

3

Um Modelo para Discussões sobre Evolução de Sistemas

Neste capítulo apresentamos o modelo conceitual proposto nesta pesquisa para estruturar a comunicação de grupo em discussões sobre evolução de sistemas de uso comum. Para contextualizar adequadamente nossa contribuição, começamos mostrando um cenário em que este modelo pode ser instanciado para em seguida apresentar a estrutura e descrever todos os elementos do modelo. Ao final da descrição, mostramos um diagrama das classes do modelo e apresentamos um fluxo de etapas do modelo em uso para consolidar a nossa proposta de forma concisa. Por fim, mostramos como o modelo atende às propriedades pretendidas de representação do discurso sobre sistemas de grupo, consideradas na literatura e apresentadas no Capítulo 2.

3.1

Cenário de Uso do Modelo

Antes de apresentarmos o modelo proposto, ilustramos o contexto em que ele deve ser usado, usando para isto um cenário de exemplo.

Podemos imaginar o seguinte cenário: *João é coordenador de pós-graduação de uma universidade e utiliza um sistema de controle de requerimentos (SR) para gerenciar solicitações acadêmicas dos alunos de pós-graduação, tais como: pedido de aproveitamento de créditos, prorrogação de prazo de defesa de tese, atestado de conclusão de créditos, etc. Os requerimentos passam pela avaliação de João e depois seguem para ser avaliados por outros coordenadores (Central e Setorial). João, participa de uma reunião juntamente com outros coordenadores e com os designers do SR, para discutir sobre melhorias no Sistema de Requerimentos. João, no papel de coordenador de pós-graduação, sugere a seguinte modificação: “ O sistema deve enviar email aos coordenadores que avaliaram o requerimento para avisar da sua conclusão”. Marcelo, que é Coordenador Setorial (avalia todos os processos do seu centro acadêmico), discorda de João dizendo: “No meu caso, isso vai encher minha caixa de email, pois avalio muitos processos.” Paulo, no papel de Coordenador Central, discorda: “Sou eu quem dou o parecer final, portanto não preciso receber email”. Lurdes, no papel de Secretária de*

pós-graduação, acrescenta: “Este problema não se aplica para mim porque acompanho todos os requerimentos e por isso, vejo quando eles terminam.” Diante deste impasse, Mário, que é designer do sistema, propõe uma discussão um pouco mais detalhada.

É neste cenário que nosso modelo pode ser usado. A Figura 3.1 mostra esquematicamente como um grupo de usuários, com diferentes papéis, usa um ambiente tecnológico para discutir sobre mudanças em um sistema que todos utilizam. Podemos ver quatro grupos de pessoas, cada um representando um papel do sistema em discussão (Papel A, Papel B, Papel C e Papel D). Portanto, as trocas de mensagens do modelo aqui proposto estão sempre considerando os papéis do sistema, o que é, como dissemos na introdução, um requisito que estipulamos para organizar as discussões sobre evolução de sistema de grupo. Recordando a motivação deste requisito, ele se deve ao fato de que em sistemas de grupo com papéis definidos para os usuários no protocolo tecnológico o impacto das modificações definidas no escopo de um único papel tem necessariamente de ser avaliado no contexto dos demais papéis relacionados a ele. Em sistemas onde tais papéis não estão definidos na tecnologia este impacto pode existir (e muito provavelmente existe), mas para percebê-lo e discuti-lo objetivamente o grupo tem de - por vontade própria - criar signos no protocolo social que identifiquem e referenciem sistematicamente papéis, relações entre eles, e impactos da modificação de uns sobre os outros. Ou seja, o grupo tem de criar um modelo (no seu discurso). Para esta pesquisa estamos trabalhando em contextos onde este modelo foi produzido pelos projetistas e desenvolvedores da tecnologia.

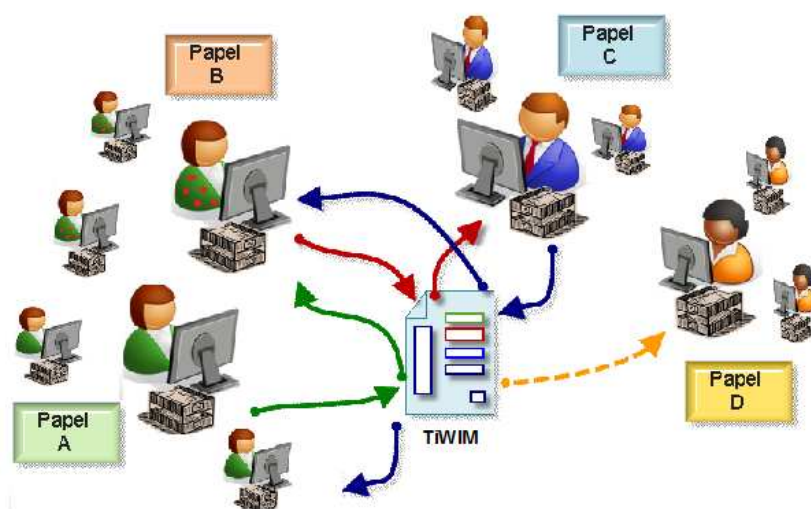


Figura 3.1: Discussão sobre Sistema de Grupo

No centro da imagem pode ser visto o objeto que representa a ideia de modificação, denominado TiWIM - *This is What I Mean*. Ele é descrito através

de uma linguagem específica de nosso modelo (apresentada posteriormente). A representação deste objeto é bem próxima do que seria o sistema com a modificação implementada.

Adicionalmente, uma pessoa pode falar no papel de outra. Por exemplo, uma pessoa que é “designer” pode falar como um “cliente” (considerando “designer” e “cliente” papéis do sistema), ou seja, ela pode falar por um “cliente” mesmo não sendo um. Esta possibilidade permite que usuários expressem o que eles imaginam que seja a função, o contexto, as prioridades, de outros membros do grupo que desempenham papéis diferentes do dele, usuário. Outra possibilidade é alguém dizer que a proposta de modificação só é válida para o Papel A, por exemplo.

Assim, o cenário de uso do modelo que estamos propondo permite a troca de ideias sobre melhorias no sistema, com a possibilidade de expressão menos abstrata de comunicar as ideias de modificação. Além disso, o discurso sobre a modificação pode levar em conta os papéis envolvidos. A seguir, descrevemos o modelo conceitual.

3.2

Descrição Geral do Modelo Conceitual para Descrever e Comunicar Mudanças em Sistemas de Grupo na Web

Apresentamos um modelo conceitual fundamentado na Engenharia Semiótica, ilustrado na Figura 3.2, para descrever e comunicar mudanças em sistemas de grupo na Web. É importante dizer que a proposta de um “modelo conceitual” é justificada pelo fato de ele não depender das tecnologias que utilizamos para implementá-lo. Portanto, tudo que apresentamos neste capítulo independe das tecnologias que adotamos para implementar parte do modelo. Uma concretização possível é apresentada no capítulo seguinte.

O modelo aqui proposto foi elaborado partindo do modelo que caracteriza os processos comunicativos relacionados à extensão colaborativa de sistemas, proposto por Cunha (Cunha, 2001) e apresentado no Capítulo 2, que por sua vez foi fundamentado no modelo de comunicação de Jakobson (Jakobson, 1960). Assim como em (Cunha, 2001), nosso modelo também possui os elementos do modelo de comunicação de Jakobson: interlocutores (emissores e receptores), canal, mensagem, código e contexto. Adicionalmente, nosso modelo herda características de Cunha:

Contexto : nosso modelo considera o domínio da aplicação, o sistema operacional, o ambiente de extensão (que no caso é o ambiente de modificação) e a experiência do usuário.

Interlocutores : consideramos emissores e receptores na função de designer, de usuário como estendedor e como co-estendedor (que no nosso modelo denominamos de Usuários Ativos) e na função de usuários (que no nosso modelo denominamos de Usuários Passivos). Basicamente só mudamos a nomenclatura mas as funções são as mesmas. Estas funções não devem ser confundidas com os “papéis” dos usuários no domínio da aplicação. Elas são definidas, como se vê, em relação a como o interlocutor atua no âmbito da “modificação” do sistema. Mais adiante elas serão definidas mais extensivamente.

Canal : corresponde ao hardware e software onde o sistema está executando.

Quanto aos **códigos** e **mensagem**, nossa proposta agrega novos conceitos e diferencia-se da proposta de Cunha. Propomos dois novos códigos que se adicionam à Linguagem de Interface já proposta por Cunha: a Linguagem de Modificação (LM) e a Linguagem de Comunicação (LC).

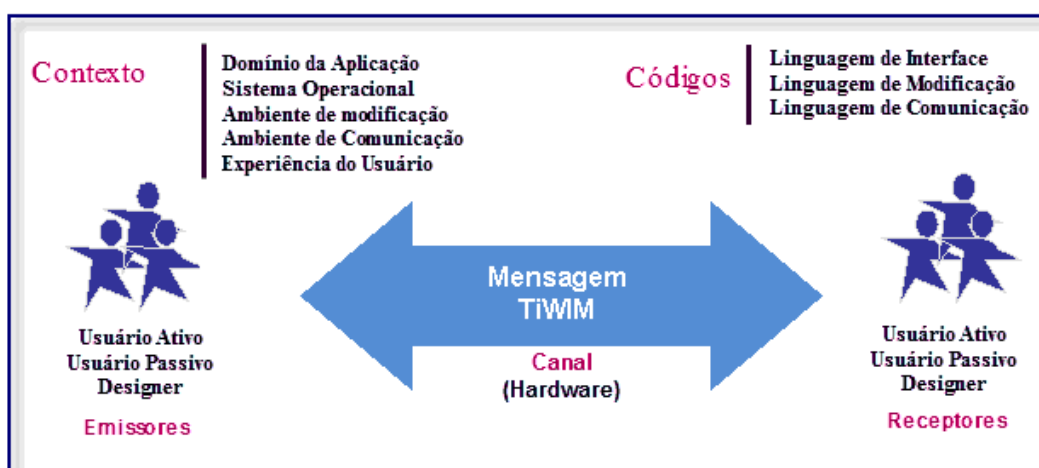


Figura 3.2: Modelo para Descrever e Discutir Evolução de Sistemas.

Interlocutores (Emissores e Receptores)

Interlocutores são os emissores e receptores das mensagens trocadas na discussão. Eles são usuários do sistema discutido ou pessoas que fazem parte da equipe de desenvolvimento do sistema em questão, incluindo programadores, projetistas, arquitetos, etc. A cada interlocutor é associada uma função. Se a função do interlocutor corresponder a de um “usuário ativo” (estendedor ou co-estendedor) ou “usuário passivo” (usuário típico), então o interlocutor terá um papel de domínio do sistema em discussão. Se o interlocutor faz parte da equipe de desenvolvimento, então ele terá o papel de “designer” (que corresponde à sua função no processo de evolução). As funções que podem ser desempenhadas pelos participantes de uma discussão sobre evolução de sistema são, portanto:

Designer do Sistema : atuam como pessoas que podem propor soluções, incrementar as propostas dos usuários com requisitos e recomendações técnicas, como por exemplo, incrementar a modificação com tratamento de erros, ou simplesmente captar as sugestões propostas pelos usuários;

Usuários Ativos : são usuários do sistema que contribuem com novas ideias de modificação (denominado estendedores no modelo de Cunha) ou que incrementam a ideia de outro usuário (denominado co-estendedor no modelo de cunha);

Usuários Passivos : são usuários do sistema que só acompanham a discussão e não opinam. Estes interlocutores basicamente só recebem mensagens; Eventualmente eles podem emitir uma mensagem, mas sem alterar substancialmente o objeto de discussão do grupo.

Como já ressaltado, as funções Usuários Ativos, Usuários Passivos ou Designers são desempenhadas na discussão. Os papéis do sistema representam as funções que os usuários exercem no sistema, como por exemplo, em um sistema de grupo no contexto de administração universitária, temos o papel de secretária, coordenador de curso, professor, aluno, etc.

Canal de Comunicação

O canal de comunicação é um conjugado de hardware e software que garante o processamento de representações computacionais que se concretizam na superfície do sistema como signos de interface e interação. Algumas propriedades do meio computacional podem impactar a comunicação. São elas: “a possibilidade de utilização de múltiplos códigos para a expressão de mensagens, a possibilidade de construção de mensagens interativas e a possibilidade de comunicação síncronas e assíncronas.” (Cunha, 2001)

Contexto

O contexto da discussão envolve o domínio da aplicação discutida, o ambiente de extensão, o ambiente de comunicação, o ambiente do sistema operacional e a experiência do usuário. No ambiente de comunicação, inclui-se um histórico de mensagens trocadas. No ambiente de extensão, inclui-se um registro das ações executadas com finalidade de armazenar o histórico das ações de projeto (*design rationale*) da modificação. O domínio da aplicação deve conter papéis pré-definidos, os quais serão mencionados nas mensagens.

No contexto de discussões sobre “evolução” de sistemas é interessante que a discussão seja assíncrona, uma vez que os participantes precisam de tempo

de reflexão. Eles precisam rever e revisar as mensagens trocadas. Portanto, as comunicações do modelo aqui proposto são assíncronas.

Mensagem

A mensagem, ilustrada na Figura 3.3, é o componente do modelo que armazena o conteúdo comunicado. Este conteúdo pode ser dividido em dois grupos: conteúdo sobre **comunicação** da mensagem (estrutura de comunicação) e conteúdo sobre o **detalhe da modificação** no sistema. O primeiro está relacionado com a organização da conversa e com a comunicação da ideia geral da mensagem. O segundo, diz respeito ao detalhamento, em uma representação menos abstrata, do que foi dito no primeiro.

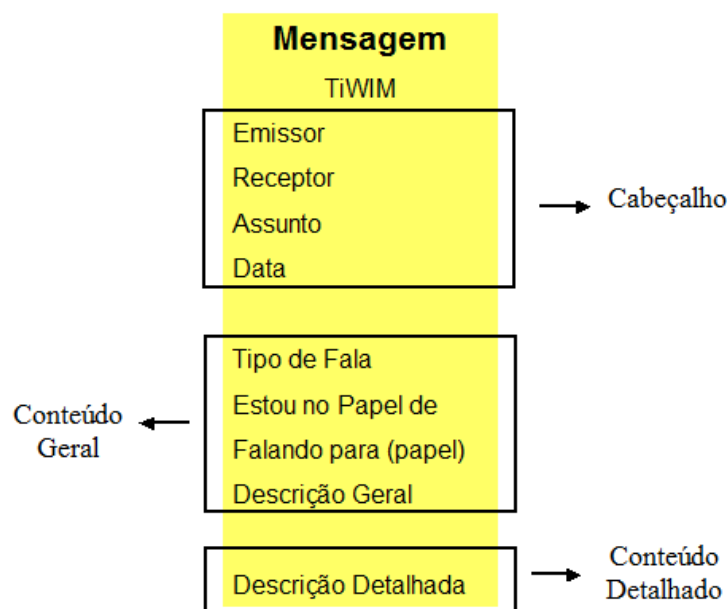


Figura 3.3: Estrutura da Mensagem do Modelo

Na Figura 3.4, apresentamos uma referência às partes do *template* de metamensagem da Engenharia Semiótica associadas a elementos da estrutura da mensagem. Queremos mostrar com isso que nosso modelo tem por objetivo contemplar todos os elementos da mensagem de metacomunicação. Na Seção 4.3 discutiremos mais sobre isto.

Os elementos da mensagem que dizem respeito ao contexto de interlocução (ou processo de comunicação que realiza a discussão) devem considerar os papéis do domínio do sistema em questão. Diante disto, o emissor precisa deixar claro: na posição de quem ele está falando, como se dissesse que está “falando no papel de quem”; Deve também definir para qual papel ele está falando, como se dissesse “estou falando para o papel [nome do papel]”.

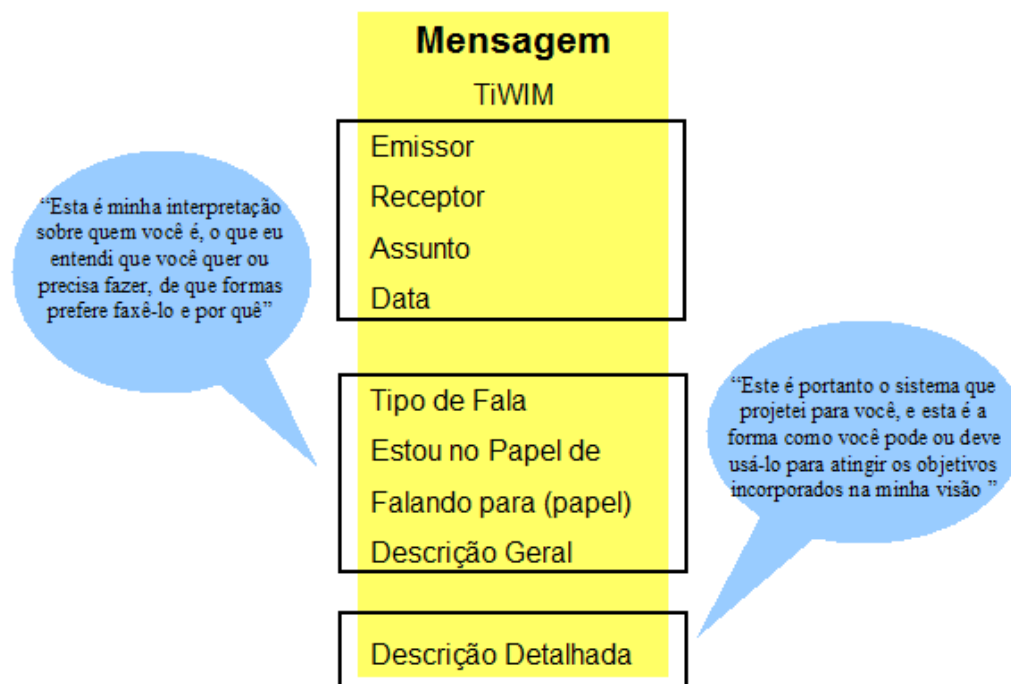


Figura 3.4: Relações entre partes do *template* de metacomunicação da Engenharia Semiótica e partes da estrutura da mensagem no modelo proposto.

Por exemplo, em uma discussão sobre um sistema de controle de processos acadêmicos, um interlocutor deve dizer se ele está falando no papel de “aluno” ou de “professor” e dirigindo-se a interlocutores no papel de “coordenador”. Estas informações são extremamente importantes para discussões em sistemas de grupo com papéis definidos no protocolo tecnológico, pois através destas falas o emissor embute na mensagem a relação entre a proposta de modificação e o papel de domínio impactado com tal mudança.

O emissor deve marcar explicitamente o propósito de sua mensagem. Para isso ele deve selecionar um dos “tipos de fala”. Estes tipos são marcados por itens lexicais representativos da intenção do falante ou de sua atitude proposicional diante do que está dizendo. São eles: “pergunto”, “sugiro”, “além disso”, “comento”, “acrescento”, “informo”, “respondo”, “concordo”, “discordo”, “para que”, “porque”, “peço”, “explico”, “exemplifico”, “não entendi”, “e”, “se”, “então”, “ou”, “mas”, “concordo mas”, “discordo mas”, “continuo”, “transcrevo”. Estes tipos de fala foram extraídos da ferramenta de discussão Orion (de Souza et al., 2006b). Esta ferramenta usada ao longo de vários anos, sugere que tais marcadores para tipos de fala ajudam a estruturar a conversa, e conseqüentemente, facilitam a reconstituição do raciocínio da discussão. Esses tipos de falas funcionam como conectores de mensagens.

Além destas informações, o interlocutor deve definir um tema para a

conversa e endereçar para quem está enviando a mensagem, podendo ser para um ou para vários membros do grupo.

A mensagem também deve conter a intenção geral do emissor, ou seja, a ideia geral da comunicação. Opcionalmente, a mensagem pode ainda conter a intenção detalhada do emissor, representada na própria interface. A seguir apresentamos todas as dimensões da estrutura da mensagem TiWIM, devidamente associadas à Linguagem de Comunicação (para conteúdos ligados ao contexto de interlocução) e à Linguagem de Modificação (para detalhamento da proposta do interlocutor).

A Linguagem de Comunicação

Esta linguagem permite a representação da estrutura de comunicação que corresponde ao “Cabeçalho” e “Conteúdo Geral” da mensagem, ilustrada na Figura 3.3. Portanto, a Linguagem de Comunicação (LC) permite aos interlocutores expressar os seguinte conteúdo:

Cabeçalho:

- Emissor: identificador da pessoa que está enviando a mensagem. Pode ser, por exemplo, um endereço de email ou o login do ambiente de comunicação.
- Receptor: identificador da pessoa que está recebendo a mensagem.
- Assunto: tema que está sendo discutido, livremente expresso em linguagem natural.
- Data: “selo de tempo” correspondente ao momento em que a mensagem está sendo enviada.

Conteúdo Geral:

- Tipo de Fala: expressão da intenção de comunicação do emissor definida através da seleção de um dos tipos pré-definidos no modelo, por exemplo: concordo, discordo, acrescento, proponho, etc.
- Estou no papel de: papel que o emissor está exercendo.
- Falando para (papel): papel que o emissor está se referindo.
- Descrição Geral: Informação livremente expressa em linguagem natural sobre a modificação. Pode conter referências diretas a elementos do sistema (ou parte deles) e também a intenção da modificação. Isto permitirá a construção do discurso e a manipulação do foco ao longo do mesmo.

O contexto de atuação direta dos interlocutores deste modelo é o ambiente de comunicação. Portanto, devem estar disponíveis no ambiente de comunicação funções para enviar e receber as mensagens, assim como recursos para registrar e disponibilizar o histórico das mesmas.

Neste modelo, estamos trabalhando com a possibilidade de termos dois ambientes: o ambiente de comunicação e o ambiente de modificação. O primeiro é onde as mensagens são elaboradas, enviadas e recebidas. O segundo é onde a modificação detalhada é construída. No primeiro, utiliza-se a linguagem de comunicação (de fato, um texto estruturado, com elementos textuais livres ou selecionados entre parâmetros pré-fixados) e no segundo a linguagem de modificação (cuja feição será apresentada adiante). Diante disso, é preciso ter um elemento que associe os objetos que foram descritos com a linguagem de modificação no ambiente de modificação com elementos do ambiente de comunicação. Portanto, o ambiente de comunicação deve ter um mecanismo para fazer referência à modificação detalhada de maneira que a modificação detalhada possa ser perceptível e compreensível pelo receptor da mensagem. Por exemplo, se o ambiente de comunicação for análogo a um sistema de correio eletrônico com mensagens estruturadas, o mecanismo de referência pode ser anexar à mensagem arquivos contendo a modificação detalhada representada na linguagem de modificação. Se o ambiente de comunicação for um fórum de discussão online, o mesmo mecanismo de anexação das representações pode cumprir o papel de fazer a ligação entre as duas partes da comunicação do falante.

A Linguagem de Modificação

A Linguagem de Modificação (LM) permite representar as propostas detalhadas de modificação que corresponde ao “Conteúdo Detalhado” da Figura 3.3.

Ela possibilita uma representação concreta da ideia de modificação do sistema, ou seja, ela é descrita de maneira que a proposta de modificação contém signos de primeiridade para representar a modificação implementada. Para isso ela combina três linguagens: (I) A linguagem de interface (UIL); (II) A linguagem natural; (III) Uma linguagem de *script* para gravar ações no sistema.

Seguimos o argumento de Cunha de que a linguagem ideal para representar a expansão do sistema é a UIL, visto que ela é compreensível para os agentes humanos e pelo computador. Porém, vimos que a UIL não cobre adequadamente todo o espectro semiótico necessário para comunicar as intenções de modificação de sistema. Priorizando os signos que evocam

a “primeiridade” dos conteúdos pretendidos pelos interlocutores (isto é, sua aparência e seu comportamento, que impactam diretamente o aparato sensorial do receptor no momento de interpretar a mensagem do emissor), a UIL torna difícil comunicar os por quês, para quês, como e outras dimensões adicionais que são criticamente importantes num processo de discussão de alternativas. Então, para aumentar o poder de expressão, acrescentamos a linguagem natural, a qual tem alto poder de expressão e baixo custo de aprendizado (considerando que os usuários a conhecem). Neste modelo, ela é usada para acrescentar informação sobre elementos da interface, através da função “ANOTAR”, e está sempre associada a um objeto da interface.

A linguagem de *script* é utilizada para representar processos ou ações do sistema que está sendo discutido. Concretamente, pode-se usar um “gravador de macros” para gravar os processos do sistema e depois associar os *scripts* gravados a elementos da interface. Para tanto, a LM disponibiliza uma função “ASSOCIAR”, a qual associa um *script* contendo as ações (interações com o sistema) a um elemento da interface.

Para alterar a UIL original, a Linguagem de Modificação permite, além da associação de *scripts* a elementos de interface, operações de duplicação, apagamento, renomeação e reposicionamento dos mesmos. Tais operações são comandadas por instruções comumente encontradas em editores textuais e gráficos: “COPIAR”, “COLAR”, “REMOVER”, “RENOMEAR”, e “MOVER”.

É oportuno comentar que o mecanismo de anotação da LM possibilita a expressão de signos metalinguísticos para a metacomunicação da interface (modificada) do sistema. O mecanismo de associação de *scripts* possibilita a expressão de signos dinâmicos e finalmente os mecanismos de duplicação, apagamento e reposicionamento de elementos da UIL possibilita a expressão de signos estáticos. A seguir, apresentamos um diagrama das classes pertencentes ao modelo de discussão.

3.3

Classes do Modelo

Apresentamos nesta seção as classes pertencentes ao modelo para discutir sobre modificações de sistemas de grupo, ilustrada na Figura 3.5.

Um **Grupo de Discussão** possui vários **Interlocutores** e **Mensagens**, além disso, está sempre associado a um sistema ou parte dele. Um sistema, por sua vez, contém **Papéis** protocolados tecnicamente (e não socialmente). As **Mensagens** trocadas pelo grupo, dizem o que eles pensam sobre a modificação do sistema. Seu cabeçalho é composto de pelo menos dois interlocutores (1

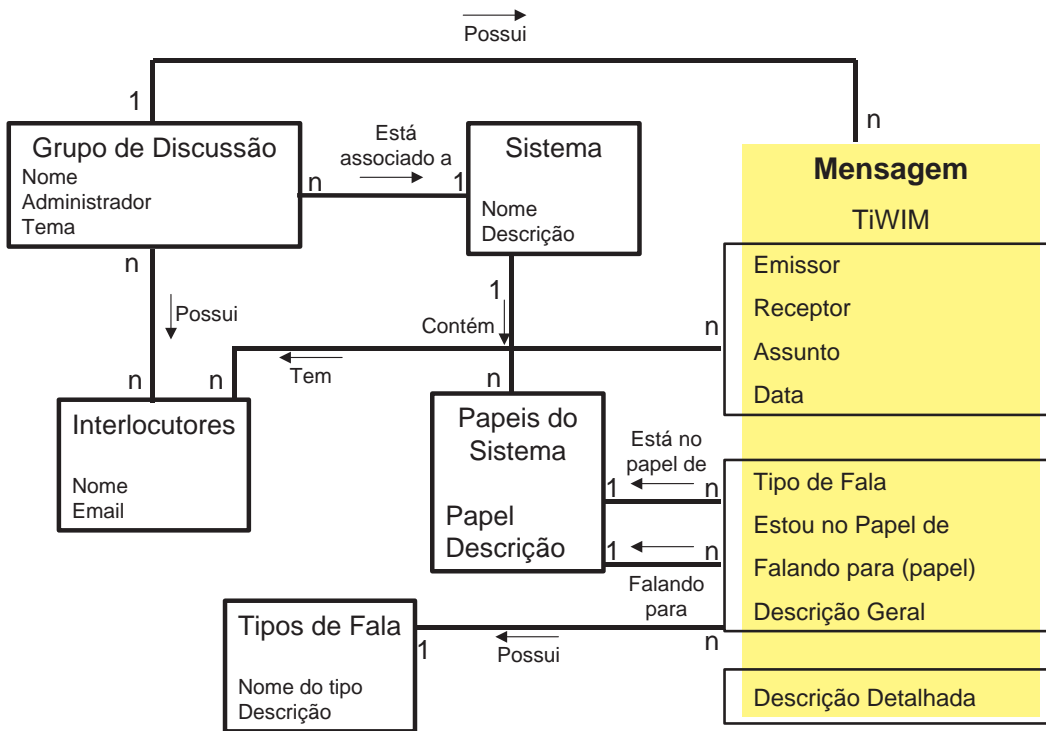


Figura 3.5: Classes do Modelo

emissor e 1 receptor), um assunto e data de envio. O corpo da mensagem pode ser dividido em dois blocos: (i) o primeiro contém o **Tipo de Fala**, o papel do sistema através do qual o emissor está falando, o papel do sistema para quem o emissor está falando (e.g., estou falando para o Papel A) e uma descrição geral do conteúdo da mensagem. (ii) O segundo bloco contém um detalhamento do que foi dito no primeiro (Descrição Detalhada), mas com um diferencial, ele é expresso através da própria interface do sistema. Portanto, ele é um arquivo contendo a porção modificada do sistema.

É importante dizer o papel de “designer” deve sempre existir na classe “Papeis do Sistema”, visto que, em nosso contexto, o designer é uma entidade que poderá sempre participar da discussão. A seguir, apresentamos o funcionamento do modelo.

3.4 Fluxo do Modelo

A explicação dos componentes do modelo conceitual mostrou quais são os elementos atuantes na comunicação sobre modificação de sistemas. Vamos agora descrever o comportamento destes elementos em atividades previstas para o contexto de uso deste modelo.

O modelo conceitual para discutir sobre alterações em sistemas de grupo na Web, envolve três grandes atividades:

1. **Criar um grupo de discussão:** atividade acontece só uma vez, pois é executada por uma pessoa que deseja discutir alterações no sistema. Ela deve importar alguns elementos do modelo do sistema a ser discutido, tais como os papéis dos usuários e os tipos de elementos da UIL. Adicionalmente, esta pessoa tem a tarefa de convidar os interlocutores para a discussão, e opcionalmente, fazer a primeira proposta de modificação.
2. **Elaborar e enviar mensagens:** esta atividade acontece iterativamente com a próxima (Análise da mensagem). Os interlocutores alternam entre emissores e receptores sobre as modificações no sistema. “A elaboração de uma mensagem é o processo no qual se dá o mapeamento de um conteúdo para um código expressivo que seja aceitável e compreensível tanto pelo emissor quanto pelo receptor da mensagem”. (Cunha, 2001) Nesta etapa de elaboração, o emissor tem dois pontos de partida a escolher, ou ele inicia a elaboração da mensagem informando elementos como: tipo de fala, assunto, papel do emissor, descrição geral, etc. Ou pode, opcionalmente, começar descrevendo o detalhe da ideia em uma representação concreta e detalhada da intenção de design. Na primeira opção ele utiliza o código de comunicação (LC). Na segunda opção, ele considera como ponto de partida a UIL original e utiliza a Linguagem de Modificação (LM). A sua mensagem necessariamente tem de conter elementos da LC e pode conter ou não elementos da LM.

Uma expressão em LM pode ser “elaborada” por um receptor e, assim, transformar-se em parte de uma nova mensagem, comunicada por ele para seus respectivos interlocutores, ou seja, uma proposta de evolução de sistema pode ser feita a partir de outra proposta.

3. **Analisar mensagens:** Esta análise pressupõe a leitura da parte da mensagem descrita através da LC e da parte da mensagem descrita através da LM, além do estabelecimento da relação entre os conteúdos expressos em LC e LM (se existirem).

O modelo não prevê um controle do fluxo de **discussão** ou de **negociação** das soluções do grupo. Trata apenas da infraestrutura de **comunicação** que viabiliza esta discussão e negociação. Por isto, o fluxo de atividades não tem qualquer previsão sobre como se dá o processo de tomada de decisão. Na prática, este é um ponto de flexibilidade de nosso modelo, que pode então ser articulado com diferentes modelos próprios dos processo de tomada de decisão em grupo.

3.5

Características Representativas do Modelo

Nesta seção discutimos características importantes do modelo. Primeiramente, fazemos uma análise da expansão semiótica do modelo. Em seguida, refletimos sobre: os **níveis de representação** do modelo, os princípios de **Abstração Interpretativa** e **Contínuo Semiótico**, e sobre como utilizamos os **tipos de signos** de Peirce no modelo.

Espaço de Expansão Semiótica do Modelo

Abordamos aqui o quanto o sistema discutido pode ser expandido, ou seja, até que ponto os participantes da discussão podem fazer propostas de modificação radicais (que mudam consideravelmente o sistema).

As mensagens trocadas em nosso modelo têm como particularidade a transmissão de unidades de conteúdo que não estão presentes na aplicação original (justamente porque são “modificações” sobre o sistema). Considerando o sistema de significação original de uma aplicação, a evolução deste sistema tanto pode acontecer no escopo do sistema semiótico da própria aplicação ou pode sofrer influência de outros sistemas de significação de aplicações conhecidas pelo usuário. Por exemplo, é possível que uma modificação seja expressa exclusivamente com duplicações ou reposicionamentos de elementos existentes na UIL do sistema, associando-se a eles anotações e/ou *scripts* que expressam aspectos do seu comportamento e motivação. Este tipo de expansão é conservadora, no sentido de que não adiciona novos “tipos” de elementos de UIL à aplicação. Entretanto, é claro que os usuários têm experiências e cultura computacional originárias de outros sistemas e podem trazer elementos desta cultura para as suas propostas. Estes elementos não fazem parte da UIL (e portanto não podem ser duplicados ou reposicionados). Uma expansão semiótica nesta direção requer que se tenha na LM um vocabulário de elementos de interface independentes da UIL (mas facilmente encontráveis em aplicações típicas da Web). Por exemplo, pode ser que determinado sistema não tenha em sua UIL signo concretizado como lista de itens abertos para múltipla seleção e inserção de novo elemento. Mas, os usuários têm grande probabilidade de conhecerem este elemento e deveriam poder expressar eventualmente uma proposta de modificação utilizando-o. Isto só será possível se o vocabulário de LM tiver uma base lexical que cubra todos os signos típicos de interfaces na web, independentemente de eles ocorrerem na UIL do sistema em discussão ou não.

Existe ainda um tipo de expansão que acontece tipicamente em aplicações de grupo, onde o sistema semiótico de um tipo de usuário é diferente do sistema

semiótico de outro tipo usuário (e.g. quando pessoas têm acesso a funções distintas do sistema). Nestes casos, pode acontecer uma expansão do sistema semiótico de um único tipo de usuário, acrescentando novos significados existentes na aplicação original apenas para um perfil de pessoas. Por exemplo, em um sistema de grupo cujos papéis são coordenadores e secretárias, uma secretária pode sugerir a expansão do sistema de significação com o filtro de consulta que é útil somente para a função que ela executa. Esta nova funcionalidade seria para as secretárias e não para todos os papéis do sistema.

Nível de Representação

O trabalho de Stenning e Oberlander (1995) foi particularmente importante nesta pesquisa porque tivemos que escolher qual a representação adequada para usuários expressarem suas ideias. Tivemos a preocupação de limitar o poder de expressão a fim de manter a identidade da aplicação, mas por outro lado precisamos dar poder de expressão ao usuário de maneira que ele possa comunicar com clareza sua ideia para outras pessoas. Diante disso, consideramos os argumentos dos autores e projetamos uma linguagem cujo sistema de representação se classifica com um LARS. Se imaginarmos o sistema de representação da mensagem do nosso modelo, ele é uma abstração intermediária (LARS) do que seria a metamensagem final da modificação, isto porque ele nem é a representação concreta da ideia (que seria o sistema já modificado expresso muito provavelmente como um MARS (linguagem de interface com um único modelo semântico completamente definido para cada uma de suas possíveis sentenças)), mas é bem próxima dela, sendo um pouco mais abstrata (por causa das anotações e *scripts* associados a elementos da interface).

A linguagem de modificação (LM) do modelo que estamos propondo permite uma representação “concreta” da ideia de modificação do sistema. Ao utilizar a LM, o usuário final não precisa programar sua proposta (o que requer o nível de especificidade que somente um MARS poderia oferecer), ele precisa apenas construir um esboço da sua ideia, apontando, modificando e criando novos elementos na interface que representem adequadamente a sua intenção. Para isso, combinamos linguagens de diferentes níveis de representação. São elas:

1. Uma linguagem de especificação forte, da classe MARS, que cria uma semântica de identificação da forma de expressão de elementos da interface modificada que não é ambígua, cada elemento dela tem um identificador, que é um signo instanciado da linguagem de layout da interface;

2. Uma linguagem de especificação intermediária, no nível LARS, que é a linguagem de macro para representar signos dinâmicos (ações da interface), que cria uma semântica que não é única, porém, limitada por certos tipos previstos de interação;
3. Uma linguagem totalmente livre, no nível UARS, que é a linguagem natural.

Ou seja, fizemos uma combinação de linguagens forte, intermediária e livres, objetivando construir uma linguagem, cujo sistema de representação se encaixa no nível LARS. Portanto, seguimos os argumentos de Stenning e Oberlander (Stenning & Oberlander, 1995) de que sistemas no nível LARS têm maior capacidade de abstração que o MARS e operações mentais de interpretação menos complexas que o UARS.

Contínuo Semiótico e Abstração Interpretativa

A Engenharia Semiótica propôs dois princípios (apresentados no Capítulo 2) para avaliar linguagens de programação para usuários finais. Apesar de não estarmos trabalhando com linguagens de programação, a Linguagem de Modificação (LM) do modelo, serve para expandir o sistema de significação original do sistema que está sendo discutido.

Estes princípios podem ser usados para analisar linguagens, ou sistemas de significação, quando os significados de uma são definidos ou expressos em outra. No nosso caso, usamos para avaliar se a Linguagem de Modificação (LM) é boa ou não. A situação ideal é que ela atenda aos dois princípios.

Nosso modelo não implementa a modificação que está sendo proposta, ou seja, a linguagem para representar a modificação serve para comunicar a ideia sem contudo programá-la. Portanto, o princípio do “Contínuo Semiótico” não se aplica tal como proposto originalmente. Podemos dizer que nossa proposta “presume” que o princípio do Contínuo Semiótico está em vigor, tanto que permite aos usuários expressarem o que querem dizer manipulando elementos da linguagem de interface expandida, incorporando tipos gerais de elementos de interface passíveis de serem usados na expressão de modificações.

Já o princípio de Abstração Interpretativa pode ser observado, pois a semântica que impera na mente dos usuários que utilizam a linguagem de modificação é a semântica da linguagem de interface. Ou seja, eles não precisam conhecer nenhuma linguagem de programação para expressar suas ideias de modificação no sistema. Não precisam conhecer linguagens de programação para falar sobre programação.

Adotando o uso dos tipos de signos da Engenharia Semiótica

A Linguagem de Modificação (LM) é um sistema de representação de um modelo de interface e interações, o qual é formado por: “instâncias” e “tipos” de signos estáticos, “instâncias” e “tipos” de signos dinâmicos, signos metalinguísticos e relações entre os signos estáticos, dinâmicos e metalinguísticos. Para representar signos estáticos, dinâmicos e metalinguísticos o modelo conta com os “tipos” da UIL do sistema de referência, ou seja o sistema que está em discussão. No caso dos signos metalinguísticos, comumente expresso em linguagem natural, os “tipos” dos signos metalinguísticos são todos os tipos gramaticais de uma língua humana que os usuários utilizem nas discussões.

Copiar ou modificar elementos de interface em uma página existente, por exemplo, tipicamente gera novos signos estáticos. Associar *scripts* à ativação de novos elementos de interface é uma forma de incluir novos sinais dinâmicos. Finalmente, anotações sobre elementos de interface novos ou modificados carregam uma série de signos metalinguísticos que podem ser explorados pelos designers do sistema quando estes decidem sobre como deveria ser a aparência de uma nova interface e como ela deveria se comportar quando o sistema é modificado.

Adotando o uso dos tipos de signos de Peirce

As representações das intenções dos usuários, propostas em nosso modelo podem ser expressas através de uma combinação de tipos de signos: ícone (primeiridade), índice (segundidade), ou símbolo (representação descritiva ou argumentativa textual - terceiridade). A ideia principal é que a comunicação sobre modificações de sistemas parta da linguagem de interface, mais precisamente, dos ícones da interface (primeiridade). A partir da criação, remoção ou modificação deste ícones o usuário pode comunicar sua intenção. Adicionalmente, usuários podem acrescentar nomes ou textos explicativos, inserindo *tooltips* (dicas) aos elementos de interface e ainda, podem fazer anotações sobre estes elementos. Estes recursos são meios de representar a intenção utilizando as categorias de “segundidade” e “terceiridade”. O usuário pode também criar *scripts* e associá-los a elementos da interface, criando índices (“segundidade”) de novas funções no sistema.

Assim concluímos esta seção e este capítulo. Iniciamos contextualizando e apresentando o cenário em que o modelo proposto está inserido. Mostramos a estrutura do modelo conceitual e descrevemos cada um de seus elementos. Em seguida, apresentamos o diagrama de classes e o fluxo do modelo.

Por fim, destacamos e discutimos características representativas do modelo. Mostraremos a seguir a implementação de parte deste modelo e relatamos os estudos empíricos realizados.