UTILIZANDO LÓGICA MODAL DE AÇÕES PARA A FORMALIZAÇÃO DE REQUISITOS ORGANIZACIONAIS

Fernanda M. R. de Alencar
UFPE - CT - Departamento de Eletrônica e Sistemas
Rua Acadêmico Hélio Ramos, s/n - Cidade Universitária
Recife - PE - CEP: 50.740-530
e-mail: fmra@ di.ufpe.br

Jaelson B. Castro
UFPE - CCEN - Departamento de Informática
Av. Prof. Luiz Freire, s/n - Cidade Universitária
Recife - PE - CEP: 50.740-540
e-mail: jbc@ di.ufpe.br

RESUMO

Componentes organizacionais tratados como agentes oferecem um bom meio de compreender os inter-relacionamentos entre esses agentes e de visualizar, de forma mais clara, como esses inter-relacionamentos podem ser alterados, com o objetivo de melhorar os processos, identificando e selecionando alternativas que antes não tinham sido objeto de preocupação. Com este enfoque, neste trabalho utilizamos a Lógica Modal de Ações (MAL) para especificar os requisitos de um sistema bancário de transferência de fundos, especificando os requisitos em termos de estados e ações. Ao mesmo tempo é apresentada a técnica i*, capaz de reconhecer motivações, intenções e razões sobre as características dos processos, fazendo-se um paralelo entre as duas formas de modelagem de requisitos.

PALAVRAS CHAVE: Formalização, Requisitos Organizacionais, Requisitos Funcionais, Técnica I*, MAL, Modelagem.

1. INTRODUÇÃO

A fase crucial no ciclo de vida do desenvolvimento de Sistemas de Software, por tratar conhecimentos técnicos, gerenciais, organizacionais, econômicos e sociais tem sido a Engenharia de Requisitos (E. R.). A fim de apoiar a Engenharia de Requisitos a modelar e capturar os desejos dos clientes que não sabem, exatamente, o que pretendem [1], surgem várias abordagens e linguagens com grande poder de expressão e formalidade, capazes de refletir a situação desejada. Hoje, as técnicas de requisitos estão mais associadas às últimas fases da engenharia de requisitos, concentrando-se na completude, consistência e na verificação automática dos requisitos. Entretanto, é a fase inicial da Engenharia de Requisitos, feita informalmente e sem nenhuma ferramenta de suporte, que ajuda a modelar e analisar as intenções e desejos, uma vez que esses requisitos iniciais são ambíguos, incompletos e inconsistentes. Quanto mais complexo fica o domínio do problema mais evidente se torna a necessidade de uma técnica capaz de representar o conhecimento e suportar as razões e motivações envolvidas. Somente recentemente foram propostas técnicas orientadas a objetivos que envolvem múltiplos agentes. Dentre essas, destacamos a técnica i* [2]. Nesta técnica, os objetivos estão (i) associados a um conjunto de características que descrevem as restrições sobre o comportamento desejado do sistema, ou (ii) associados a regras organizacionais ou de negócios que permitem analisar as razões que conduziram à arquitetura daquele sistema. Desta forma, com a compreensão do contexto organizacional e das razões e motivações envolvidas, os requisitos podem ser mais bem definidos, para o sucesso do sistema.

O objetivo desse trabalho é utilizar técnicas de formalização, no caso específico, a Lógica Modal de Ações [3], [4], para transformar requisitos organizacionais, tratados pela técnica i*, em especificação funcional. A Lógica Modal de Ações (MAL) foi desenvolvida a fim de (i) produzir um formalismo matemático útil à especificação de requisitos, sem a necessidade de que o especificador tenha grande treinamento matemático, e de (ii) produzir um documento de especificação de requisitos do sistema em desenvolvimento.

Na seção 2, trabalhos relacionados são brevemente apresentados. É feita uma breve revisão das principais

fundos. Na seção 4, são apresentados as principais características da técnica i*, utilizando-se o mesmo exemplo da seção anterior. Na seção 5, apresentamos como os dois níveis de modelagem podem trabalhar em conjunto. Concluímos o artigo, tecendo considerações na seção 6.

2. TRABALHOS RELACIONADOS

Em [2] demonstrou-se (i) que o conceito de agentes cooperantes pode (a) oferecer um bom entendimento dos objetivos e relacionamentos organizacionais e (b) oferecer fixação e análise das especificações de requisitos, e (ii) porque a cooperação precisa ser compreendida em termos de conceitos de intenções, tais como, conhecimento, compromisso, obrigações e objetivos. A técnica i reflete a dimensão estratégica dos relacionamentos dos agentes e diminui a ênfase dos aspectos operacionais.

Em [5] foi feito um trabalho similar ao nosso, onde se buscou relacionar a técnica i* à linguagem de especificação ALBERT [6]. Os autores concentraram-se em mostrar como o ALBERT e o i* podem trabalhar em conjunto. A abordagem usando o ALBERT e i* é comparável, mas adota explicitamente um conjunto de conceitos intencionais, com semânticas mais precisas , podendo ser comparada a outras ferramentas da engenharia de requisitos que tomam uma perspectiva de multi-agente ou organizacional. Ao nível de especificação, a linguagem ALBERT está na linha de linguagens de especificação formal, projetadas com o propósito de modelagem de requisitos funcionais. A principal diferença é o escopo de aplicação do ALBERT relacionado à modelagem de sistemas cooperativos complexos de tempo real.

A técnica de Bubenko [7] é similar, em espírito, de várias maneiras. Ela enfatiza a necessidade de modelar organizações e seus atores, suas motivações e razões, utilizando, também, modelos interligados (inter-linked) e múltiplos. Os modelos organizacionais informais (mas estruturados) são ligados a modelos de especificação mais formal..

No KAOS [8] todos os objetivos são explicitamente modelados (segue o conceito de Projetos de Composição de Sistemas). Os objetivos são simplificados (reduced) através de raciocínio meio-fim para chegar às responsabilidades dos agentes. A modelagem de agentes é especificacional e prescritiva. Desde que agentes são assumidos obedecer ao comportamento prescrito, não se pode facilmente analisar os relacionamentos estratégicos e implicações.

3. LÓGICA MODAL DE AÇÕES

Nesta seção serão revistos conceitos principais da Lógica Modal de Ações (MAL) [3] [4] e como esses podem ser usados para descrever o comportamento de objetos (atores) em um sistema.

MAL tem por base a lógica de primeira ordem tipada. A sua definição inclui tipos, variáveis, símbolos lógicos, predicados, símbolos funcionais, símbolos constantes, termos, fórmulas atômicas, fórmulas e um número de axiomas e regras de inferências.

Uma extensão ao MAL é feita através da adição de: (a) tipos pré-definidos chamados "ações" e "agentes"; (b) uma modalidade [ ] ; (c) operadores deônticos per (permissão) e obl (obrigação); (d) combinadores; e (e) uma lógica temporal intervalar do tipo linear com ramificações. Os agentes definem entidades do mundo real e as ações descrevem os processos que os agentes podem executar. O operador modal [ ] é utilizado para capturar o efeito da ocorrência de ações, ou seja, provê a noção de que entidades do mundo real realizam tarefas, sendo o predicado considerado como uma pós-condição ou resultado de uma ação que foi completada, permitindo, assim, a definição da estrutura do sistema. O operador deôntico permite o controle sobre quais ações podem ser executadas pelos agentes, descrevendo se determinadas ações são permitidas (per) ou obrigadas (obl) a ocorrerem , definindo o comportamento do sistema.

Uma especificação em MAL estruturado corresponde a um conjunto descrições de agentes (ou objetos), onde essas descrições são formadas por um conjunto de declarações e axiomas que definem o comportamento desses agentes que podem interagir compartilhando atributos e ações (rótulo s - shared). Alguns atributos ou ações podem ter efeito apenas local, sendo neste caso rotuladas de l - local.

Considere como exemplo um sistema de transferência bancária simplificado, composto de agentes (objetos) de três tipos: Controlador de Contas (Accounts Handler), Cliente (Customer) e Funcionário (Teller). A especificação parcial do sistema e seus componentes (objetos) pode ser vista nas Figuras 1 a 3. Na primeira figura definimos o sistema Banking System como sendo composto de (inclui) objetos do tipo Customer, Teller (especificados parcialmente na figura 2) e Account Handler (vide Figura 3). Duas propriedades de interação entre os componentes do sistema são descritas. Se o cliente pedir a transferência de fundos entre contas, então o caixa (funcionário do banco) solicitará sua realização. Uma solicitação de transferência feita pelo funcionário do banco será seguida de ações débito e crédito no objeto que gerencia as contas do banco (Accounts Handler).

Object Banking System
.....
includes
Customer:
Accounts Handler:
Teller
end
axioms
Customer.TransferR(a1,a2,n) →Teller.Transfer(a1,a2,n)
A transfer request from the Customer is echoed by a transfer order sent to the Teller
Teller.Transfer(a1,a2,n) → Accounts Handler.Debit(a1,n)
*Accounts Handler.Credit(a2,n) **end**
A transfer order should be followed by the corresponding credit and debit operation
end Object Banking System

Fig. 1 - Sistema de Transferência Bancária

Vide na figura 2 a descrição em MAL dos objetos Customer e Teller. O objeto **Customer** (cliente), pode apenas solicitar a transferência de fundos entre suas contas (TransferR). Por ser uma ação compartilhada, ela será usada na interação com outros componentes, no caso o objeto **Teller** (funcionário do banco). Observe que está ação está sempre habilitada para ocorrer (permitida). Já o objeto **Teller** (caixa do banco) será responsável por verificar a assinatura do cliente, conferindo por exemplo sua assinatura. Será responsável por solicitar ao objeto que controla as contas do banco (Accounts Handler) a efetiva transferência dos fundos, após receber uma solicitação do cliente.

O objeto controlador das contas do banco (**Accounts Handler**) é descrito na Figura 3. Os atributos, valor a ser transferido (v), número da conta (acc#) e cliente (c) assumem valores inteiros. Os axiomas informam que inicialmente o saldo da conta **acc#** é nulo, e que após a ação de crédito (Credit(acc#,v)) ser executada, o saldo desta conta **acc#** será acrescido de v. Da mesma forma, caso haja saldo suficiente (ou seja a conta não fique negativa), após a ação de débito (Debit(acc#,v)) ser executada, o saldo da conta **acc#** será diminuído de v. Não é especificado o que acontece caso a conta fique negativa. Por exemplo, dentro do limite de crédito concedido ao cliente, poderia ter sido optado debitar automaticamente nas contas de investimento ou de poupança do cliente. O axioma deôntico (**per**) descreve que será permitido efetuar um débito (Debit(acc#,v)), mesmo que a conta não tenha saldo suficiente (até o limite de RS 2.000,00 por exemplo).

4. A TÉCNICA I*

Nesta seção são revisados os principais conceitos da técnica **i***, sendo ilustrada sua utilização através de um estudo de casos ao mesmo sistema bancário de transferência de fundos da seção anterior e a um sistema de transferência de fundos via telefone.

Tentando-se compreender uma organização, os conhecimentos capturados nos diversos modelos não são, frequentemente, bastante. A maioria dos modelos existentes fornecem uma descrição das funções, mas não conseguem expressar as razões envolvidas no processo. Faz-se necessário uma ontologia mais rica, capaz de reconhecer motivações, intenções, raciocínios sob as características de um processo. A técnica **i*** [2] permite a descrição dessa ontologia e é utilizada para (i) obter uma compreensão maior sobre os relacionamentos organizacionais entre os vários agentes de um sistema, (ii) para entender as razões envolvidas, e (iii) para ilustrar as várias características de modelagem que podem ser apropriadas à Engenharia de Requisitos, principalmente na fase inicial da especificação dos requisitos [5]. Nesta técnica, os participantes do processo de software são atores com propriedades intencionais, tais como, objetivos, crenças, habilidades e compromissos. Esses atores dependem

```
Object Customer
    attributes
        v : int;
        s acc1#; acc2# :   int
        s  Local_request (acc1#)  : bool  end
    actions
        s TransferR(acc1#,acc2#,v)  end
    axioms
        [TransferR(acc1#,acc2#,v) ] Local_request(acc1#)
        per(TransferR(acc1#,acc2#,v)) end
/ In any situation, the customer is permitted to send his or her transfer /
    end Object Customer
Object Teller
    attributes
        v : int;
        l signature_ok(acc1#) : bool,
        s acc1#; acc2#:     int  end
    actions
        l Check_signature(acc1#)
        s Transfer (acc1#,acc2#,v)    end
    axiom
        [Check_signature(acc1#)] signature_ok(acc1#) ∨ ¬signature_ok (acc1#)
        per(Transfer(acc1#,acc2#,v)) → Local_request (acc1#) ∧ signature_ok (acc1#)
/The transfer is permited if and only if a request exist  and the signature  of the client is true/
        Local_request(acc1#) → obl [Check_signature (acc1#)  end
    end Object Teller
```

Fig. 2 - Componentes Customer e Teller

uns dos outros para terem seus objetivos alcançados e suas tarefas realizadas, além de objetivos que não seriam individualmente alcançados, passarem a ser. A técnica **i*** é composta por dois modelos: o modelo de Dependências Estratégicas (SD) e o modelo de Razões Estratégicas (SR). São descritas, a seguir as características básicas de cada um dos modelos, maiores detalhes podem ser encontrados em [2].

O modelo de Dependências Estratégicas (SD) consiste em um conjunto de nós e elos de ligação entre esses, onde os nós representam os atores e cada elo indica uma dependência entre dois atores. Assim, um processo é descrito em termos de uma malha de relacionamentos de dependências entre os vários atores, capturando-se as motivações e os desejos envolvidos nas atividades. Distingue-se quatro tipos de dependências, das quais três representam as intenções existentes - dependências de objetivos (goals), de recursos (resources) e de tarefas (tasks) - e a quarta está relacionada com a noção de requisitos não-funcionais na Engenharia de Requisitos - dependência de objetivos-soft (soft-goals). Na dependência de objetivo, um agente depende que um outro forneça a condição desejada, não interessando de que maneira a condição é alcançada. Na dependência de tarefa, um agente informa ao outro o que deve ser feito, sem informar o porque fazer. Na dependência de recurso, o agente depende da disponibilidade de um recurso físico ou de uma informação. A dependência de objetivo-soft é similar à dependência de objetivos, exceto que a condição não é precisamente definida a priori, ou seja, objetivos envolvem aspectos subjetivos que vão sendo mais esclarecidos durante o desenvolvimento do processo. Ainda nesse modelo SD podem ser identificados (i) diferentes graus de dependências: aberta (open), compromissada (committed) e crítica (critical) [9], (ii) distinguir-se *agentes*, *papéis* e *posições* e (iii) fazer-se uma análise em termos de oportunidades e vulnerabilidades [10].

O segundo modelo que compõe a técnica **i***, é o modelo de Razões Estratégicas (SR) usado para: (a) descrever os interesses, preocupações e motivações dos participantes de um processo; (b) possibilitar avaliação das possíveis alternativas de definição do processo; (c) investigar mais detalhadamente as razões existentes por detrás das dependências entre os vários atores. Esse modelo é composto, também, por nós e elos que juntos fornecem uma estrutura para expressar as razões envolvidas nos processos. Existem (i) quatro tipos de nós, que se baseiam nos tipos de dependências do modelo SD: objetivo (goal), tarefa (task), recurso (resource) e objetivos-soft (soft-goal); (ii) dois tipos principais de relacionamentos: meio-fim (means-ends) e decomposição de tarefas (task-decomposition). Os relacionamentos meio-fim sugerem que podem existir outros meios de alcançar o mesmo fim, exprimem as alternativas existentes em processo. Os relacionamentos de decomposição de tarefa exprimem o que deve ser feito para se ter a tarefa realizada.

A técnica i*, através dos modelos SD e SR, fornece uma descrição, que pode ser mais detalhada a depender da necessidade, dos aspectos intencionais e de atores (itens) considerados estratégicos para a compreensão clara e completa de um processo de software.

```
Object Accounts Handler
        attributes
            s v : int; /The value to be transfered
            s acc=:    int; The accounts table is always shown to all Objects, i.e it is shared/
            l c :  int  /Represent the client/ end
        actions
            l Balances(acc=):
            s Accounts (c):
            l Credit(acc#,v): Debit(acc#,v)  end
        axioms
            [ ]  Balance(acc=) = 0;
            [Credit(acc#,v)] Balance(acc#) = Balance(acc=)+v;
The account acc# is credited with amount v.
            Balance(acc#)-v  >0 → [Debit(acc#,v)] Balance(acc#) =
                                                    Balance(acc#)-v;
Provided there is enough funds, the account acc# is  debited by amount v.
            per (Debit(acc=,v)) → Balance(acc#)-v < 2000 end
A debit  order is processed by the Account Handler  only if the resulting balance of  the customer's account dones not reach more than a
$2000 overdraft
        end Object Accounts Handler
```

Fig. 3 - Componente Accounts Handler

4.1. O SISTEMA BANCÁRIO DE TRANSFERÊNCIA DE FUNDOS

Nesta seção, utilizando-se o mesmo serviço fornecido por uma instituição bancária a seus clientes da seção 3, transferência de fundos entre duas contas, apresentaremos um estudo feito com a técnica i*, a fim de obter uma melhor compreensão dos requisitos envolvidos e assim, ser possível migrar de um modelo organizacional para um modelo funcional, onde se possa obter uma especificação mais completa e precisa dos requisitos do sistema. Será dada ênfase na modelagem das várias alternativas para o serviço de transferência e os aspectos não-funcionais. Na figura 4, pode ser visto o modelo SD do sistema de transferência de fundos convencional. Nesse modelo, o cliente (Customer) tem por objetivo transferir uma dada quantia (n), entre duas contas (a1 e a2), esperando (objetivo-soft) que o banco seja capaz de realizar essa transação com segurança (Secure [Transfer]). Para tratar da operação de transferência, existe no banco, um agente que cuida das contas (Accounts Handler). Entre suas atribuições, esse agente tem por objetivo verificar a identidade do cliente. Para atingi-lo, precisa que um outro agente, um funcionário do banco, (Teller) faça a identificação da identidade do cliente, para tanto, precisa que o cliente (Customer) esteja no banco. Do ponto de vista do cliente, este espera (objetivo-soft) que a operação de transferência seja executada com segurança.

Para melhorar o serviço prestado pelo banco é sugerida a transferência de fundos através do sistema telefônico, sendo introduzidas tarefas que visam proteger as informações que irão trafegar pela rede de comunicação pública. Na figura 5, podemos identificar as alterações propostas ao sistema convencional. Faz-se necessário um novo agente, Sistema Bancário por Telefone (Phone Banking System), que fará a verificação da identidade do cliente habilitando a transferência. Para isso, é necessário que o cliente forneça um número de identificação pessoal (PIN), que o banco espera (objetivo-soft) que seja mantido em sigilo pelo cliente (Confidential [PIN]). Por sua vez, o cliente espera (objetivo-soft) que o seu número de identificação seja mantido seguro. Com o objetivo de garantir a segurança da transação existirá duas alternativas: (i) garantir-se a segurança do PIN através de técnicas criptográficas, neste caso o banco fornece ao cliente uma chave de cifragem (Encryption Key); (ii) garantir-se a segurança por meio do fornecimento ao banco, de dados pessoais que identifique o cliente (CPF, Data de Nascimento, Filiação, etc.).

O modelo de Razões Estratégicas (SR) para o sistema bancário que utiliza o serviço telefônico pode ser visto na figura 6. Detalhando-se o agente cliente, observa-se que para satisfazer o seu *objetivo de ter fundos transferidos* é necessário a realização de uma *tarefa de solicitação* de transferência de fundos (Request Funds Transfer) , a qual fará com que o seu objetivo de ter fundos transferidos seja alcançado. Por outro lado, essa tarefa de solicitação pode ser realizada com a presença física do usuário no banco (Request Transfer at Bank) ou através do sistema telefônico

(Request Transfer by Phone). Nesta última opção, para que essa tarefa seja realizada, pode-se optar por realizar a cifragem do PIN (PIN Encryption) ou pela transferência de dados pessoais.

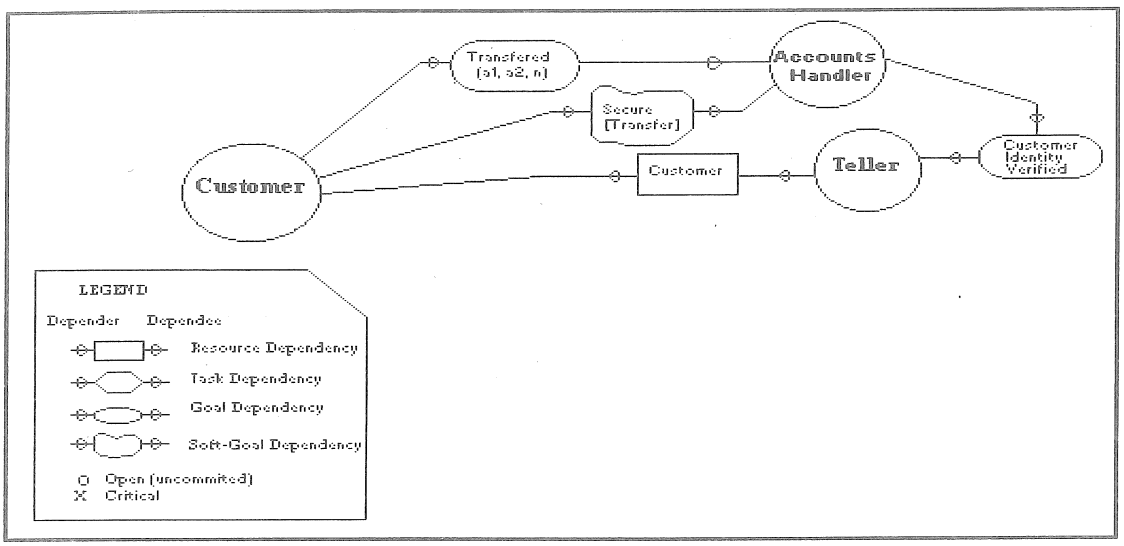

Fig. 4 - Modelo de Dependência Estratégica de Transferência de Fundos Convencional

No modelo SR, podem ser avaliadas as contribuições positivas (+) ou negativas (-) introduzidas aos objetivos-soft , relacionados a transferência, requeridas pelo cliente, tais como, segurança, conveniência, rapidez e interação amigável. No sistema convencional (Fig. 6), tem-se contribuição positiva para que sejam garantidos os objetivos de segurança e interação amigável no processo de transferência, enquanto contribuição negativa para a conveniência e rapidez, uma vez que o cliente terá que se dirigir à agência e, possivelmente, enfrentar longas filas. A introdução da transferência por telefone contribui positivamente para a rapidez e conveniência da transferência, justamente por razões opostas ao caso convencional, enquanto que contribui negativamente, para manter a interação amigável, uma vez que o cliente não contará com um atendente humano (Teller) para lidar com suas solicitações. Quanto ao aspecto de segurança, nada conclusivo se pode afirmar. Contudo, com a introdução das tarefas de cifragem e transferência de dados pessoais, este aspecto passa a ter contribuição positiva. Os objetivos-soft podem (i) estar correlacionados uns com os outros, por exemplo, a rapidez contribui positivamente para a conveniência da transferência, e (ii) servirem como critério de avaliação na introdução de novos aspectos. Do ponto de vista do sistema bancário, podemos destacar os agentes gerenciador de contas (Accounts Handler), funcionário (Teller) e o sistema por telefone (Phone Banking System). Tomando-se o gerenciador de contas, verifica-se que para realizar a tarefa de gerenciar a solicitação de transferência feita pelo cliente, faz-se necessário a identificação da identidade do mesmo, além da realização da própria tarefa de realização da transferência (Transfer). Para a realização desta última tarefa, faz-se necessário que o gerenciador tenha acesso as contas dos clientes (Accounts Ledger), a fim de verificar a existência das contas e dos saldos disponíveis que serão atualizados (essas operações não encontram-se explicitamente no modelo da figura 6). Quanto aos aspectos não-funcionais (objetivos-soft), o banco tem por preocupação aumentar sua lucratividade (Profitable [Bank]). Garantido-se o aspecto de segurança da transferência (Secure [Transfer]) e o aumento da atuação de mercado através de mais negócios e clientes (Market Sahare [RetailBanking]), ter-se-á uma contribuição positiva para o aumento da lucratividade do banco. Ao mesmo tempo, vê-se que a melhora na atuação do mercado será conseguida a medida em que a transferência se tornar mais conveniente para o cliente e para o próprio banco (Convenient [Customer,]).

A técnica i* foi apresentada informalmente, através de uma notação gráfica, de uma descrição textual e de exemplo ilustrativo, sendo possível verificar que se trata de uma técnica com capacidade para modelar requisitos necessários, tais como, requisitos não-funcionais, que não são representados com as técnicas de modelagem convencionais. Pretendemos realizar uma abordagem mais formal, a fim de se ter um sentido mais preciso para os conceitos envolvidos, de forma que seja possível obter-se uma especificação de requisitos menos ambígua do sistema, para tanto, procuraremos combinar a técnica i* com uma especificação desenvolvida em Lógica Modal de Ações.

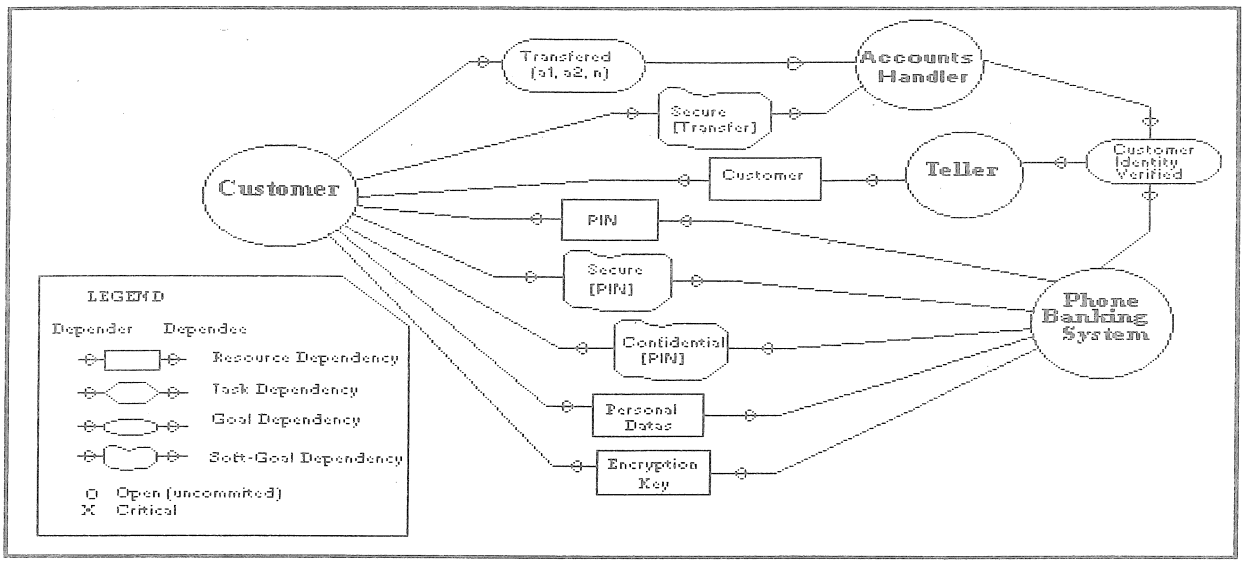

Fig. 5 - Modelo de Dependência Estratégica de Transferência de Fundos via Telefone

5. COMBINANDO A TÉCNICA i* E MAL

Nesta seção, abordaremos a questão da formalização dos requisitos expressos através da técnica i*. Mostramos como a técnica i*, trata aspectos organizacionais e não-funcionais, enquanto a MAL é usada para a especificação dos requisitos funcionais do sistema. Através do uso de ambas as técnicas, ganhamos confiança nas especificações em MAL, através da ligação de fragmentos da especificação em MAL a algum objetivo organizacional descrito nos modelos de i*. A partir de alguns objetivos organizacionais, o processo de obtenção dos requisitos (funcionais) do sistema não é tão óbvio. Tipicamente, será necessário alternar entres os modelos, pois certos aspectos descobertos em um nível de descrição precisará ser observado no outro modelo.

Parte dos conceitos empregados na técnica i* poderão ser traduzidos diretamente em MAL, por exemplo, a partir do modelo de razões estratégica SR, os agentes serão traduzidos como agentes em MAL, as tarefas serão transformadas em ações dos agentes em MAL e os recursos transformados em atributos dos agentes em MAL ou parâmetros das operações.

Na versão revisada do nosso sistema bancário, Transferência de Fundos via Telefone (Figura 5), foi introduzido o conceito do uso de senhas (PINs) e técnicas de cifragem para garantir a segurança do sistema. Será portanto necessário reavaliar a especificação original do sistema em MAL para incluir o agente (Phone Banking System), bem como adicionar novas ações para tratar do objetivo-soft de segurança (Secure[PIN]). Em particular será necessário adicionar novos axiomas ao sistema restringindo a transferência de fundos. Como há a opção do uso de senhas, somente será autorizada a transferência fundos após a conferência da identidade do cliente, através da comparação da senha (PIN) codificada ou dos dados pessoais do cliente. MAL será útil para descrever precisamente estes aspectos. Nas figuras 7 a 9, tem-se a nova especificação do sistema, com os componentes (objetos) que necessitam ser revisados e mesmo acrescentados, como o **PhoneBankingSystem** (Sistema por Telefone). O novo sistema **Banking System** (Fig. 7) inclui um novo objeto, **PhoneBankingSystem** (Vide Fig. 9) e novos axiomas que dizem respeito à opção do cliente de solicitar uma transferência através de um funcionário (Teller) ou por telefone (PhoneBankingSystem). Em ambas as situações, uma solicitação de transferência será sempre seguida de ações de débito e crédito no objeto que gerencia as contas do banco (**Accounts Handler**). Na figura 8, tem-se o novo objeto cliente (**Customer**) com um novo atributo, o PIN (senha codificada) e com uma nova ação de solicitação de transferência (TransferPhoneR), onde o PIN será passado como parâmetro, caso o cliente opte pela transferência por telefone. Ao mesmo tempo, será dado ao cliente sempre as opções de transferência através de um funcionário ou através do telefone. Essas ações de transferência do cliente são compartilhadas, permitindo que outros componentes as utilizem. O objeto **PhoneBankingSystem** fará a identificação do cliente através da ação de autorização do cliente

com o seu respectivo PIN (Check_PIN(c,PIN)). Só será permitida a operação de transferência (per(TransferPhone(PIN,acc1#,acc2#,v))) se houver uma solicitação do cliente e sua senha for validada.

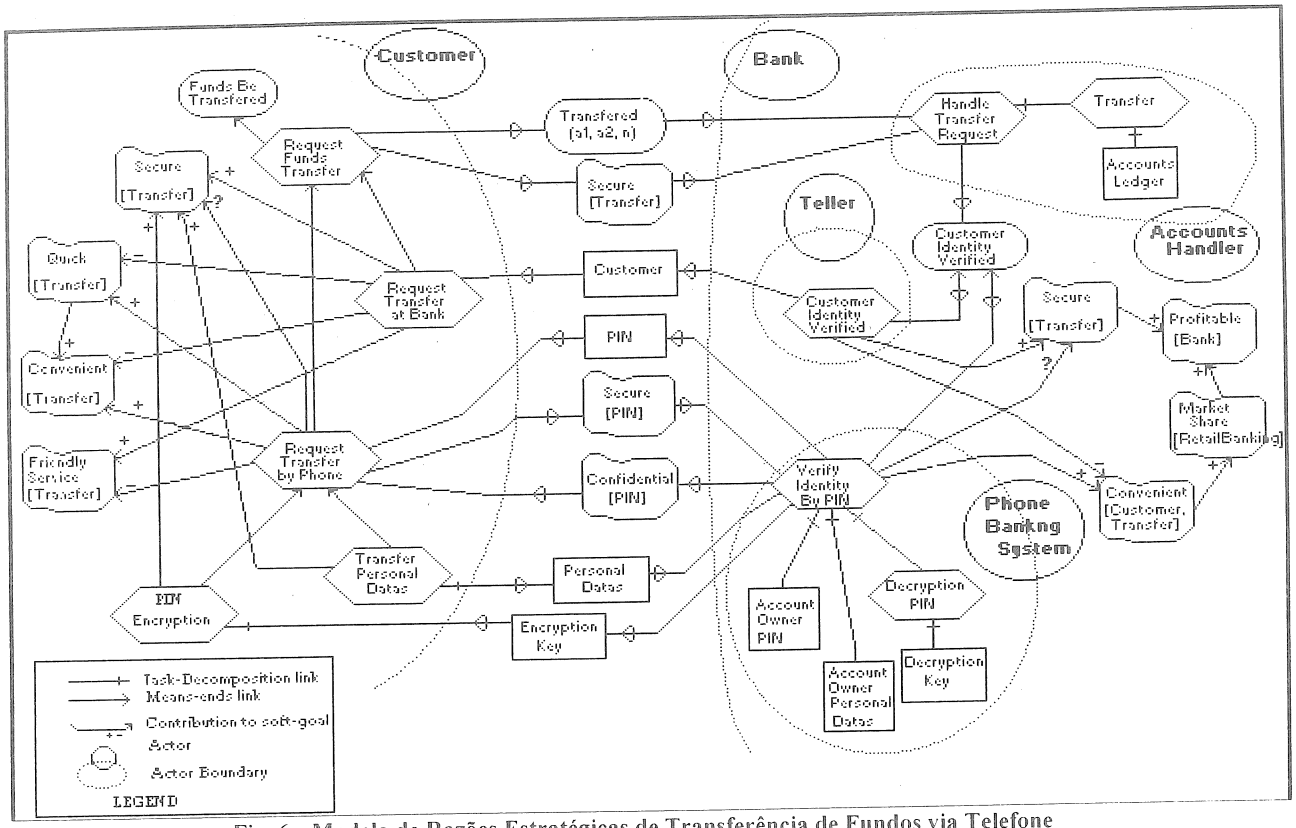

Fig. 6 - Modelo de Razões Estratégicas de Transferência de Fundos via Telefone

A nível de i*, é possível identificar correlações entre outros objetivos que surgem da necessidade de se manter a senha (PIN) confidencial. Por exemplo, mante-la confidencial, contribui negativamente para conveniência pois um cliente pode esquecer sua senha. Requisitos adicionais, também relacionados a propriedades do código da senha (PIN), podem ser identificados. Por exemplo, com o uso de técnicas de cifragem, será mais difícil um usuário não autorizado obter acesso ao sistema. As propriedades de segurança do algoritmo de cifragem poderiam também ser especificadas em MAL. Alternativamente, há a possibilidade do uso de dados pessoais para identificação do cliente. Neste caso MAL poderia ser usado para especificar a estrutura e tipo destes dados. As formas de garantia da segurança estão relacionadas com a operação de transferência, a qual, por questão de espaço, não se encontra totalmente especificada em MAL.

O que se tem de mais importante é que durante o processo de requisitos, o analista precisará interagir navegando entre os modelos da técnica i* (SD e SR) e a especificação em MAL, de forma a poder lidar com impactos em cada um deles. Ao nível de compreensão dos relacionamenos organizacionais, faz-se necessária uma noção de agente que reconheça que os agentes têm liberdade e podem violar restrições e compromissos. Precisa-se raciocinar sobre as implicações dessas violações. Os modelos são usados descritivamente para entender as condições organizacionais como são ou podem ser, no caso de configurações propostas. Precisa-se tomar uma visão estratégica dos relacionamentos dos agente porque novas disposições de trabalho alteram a configuração das dependências. A introdução de novos sistemas e/ou processos de trabalho mudam o que é possível ou não é possível, ou mudam o grau de dificuldade em alcançar os objetivos. Modelos a esse nível tendem a ser muito incompletos, mas isso é apropriado desde que apenas aspectos que são de significância estratégica necessitam ser considerados. A nível de especificação dos requisitos, uma visão prescritiva é mais apropriada que uma visão descritiva. Analistas desejam ser capazes de confirmar que uma configuração organizacional tem certas propriedades desejadas, assumindo que agentes são fiéis às restrições fixadas em seus comportamentos (as obrigações) em forma declarativa. O nível de especificação

tipicamente requer um mais alto grau de completude, a fim de ser capaz de garantir certas propriedades. Assim, conceitos de modelagem mais refinados são apropriados, tais como, estados, ações , obrigações, informação, percepção e restrições em tempo real.

```
Object Banking System
  .....
 includes
        Customer,
        Accounts Handler;
        Teller,
        Phone Banking System
 end
 axioms
   Customer.TransferR(a1,a2,n) →Teller.Transfer(a1,a2,n) :
A transfer request from the Customer is echoed by a transfer order sent to the Teller
   Customer.TransferPhoneR(PIN,a1,a2,n)→
                                        PhoneBankingSystem.TransferPhone(PIN,a1,a2,n);
A transfer request over the phone  from the Customer is echoed by a transfer order sent to the PhoneBankingSystem/
   Teller.Transfer(a1,a2,n) V PhoneBankingSystem.TransferPhone(PIN,a1,a2,n) →
                                  Accounts Handler.Debit(a1,n) *Accounts Handler.Credit(a2,n) end
A transfer order should be followed by the corresponding credit and debit operation
end Object Banking System
```

Fig. 7 - Sistema de Transferência Bancária com opção via Telefone

```
Object Customer
 attributes
        s Local_request(acc1=): bool,
        s Phone_request(acc1=) : bool,
        l v : int;
        s PIN: int;
A Personal Indentification Number that will be passed as key to access the banking system
        acc1#; acc2=; :    int  end
 actions
        s TransferR(acc1=, acc2#, v);
        s TransferPhoneR(PIN, acc1#, acc2#, v) end
 axioms
        [TransferPhoneR(PIN, acc1#, acc2#, v)] Phone_request(acc1#)
        [TransferR(acc1=, acc2#, v)] Local_request(acc1=)
        per(TransferR(acc1#, acc2#, v));
In any situation, the customer is permitted to send his or her transfer/
        per(TransferPhoneR(PIN, acc1#, acc2#, v)) end
In any situation, the customer is permitted to send his or her transfer over the phone, that will be validated with his/her PIN code
 end Object Customer
```

Fig. 8 - Componente Customer

```
Object PhoneBankingSystem
 attributes
        l v : int; Value to be transfered
        s c : int; Represent the client/
        s PIN: int;   The Client's Personal Indentification Number
        s acc1#: acc2=:      int
        PIN_ok (c,PIN): bool      end
 actions
        l Check_PIN(acc1#)
        s TransferPhone (PIN, acc1#, acc2#, v):   end
 axioms
        [Check_PIN(acc1=) Pin_ok(acc1#) ∨ ¬PIN_ok(acc1=)
        per(TransferPhone(PIN, acc1#,acc2#,v)) → Phone_request(acc1#) ∧ PIN_ok (c,PIN)
        Phone_request(acc1=) → obl [ Check_PIN(acc1=)]   end
 end Object PhoneBankingSystem
```

Fig. 9 - Componente PhoneBankingSystem

6. CONSIDERAÇÕES FINAIS

Na área de modelagem de requisitos é bem reconhecida a necessidade de modelar o ambiente. Modelos de organizações e empresas têm sido desenvolvidos nas áreas de computação organizacional e integração de empresas.

Muitas ferramentas e técnicas da engenharia de requisitos têm sido desenvolvidas para se refinar os requisitos, a fim de se alcançar melhor precisão, completude e consistência. Contudo, para desenvolver sistemas que verdadeiramente traduzam as necessidades reais de uma organização, faz-se necessário se ter o conhecimento aprofundado de como o sistema é encaixado no ambiente organizacional, ou seja, devem ser considerados aspectos intencionais e estratégicos dos agentes do sistema.

Neste trabalho verificamos que os modelos da técnica i* (modelos descritivos) e o modelo prescritivo da técnica MAL se complementam na obtenção de uma descrição mais completa. Com a integração das duas técnicas obtivemos uma especificação mais precisa dos requisitos, em termos de lógica de ações. Os benefícios são muitos pois a linguagem formal permite a geração de animações da especificação, que podem ser usadas para validação dos requisito e, adicionalmente, todo o potencial de raciocínio formal que a linguagem permite.

Algumas ferramentas estão disponíveis para uso das técnicas mencionadas. Para modelagem estratégica de i* existe o OME, enquanto para o MAL, a plataforma MULTIVIEW permite a geração de especificações em MAL, através do Método VSCS. Trabalhos futuros permitiram a integração destas ferramentas dentro de um único ambiente. Para uma melhor avaliação da potencialidade das técnicas, estudos de caso com parceiro industrial se faz necessário. No momento estamos experimentando o uso da técnica MAL na definição da funcionalidade de um central telefônica. Em seguida será feita a modelagem organizacional do sistema, visando o refinamento da especificação formal.

7 - REFERÊNCIAS

[1] Berry, Daniel. "The Requirement Iceberg and Various Icepicks Chipping at it". VIII Simpósio Brasileiro de Engenharia de Software, Curitiba - PR, 25-28 Out., 1994.

[2] Eric S. K. Yu, "Modelling Strategic Relationships for Process Reengineering". PhD thesis, Computer Science Department, University of Toronto, Toronto (Canada), 1995. Also appears as Technical Report DKBS-TR-94-6, December 1994.

[3] M. D. Ryan, J. Fiadeiro, and T. Maibaum. "Sharing Actions an Attributes in Modal Action Logic". T. Ito and A. Meyer, editors, Theoretical Aspects of Computer Software. Springer Verlag, 1991.

[4] S. Kent, T. Maibaum, and W. Quirk. "Formally Specifying Temporal Constraints and Error Recoverey". Proceedings of IEEE International Symposium on Requirements Engineering - RE93, p. 208-215, Jan. 1993.

[5] Eric Yu, Philippe Du Bois, Eric Dubois, John Mylopoulos, "From Organization Models to System Requirements - A Cooperating Agents Approach". The 3rd International Conference on Cooperative Information Systems - CoopIS-95, Vienna (Austria), May, 9-12, 1995.

[6] E. Dubois, P. Du Bois and M. Petit. "ALBERT: An Agent-Oriented Language for building and eliciting requirements for real-time systems". In J. F. Numacker and R. H. Sprague, editors. Proc. of the 2th HICSS, vol. 4, p. 713-722. IEEE Computer Society Press, Jan. 1994.

[7] Janis A. Bubenko. "On Concepts and Strategies for Requirements and Information Analysis". In Information Modeling, p. 125-169. Chartwell-Brant, 1993.

[8] A. Dardenne, A. van Lamsweerde, and S. Fickas. "Goal-directed Requirements Acquisition". Science of Computer Programming, 20:3-50, 1993.

[9] Philippe Du Bois, Eric Dubois, Jean-Marc Zeippen. " On the Use of a Formal RE Language: the Generalized Railroad Crossing". Proceedings of the 3rd International Requirements Engineering Symposium RE-97, Annapollis, pp.128-137, USA, January, 1997.

[10] Eric S. Yu. "Towards Modelling and Reasoning Support for Early - Phase Requirements Engineering". Third IEEE International Symposium on Requirements Engineering - RE'97 - Annapolis, USA , Jan., 1997.