

# Uma Abordagem Teleológica para Construção de Casos de Uso e Cenários

**Roberto Vedoato<sup>1</sup>**  
roberto.vedoato@uol.com.br

e

**Marcelo Soares Pimenta<sup>1</sup>**  
mpimenta@inf.ufrgs.br

<sup>1</sup>Pós-Graduação em Computação, Instituto de Informática, UFRGS  
Caixa Postal 15.064  
91501-970 – Porto Alegre – RS Brasil

## Abstract

Interactive systems must be useful and usable. For that it is necessary to consider jointly the different perspectives of Software Engineering (SE) and of Human Computer Interaction (HCI) from the early phases of development. Scenarios and use cases are important modeling techniques used in HCI and SE that allow integrating the perspectives of both areas. However, use cases present conceptual problems and there are not ripe methodologies to build them systematically. This work investigates and proposes a process of systematic construction of use cases with different abstraction levels, based on the teleological paradigm, in order to obtain a use case model that specify functional and interaction requirements in a comprehensible and practicable fashion.

**Keywords:** Use Cases, Scenarios, Goals, Interactive Systems, Human Computer Interaction.

## Resumo

Para que sistemas interativos sejam úteis e usáveis é necessário que desde as primeiras fases de desenvolvimento as diferentes perspectivas das áreas de Engenharia de Software (ES) e de Interação Humano-Computador (IHC) sejam consideradas simultaneamente. Cenários e casos de uso são importantes técnicas de modelagem utilizadas em IHC e ES que permitem integrar as perspectivas de ambas as áreas. Contudo, casos de uso apresentam problemas conceituais e faltam metodologias maduras para construí-los sistematicamente. Este trabalho investiga e propõe um processo de construção sistemática de casos de uso com diferentes níveis de abstração, baseado no paradigma teleológico, a fim de se obter um modelo de casos de uso que especifique requisitos funcionais e de interação, de maneira compreensível e praticável.

**Palavras Chaves:** Casos de uso, Cenários, Objetivos, Sistemas Interativos, Interação Humano-Computador.

## 1 Introdução

Freqüentemente sistemas interativos são desenvolvidos com visões isoladas das áreas de Engenharia de Software (ES) e de Interação Humano-Computador (IHC) [20]. ES tradicionalmente prioriza aspectos essencialmente funcionais dos sistemas, como eficiência, manutenibilidade e portabilidade, conferindo ao desenvolvimento uma orientação funcional em detrimento da operacional. Já IHC foca principalmente na usabilidade dos sistemas, concentrando atenção nos aspectos de interação e não considerando adequadamente os aspectos enfatizados pela ES [24]. Essas duas abordagens refletem diferentes perspectivas, uma mais tecnológica e outra mais humana, sobre a mesma atividade de desenvolvimento, sendo utilizados diferentes conceitos, diferentes metodologias, etc.

Recentes trabalhos de pesquisa convergem para idéia central de que para desenvolver sistemas interativos úteis e usáveis é necessário que desde as primeiras fases do desenvolvimento do sistema sejam consideradas as perspectivas de ES e IHC, exigindo a utilização conjunta e integrada de conceitos, técnicas e metodologias de desenvolvimento de ambas as áreas. Porém, essa interseção não é ainda nem natural nem objetiva, cada área considera aspectos diferentes e muitas vezes disjuntos do sistema sem nenhuma correspondência explícita e sistematicamente estabelecida, tendo como consequência o fracionamento de requisitos [21].

Cenários e casos de uso (*use cases*) são importantes técnicas de modelagem utilizadas em IHC e ES, em diferentes contextos, com diferentes visões, mas apresentando muitas similaridades. Cenários são comumente utilizados em IHC para inspecionar atividades humanas ao usar um artefato (presente ou futuro), enquanto que casos de uso são usualmente utilizados em ES para modelar requisitos funcionais. Estamos convictos que são conceitos valiosos para integrar as diferentes perspectivas das duas áreas citadas. Partindo do enfoque que cenários são instâncias concretas de casos de uso, podemos combinar as aplicações e os propósitos de ambas as técnicas, utilizando o conceito de níveis de abstração, o que possibilita uma busca integrada da qualidade interna e externa dos sistemas.

Contudo, abordagens apontam que casos de uso apresentam problemas conceituais e a maneira pela qual são comumente usados na *Unified Modeling Language* (UML) [13], não considera adequadamente aspectos de usabilidade. A abordagem original de casos de uso [15], posteriormente adotada na UML, não define precisamente nenhum formato específico para descrever seus conteúdos, nem um processo sistemático para construí-los. Desenvolvedores freqüentemente não sabem como identificar casos de uso, o que incluir neles, qual o nível de detalhamento, como representá-los e como estruturá-los. Várias pesquisas têm focado principalmente nas funções e nos usos de cenários e casos de uso no processo de desenvolvimento de sistemas, enquanto o processo de construção destes muitas vezes é negligenciado. Trabalhos de pesquisas apontam objetivos como chave para a construção de cenários e/ou casos de uso [1,7,14,18], todavia falta um processo maduro de construção e validação sistemática de casos de uso a partir dos objetivos.

Neste contexto, este trabalho tem como objetivo investigar e propor uma abordagem teleológica - baseada em objetivos - para construção sistemática de casos de uso com diferentes níveis de abstração, a fim de se obter um modelo de casos de uso que especifique requisitos funcionais e de interação, de maneira compreensível e praticável. Na próxima seção são vistos alguns aspectos relevantes sobre casos de uso e cenários, em seguida alguns conceitos e considerações sobre abordagens teleológicas são mostrados na seção 3. A seção 4 apresenta nossa proposta e finalmente na seção 5, considerações finais são apresentadas.

## 2 Casos de Uso e Cenários

Existem ainda várias discussões ao redor de cenários e casos de uso. Os termos “cenário” e “caso de uso” significam diferentes coisas para diferentes pessoas: a definição de seus conceitos muitas vezes não é clara, ora são utilizados como sinônimos ora são usados distintamente. São importantes técnicas de modelagem utilizadas em IHC e ES, em diferentes contextos, com diferentes visões, mas apresentando muitas similaridades. Cenários são comumente utilizados em IHC para inspecionar atividades humanas ao usar um futuro artefato, concentram-se na usabilidade dos sistemas [16]. Ao passo que casos de uso são usualmente utilizados em ES para manipular modelos de objetos e para modelar requisitos funcionais [15].

O conceito original de casos de uso proposto por Jacobson [15], posteriormente adotado na UML é confuso, existem ainda aspectos problemáticos que precisam ser tratados. Alguns deles são destacados em [6,19] e listados a seguir:

- Estrutura e conteúdo: O que exatamente casos de uso contém? Como são organizados os conteúdos?
- Expressões idiomáticas: Qual linguagem é usada para expressar os conteúdos que definem um caso de uso?
- Granularidade: O tamanho e o nível de detalhamento de cada caso de uso é escolhido arbitrariamente?
- Cobertura: Casos de uso podem garantir apenas uma cobertura parcial de todos possíveis usos do sistema?
- Contexto: Casos de uso ocorrem sobre quaisquer situações ou sobre condições específicas?

- Interseção: Casos de uso são independentes ou podem sobrepor-se totalmente ou parcialmente?
- Concorrência: As concorrências internas e entre casos de uso não são modeladas?

Outro problema pertinente é que na UML casos de uso são geralmente referentes à interação sob a ótica do sistema, não considerando adequadamente aspectos de usabilidade.

Nosso entendimento de cenários e casos de uso é uma combinação das abordagens de Pimenta [20], Cockburn [1], Potts [9] e Carrol [16], considerando também idéias presentes em demais trabalhos. Para nós, caso de uso é considerado como um modelo narrativo de um conjunto de seqüências de interações, inclusive variantes, entre um ator e o sistema, agrupadas por um objetivo comum do ator, representando potenciais ou reais situações de uso do sistema, onde o objetivo é alcançado ou não. Ao passo que um cenário é uma instância específica e representativa de um caso de uso tendo um resultado particular com respeito ao objetivo do ator, corresponde a descrição de um comportamento do ambiente e do sistema que aparece em uma situação concreta e restrita de uso. Um ator representa um papel específico que agentes externos ao sistema (usuários, dispositivos de hardware, outros sistemas) desempenham com o software. Ao longo deste trabalho utilizaremos os termos “cenário” e “caso de uso” de acordo o entendimento acima explicado, embora sejam utilizadas terminologias diferentes em algumas das abordagens referenciadas.

Há uma grande variedade de estilos para descrever o conteúdo narrativo de um caso de uso [4, 8, 18]. Originalmente casos de uso não possuem nenhum formato específico para descrever seus conteúdos, sendo representados na forma de narrativa textual contínua. Os problemas deste estilo de narrativa são numerosos. A narrativa mistura exigências internas e externas e salta irregularmente entre perspectivas internas e externas. Os elementos essenciais à natureza do problema são misturados com decisões de projeto e a falta de estrutura força o leitor a seguir o texto inteiro apenas para obter uma idéia geral do caso de uso [18]. A abordagem original de casos de uso também não especifica nenhum processo de construção sistemática, nela a criação destes depende de pelo menos uma compreensão parcial dos requisitos do sistema, idealmente expressas num documento de especificação de requisitos. Algumas heurísticas e questões para responder para cada ator ajudam a identificar casos de uso.

### 3 Abordagens Teleológicas

Ao criarmos um software não planejamos e desenvolvemos apenas um artefato, criamos novas possibilidades para ações e interações humanas [16]. Todo software é uma ferramenta. Como boa ferramenta deve suportar o trabalho de alguém, tornar o trabalho mais fácil, mais rápido, mais simples e mais flexível. Para desenvolvermos sistemas que suportam adequadamente o uso, trabalhos de pesquisa enfatizam que precisamos entender melhor as tarefas realizadas pelas pessoas e aplicar de maneira mais eficiente nossa compreensão das tarefas no processo de desenvolvimento de software. Seguindo esta perspectiva, está claro que para realmente definirmos os requisitos de um sistema interativo é necessário em primeiro lugar conhecermos os objetivos do usuário para então propor a especificação dos mecanismos que sustentaram as tarefas necessárias para alcançar os objetivos [20].

Enquanto análises de sistema tradicionais focam em “quais” características um sistema irá suportar, abordagens baseadas em objetivos focam em “por que” sistemas são construídos provendo motivação e argumento para justificar os requisitos de software [3]. Focar em objetivos, ao invés de requisitos específicos, permite analistas comunicarem com *stakeholders* usando uma linguagem baseada em conceitos que ambos têm familiaridade [3].

Especificar os requisitos de um sistema a partir dos objetivos dos usuários não é uma tarefa trivial. Objetivos são condições desejáveis de serem alcançadas, não ações a serem realizadas [3]. O refinamento de objetivos em soluções de projeto é um processo de múltiplos estágios, vagos objetivos de alto nível devem ser refinados em concretos objetivos formais [12]. Isto é necessário, pois somente objetivos primitivos podem ser operacionalizados - traduzidos em ações atômicas do usuário, do hardware ou do sistema – para então se tornarem requisitos operacionais na especificação final de requisitos [9].

Um dos fatores positivos de enfatizarmos objetivos como fontes para o desenvolvimento é que eles são consideravelmente estáveis [3]. Vários sistemas podem ser propostos para realizar os objetivos dos usuários, eles diferirão basicamente em como os objetivos são operacionalizados, ou seja, quais ações serão executadas para realizar os objetivos e quem ou o que as realizarão. Isto implica que é possível, a partir dos objetivos, especificar as futuras tarefas com o sistema (as tarefas interativas) [20].

Devemos considerar que a operacionalização dos objetivos não deve ser predominantemente *top-down*, pois é altamente interativa [12]. Derivação de casos de uso e refinamento de objetivos são atividades do desenvolvimento de software que se apóiam mutuamente [9]. O conhecimento de teleológico preocupa-se com os propósitos específicos para os quais o sistema foi projetado, enquanto a análise de casos de uso é dedicada a preencher a lacuna entre tais propósitos abstratos e a real estrutura e comportamento do sistema [12]. Assim, é possível obter um conjunto de casos de uso analisando objetivos, alocação de objetivos, singularidades, etc. Reciprocamente, é possível retornar e refinar a análise de objetivos quando caminhamos através de um caso de uso, já que a análise do

caso de uso pode fornecer percepções concretas sobre o comportamento do macrosistema e as razões para tal [9]. Casos de uso clarificam o pensamento do projeto de várias maneiras sendo possível comparar alternativas de automação e avaliar a consequência de singularidades e possíveis mecanismos para defender ou suavizar seus efeitos [9]. Também permitem estabelecer proposições de novas ações para alcançar um determinado objetivo e descobrir novos objetivos para a nova situação futura [20].

Através da abordagem Kaindl [14] é possível compreender a importância dos objetivos na construção de casos de uso. Somente podemos entender as interações descritas em um caso de uso quando conhecemos seus objetivos. Quando estamos familiarizados com um sistema os objetivos das interações parecem óbvios, mas no processo de elicitação de requisitos e modelagem de tarefas de um novo sistema, ainda desconhecido, essas informações são cruciais para uma melhor compreensão das interações como um todo [14]. Faltam representações adequadas dos objetivos nos casos de uso.

Na realidade, demais trabalhos de pesquisa já consideram que objetivos são fundamentais para a construção de cenários e/ou casos de uso [1,7,9,12,14,18], porém poucos [11,23] oferecem uma maneira de utilizarmos complementarmente as duas técnicas para a determinação dos requisitos. Na maioria das abordagens ou não existe uma notação para objetivos, ou não existe um processo sistemático de construção e validação de casos de uso a partir dos objetivos ou não existem ambos.

## 4 Uma proposta teleológica

### 4.1 Introdução

Nosso trabalho investiga e propõe uma abordagem que faz uso de conceitos teleológicos para gerar, estruturar e validar casos de uso com diferentes níveis de abstração. Temos como base o modelo de casos de uso proposto originalmente por TAREFA (*Task Analysis based Requirements Engineering Framework*) [20], uma abordagem concebida para sistematizar a engenharia de requisitos de sistemas interativos. Nesta proposta os aspectos construtivos de casos de uso propostos em TAREFA [20] são aprimorados com idéias de outras pesquisas relacionadas juntamente com algumas novas considerações julgadas pertinentes.

A proposta tem como principal objetivo sistematizar a construção de casos de uso e cenários, propondo um método que a partir de um modelo explícito de objetivos do usuário, refinados e representados, permite definir, refinar e representar os objetivos do sistema que servirão como base para a construção de casos de uso com diferentes níveis de abstração. O método é possivelmente um processo iterativo de construção de casos de uso num nível mais abstrato, de refinamento e de instanciação de casos de uso (cenários), pelo analista, de definição e investigação de requisitos a partir de experimentação e avaliação da funcionalidade e do comportamento dos cenários envolvendo a participação de usuários. Isto é, casos de uso serão formalizados, refinados e verificados iterativamente.

Acreditamos que a combinação de casos de uso e cenários com uma boa compreensão das tarefas e do contexto, juntamente com a explicitação dos objetivos, previnem a possibilidade de que especificações tornem-se desconectadas das reais necessidades dos usuários e devem ser a base para um processo de desenvolvimento de software interativo.

O diagrama apresentado na figura 1 representa uma visão geral do processo de construção sistemática de casos de uso e cenários. Ao final do processo, deseja-se obter um modelo de casos de uso que especifique requisitos funcionais, requisitos de interação e requisitos não funcionais, de maneira compreensível e praticável, que sirva como ponto de partida para a continuidade do desenvolvimento de software.

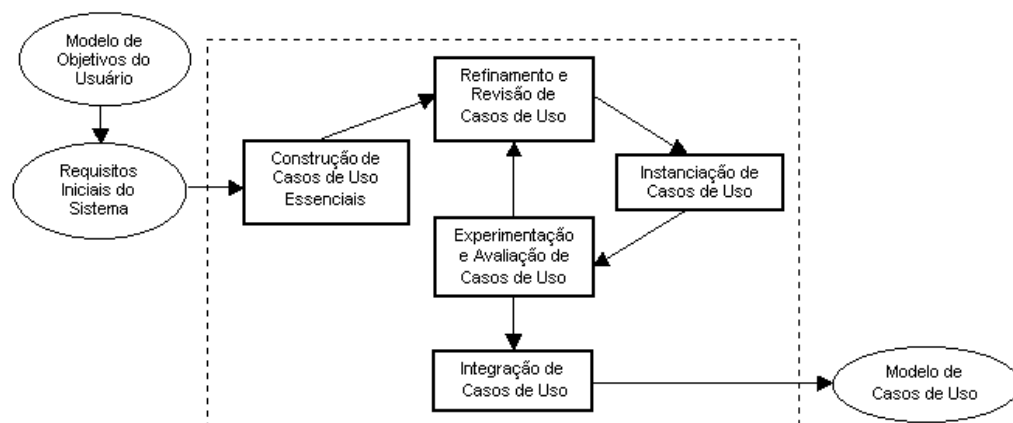


Figura 1 – Visão geral da abordagem.

Lembramos que algumas atividades são necessárias antes que os casos de uso possam ser produzidos. Levando em consideração o que foi apresentado na seção 3, fica fácil compreendermos que para considerarmos adequadamente aspectos de usabilidade, os objetivos dos usuários são fundamentais. Devemos então, ter como ponto de partida em nosso processo de construção de casos de uso um modelo explícito de objetivos do usuário, refinados e representados.

Uma contribuição original de TAREFA [20] é que a hierarquia de objetivos dos usuários é obtida e representada através de um tipo particular de modelo de tarefa. A análise de tarefa emergiu da ergonomia como um importante apoio ao desenvolvimento de sistemas interativos devido ao fato de ser um método empírico de compreender como as pessoas executam suas tarefas. Uma análise de tarefa produz um modelo explícito de tarefas em um domínio, denominado modelo de tarefas, representando dois tipos de informação: objetivos e ações. Em [20] é proposto um modelo, derivado diretamente de um modelo de tarefas, onde o elemento principal é a estrutura dos objetivos e sub-objetivos, denominado modelo de tarefas minimal. O modelo de tarefas minimal é um modelo abstrato sem considerações tecnológicas, representando as intenções do usuário, ao invés de ações e serve como base para a construção dos casos de uso. A notação utilizada neste modelo é MAD (*Méthode Analytique de Description*) [20].

Entretanto, a determinação dos requisitos dos usuários não é suficiente para a definição dos requisitos do sistema. Também em TAREFA [20] são propostas duas atividades para a definição dos requisitos iniciais do sistema, chamados ‘iniciais’ pois só representam o ponto de vista do usuário sobre os requisitos do sistema. A primeira é uma Re-engenharia de tarefas com o intuito de reorganizar as tarefas que se deseja corrigir para eliminar as redundâncias e as ineficiências, antes de propor uma solução computacional. E a segunda é uma Alocação de Funções, para determinar que tarefas serão manuais, automáticas ou interativas. Após a execução desses dois passos os requisitos iniciais do sistema são determinados e podemos pelas atividades de construção, de experimentação, de modificação e de integração dos casos de uso especificar as características que um sistema tem que possuir para satisfazer os requisitos dos usuários.

Segundo Leite [17], o desenvolvimento de software baseado em casos de uso apóia-se no conceito de que a utilização da linguagem do problema é benéfica na interação entre usuários e desenvolvedores. Para evitarmos alguns problemas das linguagens naturais defendemos o uso de terminologia unificada. A unificação terminológica é especialmente importante, porque diferentes casos de uso podem ser descritos e analisados por pessoas ou grupos diferentes, e ela auxilia a estabelecer um entendimento comum entre os *stakeholders*. Em adição, a ontologia pode ajudar a encontrar e prevenir redundâncias, um problema que geralmente aparece conforme aumenta o número de casos de uso. Podemos representar o modelo ontológico através da notação LEL (*Language Extended Lexicon*) [17]. A ontologia já deve ser previamente gerada ao determinarmos os objetivos dos usuários, mas é gradualmente estendida e revisada conforme os casos de uso são identificados e inspecionados.

#### 4.2 Modelo de Casos de Uso

Considerando o que foi já foi exposto, é fácil compreender que nosso conceito de casos de uso está ancorado no paradigma teleológico. Casos de uso devem ser vistos como planos de execução de objetivos ao invés de meras seqüências cronológicas de eventos. Como já observado em [1] uma das dificuldades desta perspectiva é o fato de objetivos existirem em vários níveis (objetivos, sub-objetivos, ações individuais) e a operacionalização requer uma freqüente mudança entre os níveis. Outra característica relevante observada em [14,20] é que um caso de uso é ao mesmo tempo uma especificação da estrutura, da funcionalidade e do comportamento de uma situação de uso. Como toda a especificação, tem dois aspectos: o aspecto técnico, pois age como ponto de origem para a concepção e a implementação do sistema, e o aspecto social, porque serve para comunicação entre os *stakeholders*, em particular o analista e o usuário.

Para lidar com esses aspectos, utilizamos o conceito de níveis de abstração. Abstração é definida como um mecanismo para esconder detalhes a fim de focar em aspectos essenciais e o refinamento é utilizado para descrever gradualmente os casos de uso em diferentes níveis de abstração [11]. Adotamos em nossa abordagem quatro níveis de abstração propostos em TAREFA [20], para descrevermos os casos de uso, cada um adaptado a um determinado objetivo e relativo a um tipo específico de conhecimento e representação. Os quatro níveis definidos são:

- Essencial: Casos de uso no nível mais abstrato. São construídos a partir dos objetivos de tarefas, presentes no modelo de tarefas minimal. Devem ser realizados para a satisfação dos objetivos das atividades dos usuários, sempre consideradas num contexto organizacional;
- Singular: Casos de uso essenciais refinados através da determinação de singularidades (desvios do curso “normal” que incluem problemas, disfuncionamentos, exceções, interrupções e obstáculos). Para cada singularidade pode haver a identificação de possíveis ações defensivas e corretivas que o sistema poderia realizar para respectivamente evitar ou corrigir um problema. Portanto, um caso de uso essencial pode dar origem a diversos casos de uso singulares, um para cada ocorrência de uma singularidade;
- Operacional: Um caso de uso operacional descreve o comportamento de um caso de uso singular em termos de

objetos clientes, objetos servidores, atributos, serviços e estados, num nível similar ao da Análise Orientada a Objeto. Estes casos de uso estão num nível de abstração abaixo dos casos de uso singulares e são voltados especificamente para a geração preliminar de arquiteturas de sistemas orientados a objeto.

- Concreto: Casos de uso no menor nível de abstração. Um caso de uso concreto é uma ocorrência de uma execução particular de um caso de uso singular do ponto de vista do usuário. Ele corresponde a uma descrição das características de uma situação concreta de uso (protótipos das IUs, comportamento detalhado do usuário e do sistema, condições de uso, singularidades ocorridas) sem levar em consideração os detalhes técnicos e arquitetônicos do sistema no nível operacional. Por definição, corresponde ao conceito de cenário utilizado em IHC.

Não é necessário fazer a descrição completa de um caso de uso de acordo com os quatro níveis. Por exemplo, não há necessidade de se fazer um refinamento passo a passo de um caso de uso essencial até o concreto: o caso de uso operacional é opcional e seu uso só é interessante para a concepção orientada a objeto [20]. Devido ao limite de páginas não detalharemos neste artigo o nível operacional como faremos a seguir com os demais níveis. Existe uma grande variedade de estilos de representação utilizados para descrever o conteúdo narrativo de um caso de uso (maiores detalhes em [4,18]). Mas a notação nas quais os casos de uso são escritos tem um profundo efeito em suas utilidades e na qualidade dos artefatos que resultam [18]. Para cada um dos níveis de abstração adotamos uma notação específica de acordo com o propósito do nível.

#### 4.2.1 Casos de Uso Essenciais

##### 4.2.1.1 Introdução

Nosso conceito de caso de uso essencial [20] é similar ao conceito de caso de uso essencial de Constantine [18], diferindo basicamente no fato de que os casos de uso essenciais de Constantine [18] são casos de uso expandidos, que descrevem além do curso “normal”, as alternativas importantes ou exceções que podem surgir em relação ao curso “normal”, enquanto que nossos casos de uso essenciais são de alto nível, descrevendo apenas o curso “normal” das atividades, onde o objetivo é alcançado sem nenhuma obstrução. Consideramos que é benéfico o uso de casos de uso essenciais como casos de alto nível, pois desta maneira tornam-se úteis para compreendermos rapidamente o grau de complexidade e a funcionalidade do sistema proposto. Concentramo-nos primeiro em como o sistema deve funcionar normalmente antes de investigarmos as possíveis alternativas e falhas.

Segundo Constantine [19], a chave para a usabilidade é mudarmos o foco dos casos de uso de interação para intenção e de elaboração para simplificação. Casos de uso não devem ser entendidos como *logs* de atividades humanas, ao contrário, devem ser abstratos e orientados a objetivos, significativos e discutíveis pelos usuários.

Os casos de uso essenciais são expressos numa forma ideal, relativamente livre de tecnologia e de detalhes de implementação, ou seja, as decisões de projeto são postergadas e abstraídas, principalmente aquelas relacionadas com a interface com o usuário (IU) [8]. Eles são abstratos e modelam as intenções e não as ações dos usuários, as responsabilidades e não as respostas do sistema [18]. Um caso de uso essencial não especifica uma seqüência de interações concretas, mas uma seqüência de passos abstratos, permitindo um variável número de implementações [22] e encorajando a inovação criativa [18]. Além disso, como são adaptados ao nível de abstração do usuário, eles podem servir como um eficiente meio de comunicação entre o usuário e analista [20].

Existe uma sutil diferença entre objetivos e intenções. Objetivo é um estado final desejado de um sistema, sendo corretamente descrito em termos estáticos como estado e características de objetos. Já uma intenção, em contraste, é dinâmica e representa o sentido ou o progresso ao invés de um estado final. “*Objetivos são destinos, enquanto que as intenções representam a jornada do usuário*” [18]. Ainda sobre este aspecto, Kavakli [12] destaca que “necessidades” são difíceis de serem averiguadas, é mais apropriado considerarmos requisitos como “tendências”, – “intenções” para nós – aquilo que pessoas tentam fazer quando tem oportunidade [12]. Uma intenção deve ser entendida como uma versão operacional de uma necessidade, portanto ela pode ser identificada e testada observando-se o comportamento das pessoas [12]. Neste sentido, casos de uso essenciais podem definir estratégias sistemáticas de encontrarmos os requisitos dinâmicos do sistema [12].

Casos de uso essenciais foram originalmente criados para apoiar o projeto de IUs [22], uma vez que somos levados a deduzir, de forma correta, a concepção da interface a partir da lógica de utilização do sistema como um suporte a realização de tarefas e não a partir da lógica de funcionamento das funções da aplicação, como usualmente realizado [20]. Quando utilizados durante a elicitação dos requisitos, auxiliam na compreensão do escopo do problema e das funções requeridas. Eles tendem a permanecer corretos por um longo período de tempo, uma vez que excluem decisões de projeto e descrevem apenas a essência dos casos de uso, isto permite a identificação de padrões de casos de uso, sendo úteis no desenvolvimento de novos projetos [22].

##### 4.2.1.2 Notação

Estruturamos cada caso de uso essencial em duas seções principais: informação descritiva e narrativa. A primeira

seção contém dados que identificam e explicam o propósito do caso de uso. Enquanto a segunda seção é o corpo narrativo do caso de uso, descreve os aspectos dinâmicos, a interação. Poderíamos ainda ter uma outra seção opcional com dados históricos e de autoria. A figura 2 ilustra a organização do caso de uso essencial.

Os elementos da seção de informação descritiva são melhores explicados a seguir: ID é meramente um identificador único do caso de uso. Útil no processo de desenvolvimento, para rastreamento (*tracking*) e para uso em ferramentas; Nome do caso de uso é a declaração de seu objetivo. A primeira palavra do nome deve ser um verbo no tempo presente em voz ativa, que expressa o propósito do caso de uso, como sugerido em [2,11]; Ator Primário é o nome do papel que inicia a interação; Prioridade do caso de uso é estabelecida de acordo com sua importância e frequência, podendo ser classificado como primário, secundário ou opcional [8]; Recursos necessários no caso de uso. Podem ser concretos, por exemplo o cartão bancário, ou abstratos, por exemplo a conta corrente.

A seção narrativa de um caso de uso essencial é estruturada através do estilo de narração numerada particionada, ou seja, a narrativa é segmentada em etapas individuais numeradas, dispostas em duas colunas (intenção do usuário e responsabilidade do sistema). O uso da numeração é pertinente para futuros refinamentos. Esta notação torna explícito o limite das perspectivas internas e externas e apresenta uma maior legibilidade e uma menor loquacidade do que outros estilos de narração (maiores detalhes em [18]). É então razoavelmente adequada para a descrição dos principais aspectos dos casos de uso, pois permite uma comunicação mais fácil entre os envolvidos no desenvolvimento do software [20].

<b>Informação descritiva</b>	
ID:	Nome:
Prioridade:	Ator Primário:
Recursos:	
<b>Narrativa</b>	
Intenções do usuário	Responsabilidades do sistema
(eventos)	(eventos)
<b>Histórico</b>	
Data:	Autor:

Figura 2 – Caso de uso essencial estruturado.

#### 4.2.2 Casos de Uso Singulares

##### 4.2.2.1 Introdução

Casos de uso singulares são casos de uso expandidos que modelam, além do curso normal de um caso de uso essencial, os “desvios” que podem mudar a interação do curso normal. Estes desvios - englobando as anormalidades, as exceções, os problemas, os enganos, as interrupções e os obstáculos [9] - são chamados de singularidades, donde vem a expressão “caso de uso singular” [20]. Por definição, uma singularidade é a causa de um problema [20]. Para o nível dos requisitos, um problema é uma diferença semântica entre a situação atual e uma situação desejada. Então, é necessário sempre ligar uma singularidade a um problema e um problema a um objetivo [20].

Os casos de uso essenciais são refinados através da determinação de potenciais singularidades, a cada situação específica e singular um caso de uso singular. Portanto, um caso de uso essencial pode dar origem a diversos casos de uso singulares. Esta concentração em pontos específicos ajuda refinar a compreensão dos requisitos do sistema proposto [20].

A consideração de singularidades é uma questão muito importante e pode ser crucial para usabilidade de um sistema, devemos pensar em alternativas para alcançar os objetivos quando ocorrer alguma singularidade [2]. Todo objetivo pode ser bloqueado de alguma maneira por condições ou eventos no sistema ou no ambiente. Levar em conta as possíveis singularidades força o desenvolvedor (e também o usuário) a pensar em soluções flexíveis e robustas para estas situações reais (onde o uso acontece em modos não antecipados pelo desenvolvedor) e não para situações idealizadas [9,20]. Devemos capturar as condições de extensão antes de preocuparmos-nos em como lidar com elas [2]. Para cada singularidade identificada são propostas possíveis ações defensivas e corretivas que o sistema poderia realizar para respectivamente evitar ou corrigir seus efeitos.

Os casos de uso essenciais permitem avaliar a consequência de singularidades e possíveis mecanismos defensivos e corretivos [9] e também permitem estabelecer proposições de novas ações para alcançar um determinado objetivo e descobrir novos objetivos para a nova situação futura [20].

Naturalmente, um problema é o potencial número ilimitado de casos de uso que seriam necessários para representar todas as diferentes possibilidades de progresso de uma interação [20]. A solução, como proposto em [9,20], é definir um subconjunto de casos de uso em termos de singularidades e combinações de singularidades consideradas relevantes, – “salientes” [9]; “interessantes” [20] – porque eles têm características que ajudam exemplificar aspectos cruciais do comportamento do sistema proposto e não são redundantes. Em TAREFA [20] as singularidades relevantes são divididas em duas categorias não-ortogonais: singularidades críticas que são as obstruções ao curso normal do caso de uso essencial; e singularidades representativas que são as obstruções que aparecem mais

freqüentemente. Além disso, uma combinação de singularidades também é considerada como uma singularidade. Enquanto que a relevância das singularidades individuais é determinada independente do domínio da aplicação a relevância da combinação é dependente [9].

Casos de uso não ocorrem em qualquer situação é necessário considerar o contexto, que demarca seu escopo e define suas condições [7]. As condições são predicados dos estados do sistema que devem ser considerados antes (pré-condições) ou depois (pós-condições) da execução das ações [20]. Provavelmente não conheçamos todas as condições de um caso de uso desde o principio, a análise de singularidades ajudam a identificar as pré e pós-condições. As condições que levarão o usuário a seguir um determinado curso de interação. Devemos distinguir as condições que separam o curso normal dos outros possíveis cursos.

#### 4.2.2.2 Notação

Para representarmos adequadamente os casos de uso neste nível de abstração, definimos como pré-requisito que a estrutura adotada permitisse a associação de conceitos teleológicos (atores, objetivos, singularidades, etc).

Um conceito adicional é fundamental para a descrição dos casos de uso essenciais, o conceito de episódios [9]. Um episódio é um conjunto particular de ações, executadas sob condições, que torna possível a descrição operacional do objetivo de um caso de uso [9]. Assim, a seção narrativa de um caso de uso singular consiste de uma seqüência de episódios, cada um correspondente a algum objetivo ou singularidade que bloqueia a realização desse objetivo. O conceito de episódio é recursivo, cada episódio pode ser considerado como um pequeno caso de uso [17], mas no papel de caso de uso informações adicionais são necessárias [23].

O conceito de episódio auxilia a lidar com grandes conjuntos de casos de uso, pois permite o reuso de partes dos casos de uso, um mesmo episódio pode aparecer em vários casos de uso, evitando assim a duplicação do trabalho de especificação de casos de uso e oferecendo um modelo de casos de uso mais compacto [7]. Diversos autores já utilizam o conceito de episódios para estruturar casos de uso [7,9,11,17].

O elemento básico do modelo de caso de uso é a ação, podendo ser atômica ou composta. Uma ação atômica materializa uma interação de um agente para outro agente. Uma ação composta é formada por uma organização (temporal ou de dependência lógica) de ações atômicas. As ações são descritas pelos termos ator seguido da ação.

Adotamos para a representação dos casos de uso singulares uma notação que segue um “esquema gramatical” [20]. Segundo Potts [9], casos de uso são um tipo de história, o autor então propõe um esquema de caso de uso que é semelhante a um esquema de história. Um esquema gramatical não é uma gramática geradora, no senso conhecido da Teoria das Linguagens, porque permite deixar indeterminada a estrutura de certos elementos - tipicamente por uma descrição informal [20]. Esta representação é especialmente bem adaptada para abordagens baseadas em objetivos como evidenciado pela explícita associação de um objetivo a cada episódio no caso de uso [4]. Representações de casos de uso que permitem os analistas e *stakeholders* considerarem singularidades são uma fonte especialmente rica para identificar novos requisitos. Casos de uso esquemáticos orientados a objetivos parecem apoiar a identificação e refinamento de requisitos mais do que outras representações [4]. O esquema gramatical, notação dos casos de uso singulares é apresentado na figura 3.

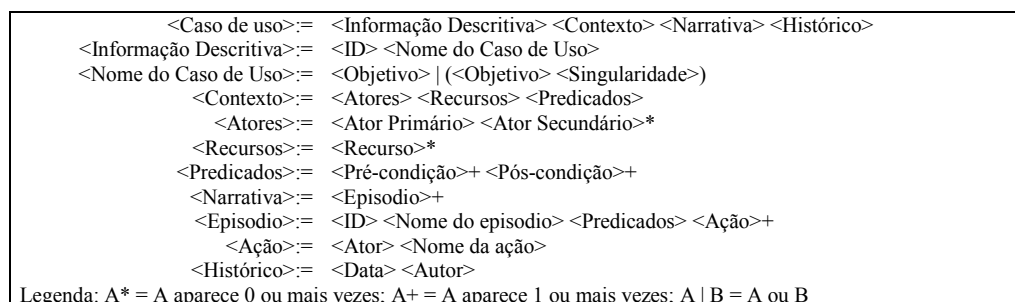


Figura 3 – Esquema gramatical para casos de uso singulares.

### 4.2.3 Casos de Uso Concretos (Cenários)

#### 4.2.3.1 Introdução

Casos de uso concretos são casos de uso no menor nível de abstração. Um caso de uso concreto é uma instância específica e representativa de um caso de uso singular tendo um resultado particular com respeito ao objetivo do ator iniciador. Ele descreve um comportamento do ambiente e do sistema que aparece em uma situação concreta e restrita de uso (presente ou futura). Corresponde ao conceito de cenário utilizado em IHC.

Durante estágios de pré-implementação, analistas e *stakeholders* podem não compreender muitas de suas decisões propostas. Cenários facilitam o melhor entendimento do futuro sistema, forçando os envolvidos no desenvolvimento a prestar atenção em particularidades do uso real [4]. Esses cenários servem para decidir se as atuais soluções



operacionais dos objetivos satisfazem as reais necessidades dos usuários. Cenários podem elucidar como o sistema proposto irá suportar praticas concretas de trabalho [4].

#### 4.2.3.2 Notação

Cenários ajudam desenvolvedores e usuários a administrar a complexidade conceitual de desenvolvimento de software clarificando as conseqüências comportamentais concretas dos requisitos ou propostas de desenvolvimento. Outra estratégia de desenvolvimento relacionada que faz isto é a prototipagem. Cenários podem ser usados independentemente da prototipagem, e protótipos podem ser usados na ausência de cenários, contudo o uso conjunto tem se mostrado benéfico [4].

Para o nível concreto é adotada a combinação de uma descrição textual em linguagem natural e um protótipo de “baixo nível”, adequado para exemplificar os aspectos concretos dos casos de uso para o usuário [20]. A forma de narrativa textual, contendo informação sobre situações e usuários ilustrativos, parece ser mais memorável e inteligível para a maioria dos leitores do que a forma estruturada [9].

### 4.3 Construção de Casos de Uso

Em nossa abordagem o processo de construção dos casos de uso é interativo e iterativo, mas a princípio segue um método *top-down*, onde casos de uso são construídos e refinados sucessivamente (pelo analista) respeitando suas propriedades estruturais. O uso de estruturas definidas para cada um dos níveis juntamente com a utilização de heurísticas guiam a construção dos casos de uso. Este método resume-se em:

- Construir os casos de usos essenciais a partir da hierarquia de objetivos representada pelos modelos de tarefas minimais e da definição das responsabilidades essenciais do sistema; Cada caso de uso é construído associando atores, objetivos, intenções e responsabilidades.
- Construir os casos de uso singulares refinando os casos de uso essenciais e definindo as singularidades associadas aos objetivos e aos serviços de cada caso;
- Construir os casos de uso operacionais definindo os objetos clientes e servidores dos serviços do sistema descritos nos casos de uso singulares, caso a concepção seja orientada a objeto;
- Construir (instanciar) a partir dos casos de uso singulares os casos de uso concretos que serão experimentados recursivamente com os usuários.

Como exemplo corrente neste artigo, utilizaremos o ATM (Banco 24 horas), este tipo de exemplo é onipresente na literatura e assim, seu uso facilita a comparação de nossa abordagem com outros trabalhos. É difícil imaginar que estejamos trabalhando no domínio de ATMs como se tais máquinas não existissem, porém, este exemplo é útil para o propósito deste trabalho, já que a familiaridade com ele nos ajuda a explicar as noções e conceitos discutidos.

#### 4.3.1 Construção de Casos de Uso Essenciais

Um usuário tem o objetivo de causar um certo efeito (F) no ambiente sob certas condições, temos então F como parte dos requisitos. Se um artefato ou dispositivo pode criar o efeito desejado, então é possível atribuir o efeito como uma função (responsabilidade) do artefato. Cabe ao desenvolvedor criar um sistema que possua esta responsabilidade, a fim de auxiliar um usuário a alcançar seu objetivo [5]. Em sistemas interativos os objetivos são alcançados por certas interações entre o usuário e o sistema, isto é, por relacionamentos entre intenções do usuário e responsabilidades do sistema, modelados através de casos de uso.

Construir os casos de uso essenciais implica em definir os dois componentes da interação usuário-sistema: as intenções do usuário, derivadas diretamente da estrutura hierárquica de objetivos do modelo de tarefas minimal e as responsabilidades correspondentes do sistema [20].

Para este mais alto-nível de descrição de um caso de uso, uma responsabilidade é uma abstração das ações a serem executadas pelo sistema para transformar o estado atual no estado final desejado, onde o objetivo da tarefa interativa, representado no caso de uso, é alcançado [20]. As responsabilidades (funções) do sistema sustentarão as tarefas interativas para atingir os objetivos e podem ser classificadas em dois tipos: responsabilidade receptiva do sistema, onde o evento do sistema é uma recepção de uma intenção ou responsabilidade expressiva do sistema, onde o evento é uma resposta do sistema a uma intenção [20].

A determinação das responsabilidades do sistema é uma atividade criativa de dominio específico, devem ser realizadas propostas alternativas do sistema [9]. Podem ser concebidos vários sistemas que cumprem os objetivos dos usuários, eles diferirão no modo que os objetivos são operacionalizados. Casos de uso permitem definir com maior precisão como o usuário deseja usar as tarefas interativas [20], casos de uso ajudam a avaliar, refinar e escolher a proposta [9]. Na realidade as responsabilidades do sistema devem ser refinadas durante todo o processo de desenvolvimento até a definição detalhada do comportamento do sistema [20].

Cada sub-objetivo do modelo de tarefas minimal, representado pela figura 4, é considerado como uma intenção do usuário. A cada intenção, as responsabilidades receptivas e expressivas são definidas, como mostramos na figura 5.

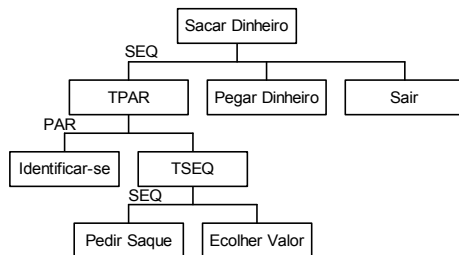


Figura 4 – Modelo de tarefas minimal “Sacar dinheiro” [20].

Intenção do Usuário	Responsabilidade Receptiva do Sistema	Responsabilidade Expressiva do Sistema
Identificar-se	Aceitar identificação	Verificar identificação Oferecer serviços
Pedir saque	Aceitar pedido	Pedir valor
Escolher valor	Aceitar valor	Verificar saldo Liberar dinheiro, recibo e cartão
Pegar dinheiro		Estar pronto para outro serviço
Sair		Estar pronto para nova seção

Figura 5 – Intenções e responsabilidades associadas para a tarefa “Sacar dinheiro”.

A organização dos eventos no caso de uso essencial segue a organização lógica e temporal do modelo de tarefas minimal [20]. O uso de seqüência de eventos como narração é encorajado baseado na abordagem de Pimenta [20]: “Mesmo entrelaçando tarefas, para uma dada instancia, o usuário só manipula uma tarefa interativa por vez, independente da possibilidade de paralelismo do computador”. Um caso de uso é resultado da escolha de uma seqüência. Em [20] são propostas as heurísticas para escolha das seqüências: seqüência habitual do usuário; seqüência prévia (formal) da organização; seqüências novas propostas para testar o usuário.

A figura 6 ilustra o caso de uso essencial que modela a seqüência habitual do usuário para realizar o objetivo “Sacar dinheiro”. O nome do caso de uso é objetivo principal do modelo de tarefas minimal e o ator iniciador deve provir da explícita associação entre atores e objetivos, representados por nós através de uma tabela. A prioridade do caso de uso essencial é na verdade a prioridade do objetivo do caso de uso que deve ser previamente discutida entre o analista e os *stakeholders* ao criar o modelo de tarefa minimal.

Informação descritiva	
ID: UC1	Nome: Sacar dinheiro
Prioridade: Primário	Ator Primário: Cliente
Narrativa	
Intenções do usuário	Responsabilidades do sistema
1. Identificar-se	2. Aceitar identificação 3. Verificar identificação 4. Oferecer serviços
5. Pedir Saque	6. Aceitar pedido 7. Pedir valor
8. Escolher valor	9. Aceitar valor 10. Verificar saldos 11. Liberar dinheiro 12. Liberar recibo 13. Liberar cartão
14. Pegar o dinheiro	15. Estar pronto para outro serviço
16. Sair	17. Estar pronto para nova seção

Figura 6 – Caso de uso essencial “Sacar dinheiro”.

Porém, devemos estar atentos ao fato de que nem sempre uma interação é estritamente seqüencial. Cada intenção pode ter relações temporais diferentes com as outras intenções do mesmo caso de uso. Por exemplo, na figura 4, *PAR* indica que as intenções podem ser executadas em paralelo e *SEQ* indica que o ordenamento deve ser seqüencial. Desta forma, a intenção *Identificar-se* pode ser executada de uma maneira integrada com as intenções *Pedir saque* e *Escolher valor*. Este aspecto de ordenação opcional ou flexível é uma situação muito comum em modelagem de tarefas, mas raramente é modelado explicitamente no corpo narrativo de casos de uso [18].

Em TAREFA [20] é definido um conjunto de sinais, inspirado na notação UAN (*User Action Notation*), para lidarmos com estas relações como partes intrínsecas da narrativa do caso de uso. A maioria destas relações é adotada por modelos de tarefas e modelos de diálogo [20]. Esta prática tem um efeito positivo sobre a usabilidade, pois

encoraja o desenhista de IU a modelar como seqüências completamente ordenadas somente as interações onde a ordem é realmente fixa ou determinada [18]. Desta forma, conforme já constatado em [1], o termo “seqüência”, utilizado na definição do conceito de caso de uso, torna-se impreciso, pois não indica uma sucessão estritamente ordenada.

#### 4.3.2 Construção de Casos de Uso Singulares

A construção dos casos de uso singulares consiste de algumas atividades: Refinamento dos casos de uso essenciais em casos de uso singulares; Identificação de singularidades; Elaboração de sub-objetivos ou ações para evitar a ocorrência de singularidades ou corrigir seus efeitos; e Finalmente construção de um caso de uso singular para cada singularidade considerada relevante.

A atividade de refinamento do caso de uso essencial em caso de uso singular é alcançada por [20]: Decomposição das intenções dos usuários, aqui consideradas como episódios, presentes nos casos de uso essenciais em ações mais elementares; Identificação de pré e pós-condições para cada caso de uso e episódio.

A figura 8 ilustra um exemplo de um episódio de caso de uso singular escrito através do esquema gramatical, o qual é um refinamento da intenção *Identificar-se* do caso de uso essencial “Sacar Dinheiro”.

Informação Descritiva	
ID: UC1	
Nome: Sacar dinheiro	
Contexto	
Ator Primário: Cliente	
Ator Secundário: ATM	
Recursos: Cartão do cliente, Conta do Cliente	
Pré-condição: ATM pronto para usar	
Pós-condição: ATM pronto para usar; Cliente com o dinheiro	
Narrativa	
Episódio	
ID: 1	
Nome: Identificar-se	
Pré-condição: ATM pronto para usar	
Pós-condição: Cliente válido	
Ação:	
Ator	Ação
Cliente	Fornecer ID do cliente
ATM	Aceitar ID do cliente
ATM	Verificar ID do cliente
ATM	Perguntar senha do cliente
Cliente	Fornecer senha do cliente
ATM	Aceitar senha do cliente
ATM	Verificar senha do cliente

Figura 8 – Episódio “Identificar-se” refinado do caso de uso essencial “Sacar dinheiro”.

Então é necessário capturar as singularidades antes de preocuparmos-nos em como lidar com elas. Em geral, um modo intuitivo de identificarmos as singularidades é analisar o caso de uso normal e realizar um questionamento sistemático [3]. Questões do tipo: “*What if...?*” [2]; “*What can go wrong with this action?*” [20]. Obtemos uma lista de singularidades e propomos ações defensivas e corretivas. Representamos através de tabelas os relacionamento entre objetivos/obstáculos/objetivos corretivos ou defensivos.

Propomos o uso de heurísticas, extraídas de [9,20] para guiar a obtenção do conjunto de casos de uso singulares:

- Cada objetivo da hierarquia de objetivos deve ser realizado com sucesso por um episódio em pelo menos um caso de uso. Esta heurística garante que todos objetivos da hierarquia serão considerados;
- Respeitar as dependências lógicas e temporais da hierarquia dos objetivos. Não há necessidade de considerar os objetivos fora da hierarquia;
- O conjunto total dos casos de usos tem que conter pelo menos uma instância de cada categoria de singularidade - crítica ou representativa - mas cada singularidade é associada somente a um caso de uso, sem repetições.

#### 4.3.3 Construção de Cenários

A construção de cenários (casos de uso concretos) é feita diretamente a partir de casos de uso singulares. Cada ação dos episódios de um caso de uso escolhido é instanciada pela determinação de suas características concretas que servirão para a experimentação com o usuário [20].

### 4.4 Experimentação dos Casos de Uso

Erros de requisitos são freqüentemente devido à validação imprópria. Durante a experimentação de casos de uso são

feitas avaliações para determinar como o sistema proposto atende as necessidades dos *stakeholders*. Revisões iterativas de casos de uso com *stakeholders* ajudam desenvolvedores a assegurar que os requisitos estão completos, consistentes e realísticos [4].

A experimentação dos casos de uso concretos consiste em simular a interação seguindo a sequência descrita pelo cenário, usando conjuntamente os protótipos desenvolvidos [20]. Esta simulação permite o usuário avaliar de cada vez a funcionalidade e a usabilidade do sistema, isto pode implicar mudanças em casos de uso, nos requisitos ou em ambos [20]. O uso destes cenários, não apenas possibilita verificar a atual solução proposta no real contexto de uso, mas também descobrir novos requisitos que emergem somente na prática. Como resultado dessa avaliação as consequências da introdução da nova tecnologia podem ser analisadas e as práticas de trabalho podem repensadas pelos usuários.

Em TAREFA [20] encontramos valiosas recomendações a serem utilizadas durante as sessões de experimentação de cenários com os usuários: usar o cenário no contexto concreto de utilização; usar um cenário por vez; explicar somente as alternativas possíveis; e deixar o usuário expor suas idéias sem repressões.

#### **4.5 Modificação dos Casos de Uso**

As críticas dos usuários são as origens das modificações dos casos de uso. Estas modificações podem ser pequenas e gerar apenas variantes dos casos de uso concretos mudando exclusivamente suas características concretas; ou grandes provocando algumas alterações a nível operacional, singular ou até mesmo essencial do caso de uso [20]. O analista deve revisar os casos de uso visando adequá-los as expectativas dos usuários. Os casos de uso modificados devem ser novamente instanciados e experimentados junto ao usuário até que se chegue consensualmente a um conjunto de requisitos considerado satisfatório pelos participantes [20].

#### **4.6 Integração dos Casos de Uso**

Até o presente momento cada caso de uso foi utilizado como um apoio para a determinação dos requisitos do sistema na perspectiva de uma situação específica de uso, então dissociada dos outros casos de uso [20]. Deste modo, obtemos uma coleção solta de casos de uso separados, modelos parciais com aspectos restritos do sistema. Necessitamos integrar os casos de uso para alcançar uma visão global de utilização para um ator do sistema.

O número de possíveis relacionamentos entre os casos de uso cresce exponencialmente com o número de casos de uso, se esses relacionamentos são formalizados eles podem mais facilmente ser identificados e suportados [23]. Episódios representam explicitamente o relacionamento entre casos de uso. Conforme abordado em [2] o relacionamento entre os casos de uso aparecem de forma direta na narrativa, não sendo necessária uma representação gráfica, como os diagramas de casos de uso da UML, para representar os relacionamentos. Episódios comuns entre os casos de uso devem ser globalmente renumerados visando à integração dos casos de uso [20]. A visão global de utilização para um ator do sistema pode ser representada através de uma lista de ator-objetivo ao invés de uma representação gráfica [2]. A informação é a mesma, o que difere é a apresentação.

### **5 Considerações Finais**

Uma visão geral de nossa proposta orientada a objetivos para construção sistemática de casos de uso e cenários foi apresentada ao longo deste trabalho. Embora o modelo de casos de uso de TAREFA tenha sido aplicado a estudos de caso reais, o modelo proposto – que contém modificações importantes – ainda está sendo aplicado. Embora nossa abordagem ainda não esteja totalmente consolidada, estamos aprimorando-a e já podemos destacar alguns de seus aspectos positivos:

O uso dos quatro níveis de abstração propostos em TAREFA [20] permitiu utilizarmos de maneira integrada casos de uso e cenários, o que possibilitou aliarmos as perspectivas de IHC e ES. O uso de conceitos teleológicos consegue resolver ou contornar grande parte dos problemas conceituais discutidos na seção 2. Por exemplo, a hierarquia dos objetivos ajuda a delimitar a cobertura dos casos de uso enquanto o escopo de um caso de uso é definido como possibilitando no mínimo a cobertura (satisfação ou não) de um objetivo.

Episódios podem ser usados como mecanismos de modularização. A estrutura episódica possibilita representarmos os relacionamentos entre os casos de uso dentro da própria descrição narrativa.

Casos de uso criados a partir de nossa abordagem são gerados de maneira objetiva ao invés de intuitiva. Casos de uso gerados sistematicamente permitem estudar os comportamentos não normativos dos usuários mais a fundo, o que contribui para uma investigação mais dirigida à usabilidade de sistemas.

O uso de estruturas definidas para cada nível de abstração juntamente com a utilização de heurísticas guia a construção dos casos de uso em nossa proposta. Estamos planejando a construção de guias (*guidelines*) adicionais a serem oferecidas para o analista no ato da descrição dos casos de uso, de maneira similar ao encontrado em [10].

## Referências

- [1] A. Cockburn, Structuring Use Cases with Goals, *Journal of Object Oriented Programming*, Sep/Oct 1997, p. 35-40 e Nov/Dec 1997, p. 56-62.
- [2] A. Cockburn, Writing Effective Use Cases. Boston: Addison-Wesley, 2000.
- [3] A. I. Antón, Goal-Based Requirements Analysis. *Proc. Second International Conference on Requirements Engineering (ICRE'96)*, Colorado Springs, Colorado, USA, Apr. 1996, pp. 136-144.
- [4] A. I. Antón, and C. Potts, A Representational Framework for Scenarios of Systems Use, *Requirements Engineering Journal*, Springer Verlag, 3(3-4), 1998, pp. 219-241.
- [5] B. Chandrasekaran and H. Kaindl, Representing Functional Requirements and User-System Interactions, in *Proceedings of AAAI-96 Workshop on Modeling and Reasoning About Function*, Portland, OR, USA, August 1996.
- [6] B. Regnell, K. Kimbler and A. Wesslén, Improving the Use Case Driven Approach to Requirements Engineering, in *Proceedings of the 2nd IEEE International Symposium on Requirements Engineering*, York, England, 1995.
- [7] B. Regnell, M. Andersson and J. Bergstrand, A Hierarchical Use Case Model with Graphical Representation, in *Proceedings of the IEEE International Symposium and Workshop on Engineering of Computer Based Systems*, Friedrichshafen, 1996.
- [8] C. Larman, Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientados a Objetos, Trad. L. A. M. Salgado, Porto Alegre: Bookman, 2000.
- [9] C. Potts, Using Schematic Scenarios to Understand User Needs, in *Proceedings of Symposium on Designing Interactive Systems Processes, Practices, Methods & Techniques (DIS'95)*, University of Michigan, Ann Arbor, Michigan, USA, ACM, August 23-25, 1995, pp. 247-256.
- [10] C. Rolland, C. Ben Achour, Guiding the Construction of Textual Use Case Specifications, in *Data & Knowledge Engineering Journal*, Vol 5, nº1-2, North Holland, Elsevier Science Publisher, March 1998, pp. 125-160.
- [11] C. Rolland, C. Souveyet and C. Ben Achour, Guiding Goal Modeling Using Scenarios, *IEEE Transactions on Software Engineering*, 24(12), December 1998, pp. 1055-1071.
- [12] E. Kavakli, P. Locopoulos and D. Filippidou, Using Scenarios to Systematically Support Goal-Direct Elaboration for Information System Requirements, in *Proceedings of International IEEE Symposium and Workshop on Engineering of Computer Based System (ECBS'96)*, Friedrichshafen, Germany, March 11-15, 1996, pp. 308-314.
- [13] G. Booch, J. Rumbaugh and I. Jacobson, *The Unified Modeling Language User Guide*, Addison-Wesley, 1999.
- [14] H. Kaindl, An Integration of Scenarios with Their Purposes in Task Modelling, in *Proceedings of Symposium on Designing Interactive Systems Processes, Practices, Methods & Techniques (DIS'95)*, University of Michigan, Ann Arbor, Michigan, USA, ACM, August 23-25, 1995, pp. 227-235.
- [15] I. Jacobson, Basic Use-Case Modeling, *Report on Object Analysis & Design*, V. 1, N. 2, August 1994.
- [16] J. M. Carroll, Artifacts and Scenarios: An Engineering Approach, in: *Monk, A., Gilbert, N. Perspectives on HCI: Diverse Approaches*. Computer and People Series, 1995, chap. 6, pp. 121-143.
- [17] J. C. S. P. Leite, et al, Enhancing a Requirements Baseline with Scenarios. *Requirements Engineering Journal*, Springer-Verlag London Limited v.2, n.4, 1997, pp.184-198.
- [18] L. Constantine, et al, Structure and Style in Use Cases for User Interface Design. *Preprint of a chapter to appear in Object-Oriented User Interface Design*, M. van Harmelen, editor. <http://www.foruse.com/Presentations/structurestyle2.pdf>.
- [19] L. Constantine and L. Lockwood, *The Usability Challenge: Can UML and the Unified Process Meet It?* Australia, 22 November 2000. <http://www.foruse.com/Presentations/tools.pdf>
- [20] M. S. Pimenta, TAREFA: Une Approche pour l'Ingénierie des Besoins des Systèmes Interactifs, *Toulouse: Doctorat de l'université Toulouse I*, 1997, 285 pp.
- [21] M. S. Pimenta, TAREFA: Uma Abordagem para Engenharia de Requisitos de Sistemas Interativos, in *Anales de Terceras Jornadas Iberoamericanas de Ingenieria de Requisitos y Ambientes Software (IDEAS00)*, Cancún, México, 5 al 7 de abril del 2000.

- [22] R. Biddle, J. Nobel and E. Tempero, Essential Use Cases and Responsibility, in *Object-Oriented Development*, New Zealand, August, 2001. <http://www.foruse.com/Files/Papers/euc-responsibility.pdf>
- [23] T. A. Alspaugh, A.I. Antón, T. Barnes, and B. Mott, An Integrated Scenario Management Strategy, *International Symposium on Requirements Engineering (RE'99)*, Limerick, Ireland, June 1999, pp. 142-149.
- [24] W. A. Cybis, M. S. Pimenta, M. C. Silveira and L. Gamez, Uma Abordagem Ergonômica para o. Uma Abordagem Ergonômica para o Desenvolvimento de Sistemas Interativos, in *Atas do IHC'98 I Workshop sobre Fatores Humanos em Sistemas Computacionais: Compreendendo Usuários, Construindo Interfaces*, Maringá-PR, Brasil, 1998.