

MODELAGEM INFORMACIONAL DE REQUISITOS

Michel Heluey Fortuna^{1,2}
Marcos R. S. Borges^{1§}

michelhf@posgrad.nce.ufrj.br , mborges@nce.ufrj.br

¹*Programa de Pós-Graduação em Informática*

Universidade Federal do Rio de Janeiro

²*Universidade Federal de Juiz de Fora*

RESUMO. Vários autores têm apontado problemas na modelagem de requisitos com casos de uso, relacionados com o detalhamento excessivo do comportamento, incompatibilidade com o enfoque de orientação a objetos, e o alcance da análise do domínio do problema. Parte desses problemas podem ser solucionados evitando-se excesso no detalhamento do comportamento através de regras mais objetivas capazes de orientar efetivamente o modelador na escolha dos níveis de abstração a considerar, e dando-se maior atenção ao detalhamento dos fluxos de informações trocados entre o sistema e seus atores. O artigo apresenta uma proposta para a modelagem de requisitos funcionais de sistemas de informação, que se baseia nestas premissas para produzir modelos concisos e abstratos, mais de acordo com o que se espera de uma primeira especificação de requisitos de um sistema.

Palavras-chave: objetivos informacionais, objetivos organizacionais, modelagem informacional.

1. INTRODUÇÃO

Casos de uso representam a interação do sistema com seus usuários [20]. O conceito foi elaborado por Jacobson [11], e posteriormente incorporado à UML [19] em 1996. Desde então, são amplamente utilizados na modelagem dos requisitos de software. Entretanto, vários problemas têm sido apontados como decorrentes do seu uso. Basicamente, estes problemas resultam da ênfase excessiva no detalhamento do comportamento do sistema, da falta de uma regra objetiva para orientar os analistas na escolha dos níveis de abstração a serem adotados na modelagem, e do insuficiente detalhamento da informação que flui entre o sistema e o seu ambiente.

As conseqüências dessas deficiências são, por exemplo, a obstrução a uma compreensão mais profunda do domínio do problema, interferências indevidas no projeto da interface homem-computador, e dificuldades para se obter o modelo de objetos do sistema consistente com a especificação de requisitos.

O presente artigo descreve uma proposta de modelagem de requisitos funcionais de sistemas de informação, que pretende solucionar alguns dos

§ Professor Visitante no Departamento de Sistemas Informáticos y Computación da Universidad Politécnica de Valencia, España.

problemas da modelagem com casos de uso. Ela se baseia em conceitos aqui elaborados, e que constituem uma resposta direta às deficiências acima indicadas. Em especial, o conceito de objetivo informacional substitui o de caso de uso. O qualificador “informacional” associa um nível de abstração específico aos objetivos considerados na modelagem.

A seção 2 destaca três dos principais problemas apontados na modelagem com casos de uso; a seção 3 apresenta o *framework* conceitual e os princípios que fundamentam a solução apresentada na seção 4; a seção 5 relata alguns experimentos realizados com a utilização da modelagem proposta; a seção 6 discute trabalhos relacionados e a seção 7 traz as conclusões e as questões a serem investigadas no futuro.

2. PROBLEMAS DA MODELAGEM COM CASOS DE USO

Dos problemas associados à utilização de casos de uso na modelagem de requisitos de sistemas de informação, destacamos os três abaixo, que serão alvo da solução que propomos mais adiante.

Problema P1: Apoio insuficiente à elicitação de conhecimento do domínio do problema previamente à definição de procedimentos [2] [8] [21, pg.115].

Problema P2: Incompatibilidade com a tecnologia da orientação a objetos, pela ênfase no ordenamento de ações e em decomposição funcional; em especial, dificuldade de transição para o modelo de (classes de) objetos do sistema [18, pg.738] [2] [10]. A modelagem por casos de uso é em grande parte omissa no apoio à determinação da persistência das informações, fundamental para a elaboração do modelo de objetos do sistema. Ao privilegiar a especificação do comportamento em detrimento da informação e seus relacionamentos, o caso de uso está mais próximo dos modelos dinâmicos (modelo de atividades, por exemplo). Isso constitui um sério problema, pois à exceção do modelo de atividades, todos os demais dependem intrinsecamente do modelo de objetos.

Problema P3: Detalhamento excessivo do comportamento com conseqüências negativas para a própria modelagem de requisitos e para outros modelos subsequentes de análise e projeto do sistema (por exemplo, interferência com o projeto da interface homem-computador) [7] [1]. Isso se dá principalmente em decorrência da falta de uma orientação objetiva sobre os níveis de abstração a adotar durante a modelagem.

Estes problemas aumentam o esforço necessário da equipe de desenvolvimento do sistema para preencher as lacunas no conhecimento do domínio da aplicação, e eliminar as inconsistências e tensões entre os modelos produzidos, na busca da qualidade final do sistema e satisfação das reais necessidades dos *stakeholders*.

Em vista disso, vários autores têm proposto soluções que endereçam um ou mais destes problemas [6] [10] [5]. Na seção 6 discutimos duas dessas propostas: Casos de Uso Essenciais [7] e o Marco de Requisitos (*Requirements Baseline*) [14] [15].

3. FRAMEWORK CONCEITUAL E ENFOQUE DA SOLUÇÃO

A solução de modelagem proposta (seção 4) está fundamentada no *framework* conceitual e nos três princípios descritos a seguir.

3.1 *Framework* Conceitual

Informação, processo, comportamento e objetivo são noções usualmente consideradas no âmbito do senso comum. A seguir, apresentamos uma caracterização desses conceitos a ser utilizada no enfoque da solução proposta para os problemas identificados na modelagem por casos de uso.

Informação. É o dado interpretado segundo um contexto. Ela é transformada e/ou comunicada pelos processos do sistema.

Processo. É um conjunto de passos ou atividades, logicamente organizadas e condicionadas, cuja execução visa alcançar um objetivo bem determinado. Ao executar, um processo pode ou não exibir comportamento.

Comportamento (de um processo). É constituído pela seqüência, no tempo, de todo efeito observável a partir do ambiente externo ao processo, desde o início até o término da sua execução. É também uma abstração do conceito de processo, pois ao considerar apenas os efeitos observáveis da execução do processo, abstrai uma série de outros atributos do mesmo, tais como as atividades que o constituem, sua ordem e condições de execução. Em sistemas de informação, a mais importante componente de comportamento é a comunicação de informações pelo sistema a seus atores¹.

Objetivo (de um ator). É a expressão de um desejo de mudança de estado, de si próprio ou do ambiente com o qual ele interage. Como tal, um objetivo fica precisamente definido a partir da explicitação de dois estados: o estado inicial ou de partida, e o estado final ou almejado (estado-objetivo).

O alcance de um objetivo pressupõe a execução de um processo. Então, podemos dizer que a noção de objetivo abstrai não apenas as atividades que o processo associado tem de executar para realizar a mudança de estado, mas também boa parte do comportamento do mesmo, pois não interessa toda a seqüência de efeitos observáveis decorrentes da sua execução, mas apenas o efeito que permite constatar o alcance do (estado-)objetivo. Comportamento é uma expressão dinâmica de processo, enquanto que objetivo uma expressão estática.

3.2 Os Princípios da Solução

Com base no *framework* conceitual acima, podemos estabelecer alguns princípios a serem seguidos pela modelagem proposta. A seguir detalhamos esses princípios e discutimos a sua contribuição na solução de cada um dos problemas identificados na seção 2.

Focar os Objetivos (e não o comportamento)

Objetivo é o nível mais alto de abstração de um processo, e portanto, o mais adequado para uma primeira representação dos requisitos de um sistema. Sendo a expressão de um desejo de um ator, é sempre significativo do ponto de vista do ambiente externo ao sistema e se baseia em (ou é expresso a partir de) conceitos do domínio da aplicação.

Fixar um nível de abstração para os objetivos

Os atores interagem com um sistema buscando alcançar seus objetivos. Para tal, geram **eventos** (estímulos) que disparam os processos do sistema, responsáveis pela realização desses objetivos.

¹ Entidades autônomas, externas ao sistema, e que com ele trocam informações.

As seguintes restrições impostas aos processos que realizam os objetivos induzem um nível de abstração especial para esses objetivos: (1) os processos devem ser atômicos em relação aos eventos gerados pelos atores, ou seja, uma vez iniciada a sua execução ela deve prosseguir até o fim sem interrupções ou temporizações resultantes de eventos gerados pelos atores; (2) os processos só podem se comunicar através de uma memória compartilhada; e (3) quando um processo termina sua execução, todo o sistema fica parado a espera de um evento que irá disparar o processo de um outro objetivo;

O nível de abstração de objetivos assim determinado corresponde a uma estratégia de particionamento funcional do sistema com base no conhecimento do domínio da aplicação, pois é induzido por eventos externos ao sistema. Esta estratégia de particionamento foi proposta inicialmente por McMenamim e Palmer, como um dos fundamentos principais da Análise Essencial [17]. Esses autores apontaram as seguintes vantagens desse tipo de particionamento do sistema: a) fornece resultados razoavelmente uniformes independentemente de quem particiona o sistema; b) resulta em uma modelagem concisa (número relativamente pequeno de processos); c) as interfaces entre os processos tendem a ser mínimas; e d) resulta em processos fiéis à essência do sistema (sem interdependência temporal artificial).

É interessante observar que esta estratégia de particionamento realmente determina um nível específico de abstração para os objetivos, pois o resultado da decomposição de um objetivo deste nível viola as restrições (2) e (3) acima, enquanto que o objetivo resultante da composição de dois ou mais objetivos deste nível viola a restrição (1). Conforme veremos na seção 4, não há necessidade de se realizar decomposição a partir deste nível, mas a composição é útil como mecanismo de agregação de objetivos, gerando outros de nível mais alto no domínio da aplicação.

Focar o detalhamento da informação (e não do comportamento)

Decorre naturalmente da decisão de deslocar o foco da modelagem, do comportamento para os objetivos. Portanto, devemos antes de tudo especificar melhor os objetivos. Cada objetivo exprime uma mudança de estado que é concretizada pela execução do seu processo subjacente. Se a meta é especificar melhor essa mudança de estado (objetivo), não precisamos detalhar indistintamente todo o comportamento do processo subjacente, como faz o caso de uso. Apenas a componente deste comportamento representada pela informação que flui entre o processo e seus atores. Ou seja, devemos detalhar os fluxos de informação que entram e saem do sistema durante a execução do processo. Denominaremos esses fluxos de **interface informacional** do objetivo. Por extensão, passaremos também a denominar os objetivos de **objetivos informacionais**, todos eles pertencentes ao **nível informacional de objetivos**.

Esses princípios, que estão presentes na modelagem de requisitos apresentada na próxima seção, configuram uma solução para os problemas apontados na modelagem por casos de uso. Ao focar nos objetivos e suas interfaces informacionais, ambos com semântica exclusiva no domínio do problema, a solução resolve o problema P1. Além disso, o detalhamento das interfaces promove a elicitação, análise, e definição precisa dessa semântica pelo modelador, contribuindo para a aquisição de conhecimento sobre o domínio. Os problemas P2 e P3 ficam solucionados com a substituição do detalhamento excessivo do comportamento pelo detalhamento da interface informacional dos objetivos do nível informacional. Em especial, a restrição aos objetivos do nível informacional conduz a uma transição simples, quase

automática, para um modelo preliminar de objetos do sistema. As regras para essa transição, e um exemplo da sua aplicação, estão descritos em outro trabalho [9].

4. MODELAGEM INFORMACIONAL DE REQUISITOS

A Modelagem Informacional de Requisitos (MIR) está baseada nos princípios apresentados na seção anterior, devendo ser realizada preferencialmente com a participação ativa dos atores, ou pelo menos, por eles validada. A MIR é composta de três elementos: **Objetivos Informacionais**, **Interface Informacional** e **Objetivos Organizacionais**. A seguir descreveremos cada um desses elementos de modelagem, utilizando para exemplificá-los uma aplicação simples de gerência de restaurante.

4.1 Objetivos Informacionais

Considere a seguinte pergunta que os modeladores devem fazer a si próprios visando eliciar os requisitos do sistema: “Que eventos originados no ambiente externo (pelos atores) exigem intervenção atômica do sistema com trocas de informações (entre o sistema e o ambiente) capazes de mudar o estado do ambiente, do sistema, ou de ambos?” Por intervenção *atômica* queremos significar uma intervenção que se processa sem interrupções ou temporizações controladas pelo ambiente, e que, uma vez concluída, coloca o sistema em estado de espera até o próximo evento. Esses eventos disparam processos do sistema responsáveis pela realização dos objetivos informacionais dos atores. A tabela 1 exemplifica os objetivos de cada ator da aplicação de gerência de restaurante.

Tabela 1 – Objetivos Informacionais da aplicação de gerência de restaurante

Objetivos Informacionais do ator <i>Cliente</i>	Objetivos Informacionais do ator <i>Gerente</i>
1 - Abrir pedido	6 - Atualizar o cardápio
2 - Cancelar pedido	7 - Solicitar consumo diário
3 - Pedir a conta	8 - Solicitar receita
4 - Pagar a conta	9 - Solicitar cópia do cardápio
5 - Pendurar a conta	10 - Cadastrar cliente habitual

Normalmente a MIR é iniciada com a identificação de pelo menos um ator e seus objetivos informacionais. Os atores e objetivos mais relevantes costumam ser rapidamente elicitados. Qualquer dificuldade para descobrir novos objetivos pode servir de indicador da necessidade de se identificar e detalhar a interface informacional dos objetivos já listados.

4.2 A Interface Informacional

A *interface informacional* (de um objetivo) descreve, em detalhe, os fluxos de informações que entram e saem durante o processamento do objetivo pelo sistema. Sua especificação é dividida em duas partes: I) especificação dos fluxos, e II) dicionário de itens elementares.

A figura 1 exemplifica uma parte da especificação dos fluxos da aplicação de gerência de restaurante. Os sinais \rightarrow e \leftarrow indicam, respectivamente, fluxo de entrada e fluxo de saída de informações, do ponto de vista do sistema. Uma informação contida em um fluxo pode ser elementar (indivisível) ou composta (agregado de outras informações elementares ou

compostas). As informações compostas também são chamadas de *pacotes*, e indicadas pelo símbolo \boxplus . A notação utilizada para descrever a composição de um pacote é a seguinte: + significa composição, ${}_n\{x\}_m$ significa de n a m ocorrências de x , () é utilizado para agrupar itens, | significa *ou* e [] delimita informações que nem sempre estarão presentes (podem não ser pertinentes, dependendo do contexto).

ATOR: Cliente	OBJETIVO 1: Abrir pedido
<p>➔ pedido = dt_pedido + nr_mesa + itens_ped</p> <p><u>Descrição:</u> Informações de um pedido de refeição. A mesa identificada por nr_mesa não pode ter pedido em aberto (ainda não pago ou pendurado).</p> <p><u>Propósito:</u> Informar ao sistema o início de uma nova refeição e o que será nela consumido.</p> <p><u>Frequência:</u> 250/dia.</p> <p>\boxplus itens_ped = ${}_1\{\text{id_item} + \text{quant_item}\}$</p> <p><u>Descrição:</u> Informações sobre os itens (pratos e bebidas) a serem consumidos.</p>	
ATOR: Cliente	OBJETIVO 3: Pedir a conta
<p>➔ solicitacao_conta = nr_mesa + [id_cliente]</p> <p><u>Propósito:</u> Identificar o pedido para a emissão da conta correspondente (pedido em aberto na mesa indicada por nr_mesa). Opcionalmente, também identificar o cliente, se ele for um cliente habitual.</p> <p>← conta = nr_conta + nr_mesa + dt_pedido + itens_conta + vl_conta + [nome_cliente + tel_cliente]</p> <p><u>Descrição:</u> Ticket impresso contendo informações sobre o consumo e o valor a pagar. Deve ser emitido mesmo no caso de conta a ser pendurada, caso em que inclui o nome do cliente.</p> <p><u>Propósito:</u> Permitir a conferência, pelo cliente, do que ele consumiu e do valor a pagar.</p> <p>\boxplus itens_conta = ${}_1\{\text{nome_item} + \text{pç_unit} + \text{quant_item} + \text{pç_item}\}$</p> <p><u>Descrição:</u> Informações sobre os itens (pratos e bebidas) consumidos.</p> <p><u>Ordenação:</u> Crescente de nome_item.</p>	

Figura 1: Fluxos informacionais da aplicação de gerência de restaurante

A especificação de fluxos e pacotes inclui, além da sua composição, alguns tópicos descritivos. Quanto a esses cabe esclarecer: a) o tópico descritivo *Ordenação* é aplicável apenas se a informação for multivalorada e fizer parte de um fluxo de saída; neste caso, fornece informações sobre a ordenação dos valores na saída; b) o tópico descritivo *Frequência* informa quantas vezes, por unidade de tempo, o fluxo de informação acontece; tem por finalidade chamar a atenção do projetista do sistema para os grandes volumes de entrada ou de saída de informações. A tabela 2 exemplifica o dicionário de itens do objetivo 1, e mostra os tópicos descritivos utilizados para especificar os detalhes de cada informação elementar.

4.3 Os Objetivos Organizacionais

Um **objetivo organizacional** é um conjunto de “seqüências admissíveis” de objetivos informacionais, que representa uma “linha de trabalho” ou um “objetivo de negócio” relevante no domínio da aplicação, para um ou mais atores. Por exemplo, no caso da aplicação de gerência de restaurante, os

conjuntos $Tomar\ refeição = \{(1\text{-Abrir pedido, 3-Pedir a conta, 5-Pendurar a conta}), (1\text{-Abrir pedido, 3-Pedir a conta, 5-Pendurar a conta, 4-Pagar a conta})\}$ e $Cancelar\ refeição = \{(1\text{-Abrir pedido, 2-Cancelar pedido})\}$ representam objetivos organizacionais do ator *Cliente*.

Tabela 2: Dicionário de itens de informação elementar do objetivo 1

Ator: Cliente		Objetivo 1: Abrir pedido	
Nome	Descrição	Tipo	Domínio
dt_pedido	Data em que o pedido foi efetuado.	Data	
nr_mesa	Ident. de uma mesa do restaurante	Nr.Natural	{1..30}
id_item	Identificador de um item pedido	Nr.Natural	Gerado na atualização do cardápio.
quant_item	Quantidade pedida do item.	Nr.Natural	

A obtenção das seqüências admissíveis de objetivos informacionais exige o conhecimento de duas relações definidas sobre os objetivos. Se o_1 e o_2 forem objetivos informacionais, a relação de **dependência temporal**² dt , é definida da seguinte maneira: $(o_1, o_2) \in dt$ sss o_2 só pode ocorrer se o_1 ocorrer antes. A outra relação, de **incompatibilidade** ic , é definida assim: $(o_1, o_2) \in ic$ sss o_2 não pode ocorrer depois de uma ocorrência de o_1 . Essas duas relações geram um grafo orientado cujos nós representam os objetivos, e as transições entre os nós representam os elementos das relações. A figura 2 mostra o grafo obtido a partir dos objetivos do ator *Cliente*. Os nós iniciais são aqueles que só possuem transições de saída, enquanto os nós finais só possuem transições de chegada. Então, uma **seqüência admissível** de objetivos informacionais é qualquer caminho completo neste grafo que represente um “objetivo de alto-nível” de um ou mais atores, no domínio da aplicação.

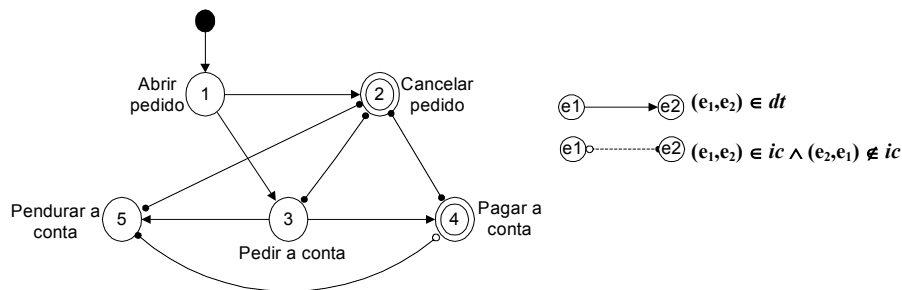


Figura 2. Grafo de admissibilidade dos objetivos informacionais do ator *Cliente*.

A cada novo objetivo informacional descoberto, pode-se atualizar a lista de objetivos organizacionais da aplicação. A primeira utilidade desse conceito é permitir agregar objetivos informacionais em abstrações de mais alto nível com forte significado no domínio da aplicação, conferindo assim escalabilidade à MIR. Outras vantagens do conceito de objetivo organizacional são: a) ajuda na validação dos objetivos informacionais, sendo facilmente percebido e validado pelos atores; b) pode sofrer uma verificação

² Relação não reflexiva e transitiva.

formal de *consistência funcional*³, capaz de detectar a existência de objetivos informacionais inalcançáveis, através da análise dos fluxos informacionais que nele ocorrem; c) estabelece um critério centrado nos atores para organizar e priorizar a implementação e os testes do sistema; d) facilita a definição dos testes de integração do sistema.

5. UTILIZAÇÃO DO ENFOQUE E FERRAMENTAS DE APOIO

A MIR tem sido aplicada em uma série de iniciativas educacionais, de pesquisa, e também na indústria. No campo educacional, ela vem sendo utilizada com alunos de graduação em *Administração de Empresas* da Universidade Federal de Juiz de Fora (UFJF). Os alunos são introduzidos ao método e desenvolvem o modelo informacional de uma aplicação. Em seguida derivam e implementam (em *MS Access*) o modelo de classes associado, e constroem consultas SQL que são executadas para concretizar algumas das saídas especificadas no modelo informacional. Essa experiência, já de vários anos e com excelentes resultados, tem demonstrado o potencial da MIR para a prototipação rápida de sistemas de informação, mesmo por profissionais não especializados em computação.

Como objeto de pesquisa, a MIR foi aplicada em vários estudos de caso, resultando em modelos informacionais de várias aplicações, tais como gerência de bar, restaurante, vídeo-locadora, biblioteca, hotel, agendamento de ações e acompanhamento de projetos, agendamento de reuniões [12] etc. Alguns desses modelos foram implementados em sistemas que são utilizados rotineiramente.

Mais recentemente, foi empreendido um experimento de maior porte envolvendo uma equipe com 1 engenheiro de requisitos, 4 analistas-programadores e 4 especialistas de domínio, no desenvolvimento de um sistema *web* multiusuário⁴, para apoiar as atividades de uma empresa de medicina e engenharia de segurança do trabalho. O sistema foi desenvolvido pelos analistas em PHP e MySQL, a partir do modelo informacional elaborado pelo engenheiro com o apoio e validação dos especialistas, constituído de 30 objetivos informacionais e 4 objetivos organizacionais, totalizando 40 páginas de especificações. O desenvolvimento demandou 4 meses de trabalho e foi considerado um grande sucesso, tanto pela equipe de desenvolvimento quanto pelos especialistas e *stakeholders*. O sentimento dos especialistas e dos desenvolvedores com relação à MIR foi sempre positivo. Alguns aspectos que valorizam os resultados obtidos são: a) nenhum dos analistas tinha conhecimento e experiência anteriores em MIR e no domínio da aplicação; b) os analistas trabalharam os 3 primeiros meses praticamente sem contato presencial com o engenheiro de requisitos, sendo a comunicação realizada por escrito via um software do tipo *instant messenger*; c) as especificações foram elaboradas em paralelo com o desenvolvimento do sistema, cada objetivo informacional sendo liberado assim que a sua definição ficava pronta; d) toda a divisão de trabalho entre os analistas foi feita por objetivo informacional; e) o esquema conceitual e o projeto do BD do sistema foram elaborados de forma incremental (por objetivo) pelos analistas, sem a participação do engenheiro. Como já se esperava, os problemas do

³ Um processo não pode gerar informação do nada. Ela deve ser obtida diretamente do ambiente externo ao processo ou indiretamente pela transformação de outras informações provenientes do ambiente.

⁴ 45 tabelas, 100.000 linhas de código.

software resultante ficaram, na sua grande maioria, no nível da interface homem-computador, por conta da falta de um projeto.

Por ora, os modelos da MIR são elaborados com o editor de texto *MS Word* e um aplicativo de banco de dados *MS Access*. O editor é empregado na especificação dos fluxos informacionais (figura 1), sendo a edição facilitada pela utilização de um modelo de documento e estilos pré-definidos. O aplicativo de banco de dados oferece formulários para o cadastramento dos diversos elementos da MIR (atores, objetivos, itens de informação), e relatórios para apresentar: (1) os objetivos informacionais por ator (tabela 1), (2) o dicionário de itens elementares por objetivo informacional (tabela 2) e (3) a composição dos objetivos organizacionais. Apesar de prestar um importante apoio ao processo de modelagem, este ferramental disponibiliza uma funcionalidade muito aquém daquela que vislumbramos ser possível. A próxima geração de ferramentas deverá tratar a especificação da interface informacional de forma integrada, implementar a automatização possível em várias tarefas ainda realizadas manualmente (por exemplo, a verificação de consistência funcional e a derivação do modelo de objetos), e incorporar requisitos de *groupware*.

6. TRABALHOS RELACIONADOS

Dentre as propostas de modelagem de requisitos existentes na literatura, destacamos duas que constituem, a nosso ver, soluções parciais para a problemática apontada na seção 2. Em especial, a primeira proposta ataca principalmente o problema P3, enquanto que a segunda, o problema P1. Nesta seção, além de descrevê-las brevemente, procuramos identificar pontos em comum e diferenças entre elas e a MIR.

6.1 Casos de Uso Essenciais

Segundo Biddle [3], um **caso de uso essencial** (CUE) é uma “narrativa estruturada, na linguagem do domínio da aplicação e dos usuários, compreendendo uma descrição simplificada, geral, abstrata, independente de tecnologia e de implementação, de uma tarefa ou interação completa, significativa e bem definida do ponto de vista dos usuários que desempenham um ou mais papéis em relação ao sistema, e que incorpora o propósito ou intenções subjacentes à interação”.

De acordo com os seus idealizadores [6], enquanto um “caso de uso de Jacobson” usualmente assume uma interface de usuário em particular, o CUE “equivalente” se abstrai dela, resultando em uma versão simplificada para um mínimo de interação. Sua criação foi inspirada no conceito de **modelagem essencial** desenvolvido por McMenamim e Palmer [17]. Desta forma, trilha o mesmo caminho da MIR, embora apenas parcialmente. Isso porque os idealizadores do CUE visavam, primordialmente, uma melhor integração da modelagem de requisitos por casos de uso com o projeto da interface homem-computador.

Em contraposição à MIR, e a despeito da inspiração comum na Análise Essencial, este enfoque não utiliza o conceito de particionamento por eventos externos. Esta forma de particionamento é um critério mais objetivo para a elicitación dos casos de uso do que simplesmente procurar por “tarefa ou interação completa, significativa e bem definida do ponto de vista dos usuários”. Além disso, o CUE mantém o foco no detalhamento do comportamento, permitindo a sua (de)composição da mesma forma prevista

nos casos de uso tradicionais. A modelagem com CUE's mantém o *gap* entre o modelo de requisitos e o modelo de classes (de objetos) do sistema. Estratégias para a descoberta das classes, como a proposta em [3], que se baseia na frequência das palavras que ocorrem nas descrições de responsabilidades do sistema, não são, a nosso ver, uma solução definitiva para o problema.

A configuração das diferenças de grau de detalhamento da informação e do comportamento essencial, entre a MIR (alto e baixo, resp.) e a modelagem por CUE's (baixo e alto, resp.), sugere um tipo de modelagem onde o detalhamento nessas dimensões seria feito de forma integrada e complementar.

6.2 O Marco de Requisitos (*Requirements Baseline*)

Leite e Oliveira propuseram em [14] um marco conceitual de requisitos (*requirements baseline*) como ponto de partida para a especificação de requisitos de sistemas de informação, que utiliza fundamentos da Análise Essencial [17], em especial o conceito de eventos externos. Segundo os autores, a elaboração do marco deve ser precedida por um processo de elicitación, durante o qual os engenheiros de software registram requisitos e definem o vocabulário do domínio da aplicação segundo um modelo denominado Léxico Estendido da Linguagem (LEL). Mais tarde, Leite et al. [15] incrementaram o marco com a introdução de cenários, visando aperfeiçoar o processo de elicitación e aumentar a sua capacidade de exprimir comportamento. Segundo esses autores, o marco de requisitos não pode ser considerado uma especificação de requisitos, mas uma base de informações a partir da qual ela é construída.

O marco de requisitos tem muito em comum com a MIR, pois ambos estão fundamentados na Análise Essencial [17], especialmente nos eventos externos com seus fluxos de entrada e de saída de informações. Da mesma forma que na MIR, os modelos resultantes do marco de requisitos estabelecem uma decomposição funcional do sistema em suas atividades essenciais, e delimitam uma fronteira entre o sistema e o seu ambiente (atores). Além da influência da Análise Essencial, as duas propostas têm em comum a preocupação com o domínio da aplicação que modelam, e a possibilidade de compor eventos externos em entidades de nível mais alto, denominadas *ações* no marco de requisitos, e *objetivos organizacionais* na MIR.

Entretanto, a MIR difere do marco de requisitos em vários aspectos. Primeiramente, a MIR associa intimamente a noção de objetivo aos eventos externos, e dá ênfase ao detalhamento da informação abstraindo boa parte do comportamento. Em segundo lugar, enquanto o marco de requisitos procura principalmente elicitar e estruturar o conhecimento do domínio da aplicação, a MIR visa obter uma especificação funcional de requisitos, possível graças ao grau de detalhamento e precisão na definição da interface informacional de cada objetivo. Finalmente, a MIR introduz uma nova relação entre objetivos informacionais – a relação de incompatibilidade – que não tem similar no marco de requisitos. Por outro lado, o marco modela explicitamente requisitos não funcionais, o que a MIR não contempla na versão atual.

7. CONCLUSÕES E TRABALHOS FUTUROS

Motivados pela busca de uma solução para três problemas identificados com a modelagem por casos de uso, e utilizando o conceito de objetivo

informacional, apresentamos um novo modelo para a especificação de requisitos de sistemas de informação. O modelo foi avaliado preliminarmente através de alguns estudos de caso, incluindo casos reais. Além de resolver os problemas indicados, este modelo apresenta outras vantagens em relação ao modelo de casos de uso, resultando mais adequado para uma primeira especificação dos requisitos de uma aplicação. As principais vantagens são:

- a) é mais conciso e abstrato, capturando um mínimo de comportamento dos atores e do sistema e evitando interferência posterior indevida no modelo da interface homem-computador;
- b) permite uma transição mais fácil (semi-automática) para o modelo de objetos do sistema;
- c) promove a elicitación e análise do conhecimento relevante do domínio da aplicação, antes da definição dos procedimentos a serem incorporados ao sistema;
- d) incorpora uma orientação mais clara e precisa sobre os níveis de abstração a considerar na elicitación de objetivos (casos de uso), além de definir regras precisas para se passar de um nível para outro;
- e) permite uma verificação formal de consistência funcional dos objetivos;
- f) da mesma forma que o modelo de casos de uso, o MIR fornece um critério simples, centrado no usuário, para organizar e priorizar a implementação e os testes do sistema.

Restam várias questões a serem investigadas para aprofundar a compreensão da MIR visando o seu aperfeiçoamento e um posicionamento mais claro em relação a outras propostas. Em primeiro lugar, embora os objetivos da MIR tenham vários aspectos comuns com os da engenharia de requisitos orientada por objetivos [13], como por exemplo, representarem desejo ou intenção dos agentes e poderem ser formulados em diferentes níveis de abstração, também há diferenças, como a inexistência na MIR dos chamados objetivos terminais. A relação entre as duas propostas deve ser investigada. Outra questão a considerar é como a MIR pode contribuir para a obtenção de uma ontologia de aplicação como subproduto da especificação de requisitos [4]. Acreditamos que a ênfase da MIR nas interfaces informacionais dos objetivos pode favorecer essa obtenção.

Outra questão a investigar é o papel de cenários na MIR. Um objetivo organizacional pode ser visto como representando um conjunto de cenários de alto-nível (ou baixa granularidade). Assim, ao mesmo tempo em que funciona como mecanismo de abstração (objetivo), também fornece uma percepção concreta de comportamento, ancorada na realidade (cenário). Para tais cenários, a recomendação de estarem associados a objetivos [15] fica atendida por construção. Pretendemos investigar a extensão do papel que os cenários podem desempenhar na MIR. Por fim, estaremos buscando formas de integrar a MIR com a modelagem organizacional [16].

Agradecimentos

Os autores agradecem aos revisores anônimos e a Tarcísio de Souza Lima e Carlos Henrique Cabral Duarte pela revisão da versão preliminar deste artigo. Marcos R.S. Borges foi apoiado com uma bolsa da Secretaria de Estado de Educación y Universidades do Governo Espanhol.

REFERÊNCIAS

- [1] Anderson, D.J. *Are Use Cases the death of good UI Design?*, Fevereiro, 1999. http://www.uidesign.net/1999/imho/feb_imho.html, acesso em Abril, 2005.
- [2] Anderson, D.J. *Use Cases still considered Dangerous!*, Outubro, 1999. http://www.uidesign.net/1999/imho/feb_imho.html, acesso em Abril, 2005.
- [3] Biddle, R. et al. *From Essential Use Cases to Objects*. Anais forUSE 2002.
- [4] Breitman, K.K. e Leite, J.C.S.P. *Ontology as a Requirements Engineering Product*, Anais 11th IEEE International Requirements Engineering Conference, Califórnia, EUA, Setembro, 2003.
- [5] Cockburn, A. *Writing Effective Use Cases*. Addison-Wesley, 2001.
- [6] Constantine, L.L. *What Do Users Want? Engineering Usability into Software*. Constantine & Lockwood, Junho, 2000.
- [7] Constantine, L.L. e Lockwood, L.A.D. *Structure and Style in Use Cases for User Interface Design*. Em: Object-Modeling and User Interface Design, Addison-Wesley, 2001.
- [8] Davis, L. e Dawe, M. *Collaborative Design with Use Case Scenarios*. ACM JC'DL'01, Junho, 2001.
- [9] Fortuna, M.H. *Derivando um Modelo de Objetos a partir da MIR*. Outubro, 2004. <http://www.mhf.ufjf.br/DCRest01b.doc>. Acesso em Abril 2005.
- [10] Glinz, M. *A Lightweight Approach to Consistency of Scenarios and Class Models*. Anais 4th Intl. Conf. on Requirements Engineering (ICRE'00), 2000.
- [11] Jacobson, I. et al. *Object-Oriented Software Engineering - A User Case Driven Approach*. Addison-Wesley, 1992.
- [12] Lamsweerde, A. *The Meeting Scheduler System - Problem Statement*. Universidade Católica de Louvain, 1992.
- [13] Lamsweerde, A. *Goal-Oriented Requirements Engineering: A Guided Tour*. Anais 5th IEEE Intl.Symposium on Requirements Engineering (RE'01), Agosto, 2001.
- [14] Leite, J.C.S.P. e Oliveira, A.P.A. *A Client Oriented Requirements Baseline*. Anais 2nd Intl.Symp. on Requirements Engineering (RE95), IEEE Computer Society Press, 1995.
- [15] Leite, J.C.S.P. et al. *Enhancing a Requirements Baseline with Scenarios*. Requirements Engineering Journal, vol.2, n.4, pags.184-198, 1997.
- [16] Martínez, A. et al. *Closing the gap between Organizational Modeling and Information System Modeling*. Anais VI Workshop em Engenharia de Requisitos, Piracicaba, São Paulo, Brasil, Novembro, 2003.
- [17] McMenamim, S.M. e Palmer, J.F. *Essential Systems Analysis*. Prentice-Hall, 1984.
- [18] Meyer, B. *Object-Oriented Software Construction (2nd ed)*. Prentice-Hall, 1997.
- [19] Object Management Group (OMG). *UML 2.0 Specification*. <http://www.uml.org/>. Acesso em Abril, 2005.
- [20] Pender, T. *UML Bible*, John Wiley & Sons, 2003
- [21] Wálden, Kim. e Nerson, J-M. *Seamless Object-Oriented Software Architecture - Analysis and Design of Reliable Systems*. Prentice-Hall, 1995