid new knowledge in hci? the case of semiotic inspection. *International Journal of Human-Computer Studies*, 68(1-2):22--40.
- de Souza, C. S.; Leitão, C. F.; Prates, R. O. & da Silva, E. J. (2006). The semiotic inspection method. Em *Proceedings of VII Brazilian symposium on Human factors in computing systems - IHC '06*, IHC '06, p. 148, New York, New York, USA. ACM Press.
- de Souza, C. S. & Leitão, C. F. (2009). Semiotic engineering methods for scientific research in hci. *Synthesis Lectures on Human-Centered Informatics*, 2(1):1--122.
- Forbrig, P.; Martinie, C.; Palanque, P.; Winckler, M. & Fahssi, R. (2014). Rapid task-models development using sub-models, sub-routines and generic components. Em Sauer, S.; Bogdan, C.; Forbrig, P.; Bernhaupt, R. & Winckler, M., editores, *Human-Centered Software Engineering SE - 9*, volume 8742 of *Lecture Notes in Computer Science*, pp. 144--163. Springer Berlin Heidelberg.
- Gill, A. J.; Vasalou, A.; Papoutsis, C. & Joinson, A. N. (2011). Privacy dictionary: a linguistic taxonomy of privacy for content analysis. Em *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pp. 3227--3236, New York, NY, USA. ACM.
- Green, T. & Blackwell, A. (1998). Cognitive dimensions of information artefacts: a tutorial. <http://www.ndirect.co.uk/~thomas.green/workstuff/papers/>.
- Green, T. R. G. (1989). Cognitive dimensions of notations. Em *People and Computers V*, pp. 443--460. University Press.
- Grudin, J. & Poltrock, S. (2013). *Computer Supported Cooperative Work*. The Interaction Design Foundation, Aarhus, Denmark.
- Martinie, C.; Palanque, P. & Winckler, M. (2011). Structuring and composition mechanisms to address scalability issues in task models. Em Campos, P.; Graham, N.; Jorge, J.; Nunes, N.; Palanque, P. & Winckler, M., editores, *Human-Computer Interaction - INTERACT 2011 SE - 40*, volume 6948 of *Lecture Notes in Computer Science*, pp. 589--609. Springer Berlin Heidelberg.

- Mori, G.; Paternò, F. & Santoro, C. (2002). Ctte: Support for developing and analyzing task models for interactive system design. *IEEE Trans. Softw. Eng.*, 28(8):797–813.
- North, D. (2006). Introducing BDD. *Better Software Magazine*, Volume 03:pp. 26–31.
- Paternò, F. (1999). *Model-Based Design and Evaluation of Interactive Applications*. Springer-Verlag, London, UK, UK, 1st edição.
- Paternò, F. (2003). Concurtasktrees: An engineered approach to model-based design of interactive systems. *The Handbook of Task Analysis for Human-Computer Interaction*, pp. 483–503.
- Paternò, F.; Mori, G. & Galiberti, R. (2001). Ctte: An environment for analysis and development of task models of cooperative applications. Em *CHI '01 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '01, pp. 21–22, New York, NY, USA. ACM.
- Pereira Junior, M. & Prates, R. O. (2015). Um modelo para apoiar designers de sistemas colaborativos na antecipação de cenários. Em *XIV Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais*, IHC 2015.
- Pereira Junior, M.; Villela, M. L. B.; Rocha, E. C. F. & Prates, R. O. (2012). Avaliação da privacidade do facebook. Relatório técnico, Universidade Federal de Minas Gerais, Departamento de Ciência da Computação.
- Pereira Junior, M.; Villela, M. L. B.; Rocha, E. C. F. & Prates, R. O. (2013a). O navegar impreciso: A privacidade no facebook vivenciada por seus usuários. Em *IX Simpósio Brasileiro de Sistemas de Informação*, SBSI 2013, pp. 121–132.
- Pereira Junior, M.; Xavier, S. & Prates, R. O. (2013b). Antecipando possíveis implicações de privacidade na postagem de fotos no facebook. Em *XII Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais*, IHC 2013.
- Pereira Junior, M.; Xavier, S. I. d. R. & Prates, R. O. (2014). Investigating the use of a simulator to support users in anticipating impact of privacy settings in facebook. Em *Proceedings of the 18th International Conference on Supporting Group Work*, GROUP '14, pp. 63–72, New York, NY, USA. ACM.
- Prates, R. O. (1998). *A Engenharia Semiótica de Linguagens de Interfaces Multi-Usuário*. Tese de doutorado, Departamento de Informática, PUC-Rio.

- Prates, R. O. & Barbosa, S. D. J. (2007). *Introdução à teoria e prática da interação humano computador fundamentada na engenharia semiótica*. Jornada de Atualização em Informática. São Paulo, Sociedade Brasileira de Computação.
- Prates, R. O.; de Souza, C. S. & Barbosa, S. D. J. (2000). Methods and tools: a method for evaluating the communicability of user interfaces. *interactions*, 7(1):31--38.
- Prates, R. O.; de Souza, C. S. & Garcia, A. C. B. (1997). A semiotic framework for multi-user interfaces. *SIGCHI Bull.*, 29(2):28--39.
- Prates, R. O.; Rosson, M. B. & de Souza, C. S. (2015). Interaction anticipation: Communicating impacts of groupware configuration settings to users. Em *Fifth International Symposium on End-User Development*, Madrid.
- Pressman, R. & Maxim, B. (2014). *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education.
- Rauber, G.; Almeida, V. & Kumaraguru, P. (2011). Privacy albeit late. *Accepted at WEBMEDIA, Brazilian Symposium on Multimedia and the Web*.
- Reynolds, B.; Venkatanathan, J.; Gonçalves, J. & Kostakos, V. (2011). Sharing ephemeral information in online social networks: privacy perceptions and behaviours. Em *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part III, INTERACT'11*, pp. 204--215, Berlin, Heidelberg. Springer-Verlag.
- Schön, D. (1983). *The Reflective Practitioner: How Professionals Think in Action*. Harper torchbooks ; TB5126. New York: Basic Books.
- Smith, M. L. & Erwin, J. (2005). Role & Responsibility Charting (RACI). http://www.pmforum.org/library/tips/pdf_files/RACI_R_Web3_1.pdf.
- Veer, G. C. V. D.; Lenting, B. F. & Bergevoet, B. A. J. (1996). Gta: Groupware task analysis - modeling complexity. *Acta Psychologica*, 91:297--322.
- Welie, M. V.; Veer, G. C. V. D. & Eliëns, A. (1998). Euterpe - tool support for analyzing cooperative. Em *In the Proceedings of the Ninth European Conference on Cognitive Ergonomics*, pp. 25--30.
- Ziviani, N. (2004). *Projeto de algoritmos: com implementações em Pascal e C*. Thomson.

Apêndice A

Termo de Consentimento Livre e Esclarecido

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Eu, Manoel Pereira Junior, orientada pela Profa. Dra. Raquel Oliveira Prates, desenvolvi como parte da minha pesquisa de doutorado um Modelo para Apoiar Designers de Sistemas Colaborativos na Antecipação de Cenários, que consiste em uma ferramenta epistêmica, fundamentada na Engenharia Semiótica, que tem como objetivo apoiar o designer na modelagem de um sistema colaborativo de interesse. Esperamos que este modelo possa auxiliar o designer na antecipação de cenários potencialmente problemáticos ainda na fase design. Isso sem que ele tenha que modelar todos os cenários possíveis e esperados no sistema

Diante disso, você está sendo convidado pelo Núcleo de Pesquisa em Engenharia Semiótica e Interação (PENSI), o grupo de pesquisa em IHC do Departamento de Ciência da Computação da UFMG, para participar do presente estudo, que tem como objetivo avaliar o seu entendimento, como designer, da linguagem de modelagem que estamos propondo. Sua avaliação da nossa linguagem nos permitirá avaliar a sua expressividade, ou seja, o quanto a linguagem que estamos propondo se adequa ao desenvolvimento de um sistema colaborativo. Além disso, esperamos ainda avaliar a sua percepção sobre os cenários gerados pelo Cenógrafo. Sua avaliação nos fornecerá indícios sobre a utilidade do modelo como um todo.

Neste contexto, gostaria de solicitar que você manifeste o seu consentimento para participar deste estudo, realizando as seguintes atividades:

- Responder um questionário com dados gerais, dados sobre a sua experiência em design de IHC, sua experiência no desenvolvimento de sistemas colaborativos e seu conhecimento sobre Engenharia Semiótica e suas ferramentas epistêmicas;
- Ouvir uma explicação sobre a proposta geral do Modelo, sua linguagem, o Cenógrafo e exemplos;
- Realizar tarefas de modelagem usando o Cenógrafo;
- Responder um questionário sobre a sua impressão e sentimentos em relação ao modelo e ao Cenógrafo e participar de uma entrevista após a realização do teste.

O estudo completo terá duração aproximada de três a quatro horas, incluindo todas as atividades descritas acima. É importante você saber que:

1. Os dados coletados durante o estudo serão utilizados **estritamente** no contexto acadêmico e de pesquisa.
2. A equipe envolvida neste estudo tem o compromisso de publicar os resultados de suas pesquisas em fóruns acadêmicos. Entretanto, a publicação é baseada em nosso respeito à **privacidade** e **anonimato** dos participantes. Assim, a sua identidade e a sua participação nesta pesquisa serão mantidas em sigilo e os dados divulgados pela pesquisa não conterão nomes ou quaisquer outras informações que permitam identificá-lo(a).
3. O consentimento para participar deste estudo é uma **escolha livre** de sua parte, realizada a partir do esclarecimento de todas as suas dúvidas e questões sobre a pesquisa.
4. Você **pode interromper a sua participação** neste estudo a qualquer momento, sem sofrer nenhuma penalidade. Neste caso, todos os seus dados e resultados parciais serão descartados.
5. Eu, Manoel Pereira Junior, responsável pela condução do presente estudo, estou **disponível** para contato pelo telefone (37) 98806-1504 e pelo e-mail *manoel@dcc.ufmg.br*.

De posse das informações acima apresentadas, gostaria que você se pronunciasse sobre a sua decisão:

Dou o meu consentimento para participar do presente estudo.

Não dou o meu consentimento para participar do presente estudo.

Formiga, ____ de dezembro de 2015.

Nome do participante: _____

Assinatura do participante: _____

Nome do pesquisador: Manoel Pereira Junior

Assinatura do pesquisador: _____

Apêndice B

Questionário Pré-teste

QUESTIONÁRIO PRÉ-TESTE

- (1) Por favor, preencha o questionário abaixo. Suas repostas irão nos ajudar a analisar as informações que serão coletadas durante o estudo.
(2) Se necessário, utilize o verso das páginas para completar suas respostas.

Data: _____

Identificador _____

1. **DADOS PESSOAIS:**

Nome: _____ Sexo (M ou F): ____ Idade: _____

Formação: _____ Profissão: _____

Email: _____ Celular: _____

2. **EXPERIÊNCIA NO DESENVOLVIMENTO DE SISTEMAS COLABORATIVOS** (Por favor, preencha os campos com a opção que melhor representa a sua resposta a cada pergunta)

2.1. Você participa/já participou da atividade de design de algum sistema colaborativo?

a) () Sim. Qual(is)? _____

b) () Não

2.2. Caso sua resposta à questão anterior tenha sido sim, houve algum momento em que, durante o design, você teve dúvidas sobre as consequências que uma ação de um usuário poderia causar no sistema como um todo?

a) () Sim

b) () Não

2.3. Caso sua resposta à questão anterior tenha sido sim, você conseguiu sanar sua dúvida?

a) () Sim. Como? _____

b) () Não

2.4. Enquanto designer de um sistema colaborativo, você considera importante ter o conhecimento de todas as consequências que cada ação de cada usuário trará ao sistema como um todo?

a) () Sim. Como acha que isso seria possível? _____

b) () Não

2.5. Caso sua resposta à questão anterior tenha sido sim, você considera difícil visualizar/compreender/entender todas as consequências possíveis das ações de cada usuário?

a) () Sim

b) () Não

2.6. Caso sua resposta à questão anterior tenha sido sim, você conhece algum modelo/linguagem que ajude o designer a antecipar as consequências das ações de cada usuário?

a) () Sim. Qual? _____

b) () Não

3. **EXPERIÊNCIA NO USO DE SISTEMAS COLABORATIVOS** (Por favor, preencha os campos com a opção que melhor representa a sua resposta a cada pergunta)

3.1. Você já usou algum sistema colaborativo (por exemplo, uma rede social, um sistema acadêmico, um sistema de aprendizagem à distância, etc)?

a) () Sim. Qual(is)? _____

b) () Não

3.2. Caso sua resposta à questão anterior tenha sido sim, houve algum momento em que você teve dúvidas sobre as consequências que uma ação sua dentro do sistema?

a) () Sim. Qual(is)? _____

b) () Não

3.3. Caso sua resposta à questão anterior tenha sido sim, o sistema lhe ofereceu alguma forma de verificar as consequências de suas ações?

a) () Sim. Como? _____

b) () Não

4. **EXPERIÊNCIA DOCENTE**

4.1. Você já lecionou alguma disciplina de desenvolvimento de software, IHC, Programação e/ou Engenharia de Software?

a) () Sim.

Quais: _____

b) () Não

5. **CONHECIMENTO SOBRE ENGENHARIA SEMIÓTICA E SUAS FERRAMENTAS EPISTÊMICAS**

5.1. Você possui conhecimento em Engenharia Semiótica?

1. () Sim

2. () Não

Apêndice C

Apresentação do SIGMa para os projetistas



Um Modelo para Apoiar Designers de Sistemas Colaborativos na Antecipação de Cenários

Manoel Pereira Junior
Raquel Oliveira Prates

UFMG

DCC

PENSI



Winweb



CAPES



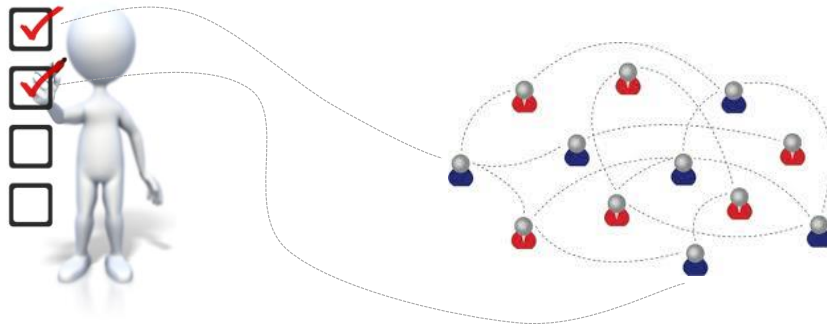
FAPEMIG

Contexto

- Sistemas colaborativos atuais
 - flexibilidade de customizar diferentes aspectos
- Designers devem planejar estas customizações em tempo de design
 - decidir quais escolhas e ações estarão disponíveis aos usuários;
 - considerar os cenários que serão gerados por elas;
 - e o seu impacto ao longo do tempo.

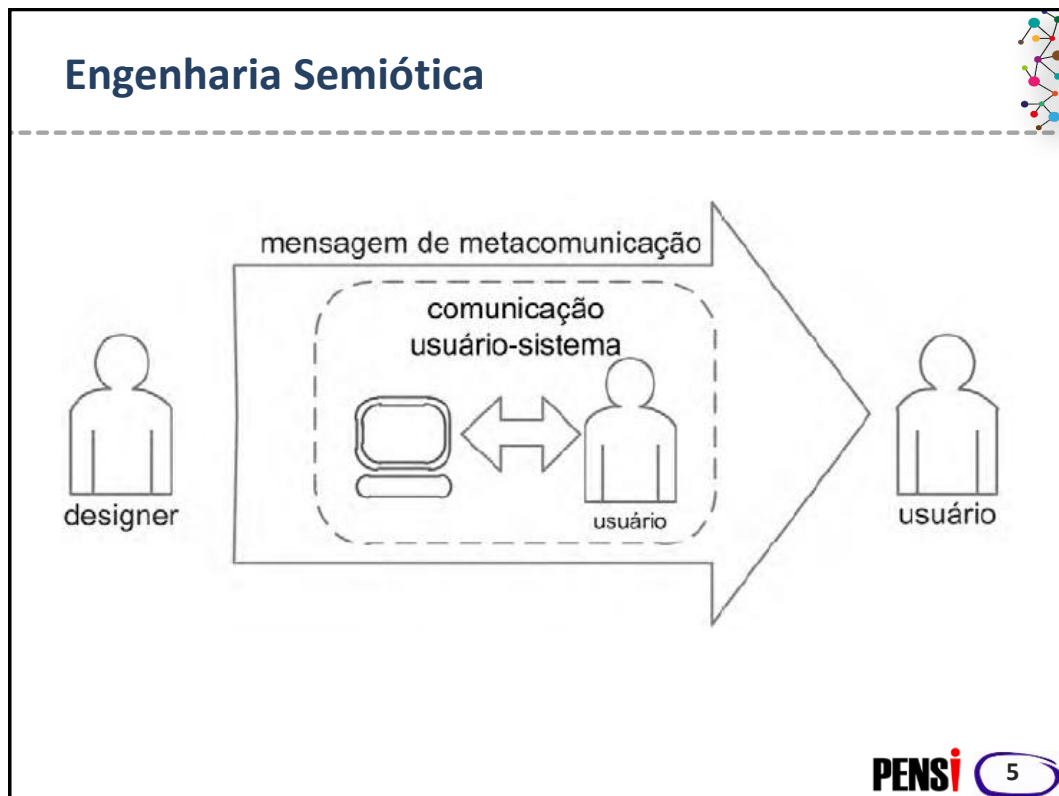
Objetivo

- Apoiar designers de sistemas colaborativos
 - na antecipação de cenários futuros
 - antes de qualquer implementação do sistema
 - para decidir se cenários são desejáveis ou não



Engenharia Semiótica

- A engenharia semiótica caracteriza aplicações computacionais como **artefatos de metacomunicação**, ou seja, artefatos que comunicam uma mensagem do designer para os usuários sobre a comunicação usuário–sistema, sobre como eles podem e devem utilizar o sistema, por que e com que efeitos.



Engenharia Semiótica

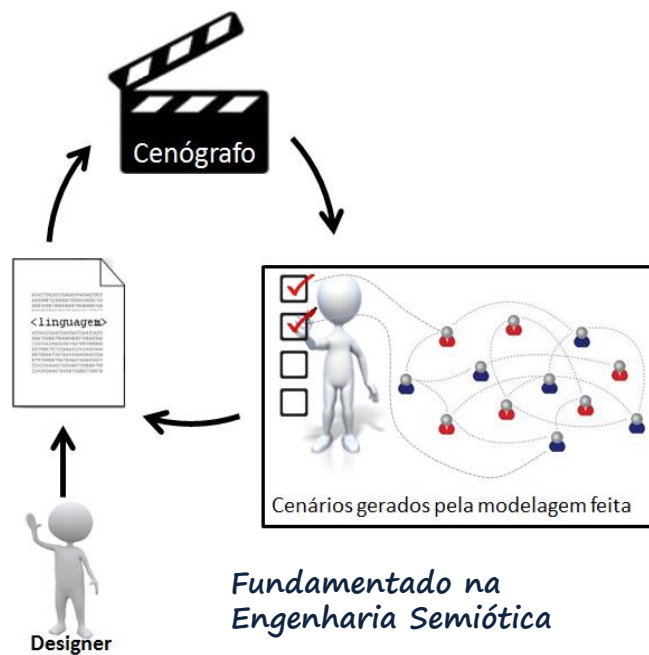
- **Comunicabilidade** é um conceito de qualidade dos sistemas computacionais interativos que comunicam de forma eficiente e efetiva aos usuários as intenções comunicativas do designer, a lógica e os princípios de interação subjacentes.
- A comunicabilidade pode ser definida tecnicamente como a “capacidade do preposto do designer de alcançar a metacomunicação completa, comunicando aos usuários a essência da mensagem original do designer” (de Souza, 2005a, p. 114), permitindo, portanto, que os usuários gerem significados compatíveis com aqueles codificados pelo designer.

PENSI 6


Engenharia Semiótica









- **Ferramenta Epistêmica:** não gera diretamente uma resposta ou solução para o problema. Em vez disso, apoia o designer na exploração do espaço e da natureza do problema, bem como das restrições sobre soluções candidatas

Proposta



Linguagem de design proposta




Papéis Grupos Artefatos Ações Posses Mudanças Relações









Espaços de tempo: são contextos do sistema onde cada papel ou grupo possui o mesmo conjunto de ações e os mesmos artefatos

Espaços de tempo

PENSI 9

Linguagem de design proposta




Espaços de tempo Grupos Artefatos Ações Posses Mudanças Relações


Papéis: representam os papéis previstos pelo designer que farão parte do sistema

Papéis


PENSI 10

Linguagem de design proposta







Espaços de tempo




Papéis




Grupos




Artefatos




Ações



Posses



Mudanças




Relações


Grupos: são formados por papéis previamente definidos. Podem ainda envolver todos os indivíduos de um papel ou apenas um subgrupo deles

Grupos


PENSI 11

Linguagem de design proposta







Espaços de tempo




Papéis




Grupos




Artefatos




Ações



Posses



Mudanças




Relações


Artefatos: são os elementos que poderão ser manipulados dentro do sistema que está sendo modelado. Podem ser atômicos ou compostos

Artefatos


PENSI 12

Linguagem de design proposta







Espaços de tempo




Papéis




Grupos




Artefatos



Posses



Mudanças




Relações


Ações: são as possibilidades de interação pensadas pelo designer e disponíveis à papéis, grupos ou sistema

Ações


PENSI 13

Linguagem de design proposta







Espaços de tempo




Papéis




Grupos




Artefatos



Ações



Mudanças




Relações


Posse: define qual papel ou grupo detêm a posse de um determinado artefato

Posses


PENSI 14

Linguagem de design proposta







Espaços de tempo




Papéis




Grupos




Artefatos



Ações



Posses




Relações

Mudanças dizem respeito ao que pode mudar no sistema, quando uma ação for executada. Podem haver mudanças no conjunto de ações, de posse, de tempo e de papel/grupo

PENSI 15

Mudanças...



- As mudanças não estão atreladas à uma ação, pois:
 - Uma ação pode causar várias mudanças diferentes no sistema
 - Exemplo: A ação de “Iniciar um diário” dá acesso ao professor (mudança 1) e aos alunos matriculados (mudança 2)
 - Uma mudança pode ser causada por várias ações diferentes
 - Exemplo: O aluno pode se tornar “desistente” (mudança) pela quantidade de faltas em uma disciplina ou por uma ação arbitrária da secretaria.

PENSI 16

Linguagem de design proposta



Espaços de tempo



Papéis



Grupos



Artefatos



Ações



Posses



Mudanças



Relações

Relações: são o elo entre uma ação e as mudanças que ela gera. As relações são as regras que estabelecem quais cenários podem ser criados no sistema

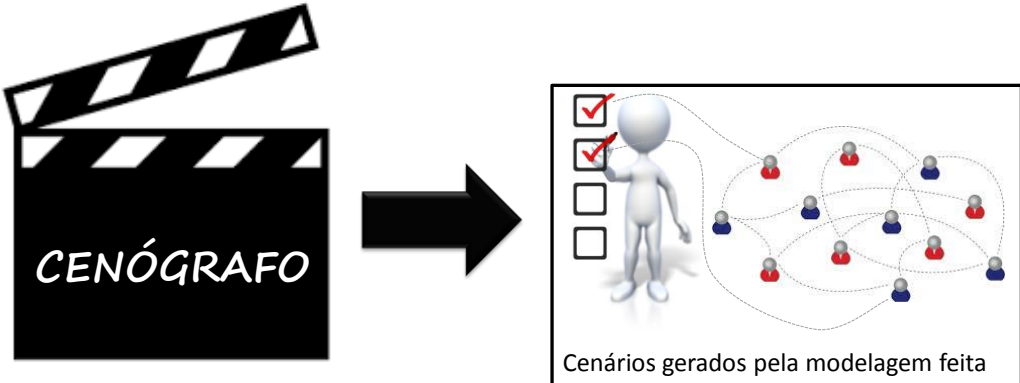
PENSI 17

Modelo gerado



PENSI 18

Cenógrafo



CENÓGRAFO

Cenários gerados pela modelagem feita

PENSI 19

Cenógrafo

- Exemplo de modelagem: sistema de gerência de conferências

PENSI 20

Apêndice D

Material do teste de inspeção de
um modelo pronto

Apêndice E

Material do teste de modelagem de um sistema

Nome do participante: _____ Identificação: _____

Agora que você foi apresentado aos conceitos básicos do SIGMa e já usou o protótipo explorando um modelo pronto, você irá utilizá-lo para modelar o sistema colaborativo descrito abaixo. Para guiá-lo nesta tarefa, considere o seguinte cenário:

CENÁRIO

Uma área de pesquisa que vem crescendo nos últimos anos é a que trata do legado digital. Pesquisadores desta área basicamente buscam respostas para a seguinte pergunta: “O que acontece com meus dados quando eu me tornar inativo?”. Tornar-se inativo pode significar diferentes coisas, desde simplesmente deixar de usar sistemas computacionais até a morte do usuário. Você faz parte de uma equipe que pretende desenvolver um sistema de gerência de legado digital, que ajudará o usuário a lidar com esta situação. A especificação do sistema é a seguinte:

Enquanto está ativo, um usuário do sistema pode criar, ver, editar e excluir documentos próprios livremente.

1. A ação de **criar** um documento permite ao usuário:

- Ver o documento
- Editar o documento
- Excluir o documento

2. A ação de **ver** um documento permite ao usuário:

- Ver o documento
- Editar o documento
- Excluir o documento

3. A ação de **editar** um documento permite ao usuário:

- Ver o documento
- Editar o documento
- Excluir o documento

4. A ação de **excluir** um documento não permite que o usuário faça mais nada sobre ele

Além das ações acima, o sistema permite ainda que o usuário configure o que acontece com seus documentos online quando ele se tornar inativo. Cada usuário pode configurar o seu próprio critério de inatividade. Por exemplo, para um usuário “ficar inativo” significa não acessar o sistema no período de 1 mês, enquanto que para outro usuário “ficar inativo” significa não acessar o sistema por 10 meses. Esta é, portanto, uma configuração independente de cada usuário. Após efetuar esta configuração, o sistema então passa a ser responsável por identificar

a inatividade de cada usuário. Além de configurar o tempo que caracteriza como “inatividade”, o usuário deve informar ainda quem receberá (outros usuários) o acesso aos seus documentos e o que poderá fazer com eles. As configurações possíveis são:

- **Apenas ler:** *permite ao usuário que recebeu o documento apenas ler o documento, não podendo excluí-lo ou editá-lo. A ação de **ler** o documento é recursiva, ou seja, o usuário que recebeu o documento pode executá-la quantas vezes quiser.*
- **Ler e editar:** *permite ao usuário que recebeu o documento ler e editar o documento, não podendo excluí-lo. A ação de **ler** o documento tem como consequências a ação de editar e a própria ação de ler. O mesmo acontece com a ação **editar**.*
- **Ler, editar e excluir:** *permite ao usuário que recebeu o documento ler, editar e excluir o documento. A ação de **ler** o documento tem como consequências a ação de editar e a própria ação de ler. O mesmo acontece com a ação **editar**. Já a ação de excluir faz com que o usuário que recebeu o documento não possa fazer mais nenhuma ação sobre ele.*

Com o cenário acima em mente, você deverá realizar as tarefas descritas nas páginas seguintes. Vale ressaltar que a tarefa de modelagem de sistemas é intrinsecamente intelectual. Por isso não há um tempo definido para cada uma das tarefas. Como o próprio modelo preconiza, você poderá alterar qualquer ponto da sua modelagem a qualquer tempo, sempre que achar necessário.

Observação 1: o objetivo da avaliação não é avaliar o protótipo em si, mas sim a linguagem e os cenários gerados. Assim, caso haja dúvidas em relação à operação do protótipo ou à sua interface, as mesmas poderão ser respondidas.

Observação2: Toda a modelagem será apoiada pelo nosso protótipo, que permite a definição visual de toda a modelagem do sistema colaborativo. O protótipo estará instalado no computador que você está usando.

Apêndice F

Questionário pós-teste

Apêndice G

Roteiro do grupo focal

TEMAS PARA O GRUPO FOCAL

BLOCO 1 – PERGUNTAS SOBRE AS DIMENSÕES DA LINGUAGEM

1. Houve alguma dificuldade em modelar o sistema em questão de acordo com as dimensões previstas? Sim/Não. Motivos?

BLOCO 2 – PERGUNTAS SOBRE O SIGN

2. Cenários gerados pelo SIGN. Identificaram problemas/cenários indesejados? Como agiram neste caso?
3. A inspeção dos cenários causou novas reflexões em relação ao modelo?
4. Você considera o modelo proposto como tendo potencial para apoiar o designer de sistemas colaborativos reais? Sim/Não. Motivos?
5. Quais são os pontos positivos do modelo proposto? E negativos? Alguma colocação sobre o protótipo?
6. Alguma outra consideração que julguem importante?