

Um Método para Elicitação e Modelagem de Requisitos Baseado em Objetivos

Marcia C. F. Carvalho & Zair Abdelouahab

Departamento de Engenharia Elétrica
Universidade Federal do Maranhão, Campus do Bacanga
marcia.carvalho@elo.com.br, zair@elo.com.br, zair@dee.ufma.br

Abstract

Este trabalho propõe uma integração da abordagem CREWS L'ecritoire baseado em cenários com a abordagem de casos de usos descrita por Regnell et al e o Método GBRAM baseado em objetivos. Dessa forma são adicionados ao trabalho de Regnell et al a noção de pedaço de requerimento (RC), as estratégias de descoberta do objetivo através dos relacionamentos AND, OR e de refinamento entre RCs, além de estender o modelo com um nível físico onde são descritas as ações internas do sistema.

Em contrapartida, é adicionado à abordagem CREWS- L' Ecritoire o acoplamento objetivo-caso de uso onde a obtenção dos cenários origina-se da aplicação de uma estratégia de refinamento sobre os casos de uso, a integração de cenários normais e excepcionais em um modelo de uso sintetizado, com uma visão única por ator.

No que concerne ao Método GBRAM, foram extendidas as relações de dependência de contrato entre objetivos para dependência alternativa e dependência condicional exclusiva, assim como foi introduzida a técnica de caso de uso, conforme descreve Regnell et al em [15,16] e extendida a noção de cenários para caracterizar situações não só excepcionais, mas também as normais e variacionais.

Com a integração desses métodos, o método resultante irá representar um adicionamento de cada um isoladamente. Esse método se propõe a acoplar objetivos e casos de uso numa decomposição top-down em diferentes níveis de abstração. Ele combina a utilização destes dois conceitos em um pedaço de requerimento (RC). Um RC é um par <objetivo, caso de uso>, relacionados através de relacionamentos AND, OR e de refinamento. Esses três tipos de relacionamentos entre RCs conduzem a uma organização hierárquica de RCs em três níveis de abstração: o funcional, o comportamental e o físico.

Keywords: Use Cases, Scenarios, Goals, Requirements, Elicitation, Modelling

1. Introdução

A fase de definição de requisitos se constitui na fase crítica do ciclo de vida de um sistema, pois uma compreensão completa dos requisitos de software é fundamental para um bem-sucedido desenvolvimento do mesmo, uma vez que os requisitos formam a base para o planejamento, o acompanhamento do desenvolvimento, e a aceitação dos resultados do projeto.

As conseqüências do desenvolvimento inadequado da atividade de definição de requisitos consistem em produção de sistemas que não atendem às necessidades dos usuários, aumento de custos, realização de atividades desnecessárias ou até mesmo duplicadas, usuários insatisfeitos e, conseqüentemente, desentendimento com os engenheiros de requisitos, além

do aumento considerável da tarefa de manutenção.

Entre as técnicas que podem ser utilizadas para apoiar a tarefa de especificação e elicitação de requisitos encontram-se use cases e cenários, cuja principal vantagem consiste em respeitar a perspectiva do usuário, fazendo com que a coleta de requisitos atenda aos seus desejos e tornando essencial sua participação no processo.

Dentre os vários métodos que adotam casos de uso como técnica de elicitação e modelagem de requisitos [6,7,11,15,16] destacamos o trabalho de Regnell et al, descrito em [15,16], que apresenta o Método UORE (Engenharia de requisitos baseada em uso) e o modelo hierárquico de caso de uso. Por outro lado, entre os métodos que utilizam cenários [10, 12, 13, 17] a abordagem CREWS- L'Ecritoire [17] emprega o acoplamento cenário-objetivo na descoberta de objetivos e obtenção de cenários. E, dentre as abordagens que utilizam objetivos [4, 5, 8, 9, 19] destacamos o Método GBRAM.[4,5].

Esse trabalho se propõe a integrar a abordagem CREWS-L' Ecritoire [17] baseada em cenários com a abordagem de caso de uso descrita por Regnell et al [15,16] e o Método GBRAM [4,5]. Com a integração desses métodos, o método resultante irá representar uma extensão de cada um isoladamente. O artigo é organizado da seguinte forma: na seção 2 apresentamos os métodos utilizados na integração. Na seção 3 é apresentado o método integrado proposto para engenharia de requisitos que consiste em um modelo de processo e um modelo de produto. Na seção 4 o método proposto é aplicado em um estudo de caso para elicitar os requisitos para um sistema de restaurantes. Na seção 5, o método proposto é comparado com outras abordagens existentes. E, por último, na seção 6 fazemos algumas considerações acerca do que foi realizado.

2. Background

2.1 Abordagem CREWS- L' Ecritoire

A abordagem CREWS- L'Ecritoire utiliza um acoplamento bidirecional objetivo-cenário permitindo mover-se a partir de objetivos para cenários e vice-versa. Considerando a direção objetivo-cenário, à medida em que é descoberto um objetivo, este é operacionalizado por um cenário. Explorando o relacionamento objetivo-cenário na direção reversa, i.e, de cenário para objetivos, a abordagem guia o processo de elicitação de requisitos, descobrindo novos objetivos através da análise de cenários textuais.

De acordo com [1] essa abordagem auxilia no endereçamento dos seguintes aspectos:

Obtenção de cenários [2], provendo guias de estilo e conteúdo informais para a escrita de cenários, auxiliando na limitação do tamanho dos mesmos, com sugestões de templates¹ para a escrita de sentenças; organização de cenários, através da integração de cenários em coleções de cenários [3]; Transição de cenários informais para formais; Suporte metodológico, com a definição de regras para a descoberta de objetivos e verificação de cenários, implementadas em um ambiente de software L' Ecritoire[20]; Análise de cenários, através da descoberta de requisitos/objetivos analisando os cenários de uma coleção objetivo-cenário; Fragmentação de cenário, através da organização de coleções objetivos-cenários de uma maneira hierárquica.

A abordagem tem como núcleo o RC, definido como o par <objetivo, cenário>, organizados em três níveis de abstração: contextual, interação do sistema e interno do sistema, onde cada nível corresponde a um tipo de RC.

No nível contextual, são identificados os serviços que o sistema deve prover para uma organização e suas razões. O RC contextual acopla um objetivo de projeto a um cenário de serviço. Um cenário de serviço descreve o fluxo de serviços entre agentes que são necessários para executar o objetivo de projeto.

¹ Estrutura pré-definida , molde

O nível de interação de sistema focaliza-se nas interações entre o sistema e seus usuários necessárias para executar os serviços associados ao sistema no nível contextual. Um RC de interação do sistema combina um objetivo de serviço a um cenário de interação do sistema, onde são descritos uns fluxos de interações entre o sistema e seus usuários para executar o objetivo de serviço.

O nível interno do sistema focaliza-se no que o sistema pode internamente executar. Um RC interno do sistema combina um objetivo do sistema a um cenário interno do sistema. Um objetivo do sistema expressa uma maneira possível de executar uma ação identificada em um cenário de interação do sistema.

Os RCs podem ser interligados através de três tipos de relacionamentos, a saber: AND (composição), OR (alternativa) e de refinamento, sendo organizados como uma coleção de requisitos em uma hierarquia de RCs.

Os relacionamentos de composição (AND) interligam aqueles RCs que requerem um ao outro para definir um sistema funcionando completamente. Os relacionamentos alternativos (OR) representam maneiras alternativas de executar o mesmo objetivo. Os relacionamentos de refinamento são utilizados para descrever RCs em diferentes níveis de abstração, sendo direcionado pela parte do cenário que considera toda interação do cenário no nível i como um objetivo a ser executado no nível $i + 1$.

O modelo de processo da abordagem CREWS- L ' Ecrtoire é representado como um mapa , i.e, um grafo direcionado com intenções como nós e estratégias como setas, mostrado na figura 1:



Fig.1: Modelo de Processo da abordagem CREWS-L' Ecrtoire.

Um aspecto importante do modelo de processo CREWS- L ' Ecrtoire consiste na provisão de diferentes estratégias para suportar a elicitação de objetivos: estratégia alternativa, de composição e refinamento. A essas estratégias são associadas regras para descoberta de novos objetivos, implementadas em um protótipo de software L' Ecrtoire.

O modelo de produto mostrado na figura 2 representa os principais conceitos utilizados por esta abordagem, relacionados por links de composição, associação ou is-a.

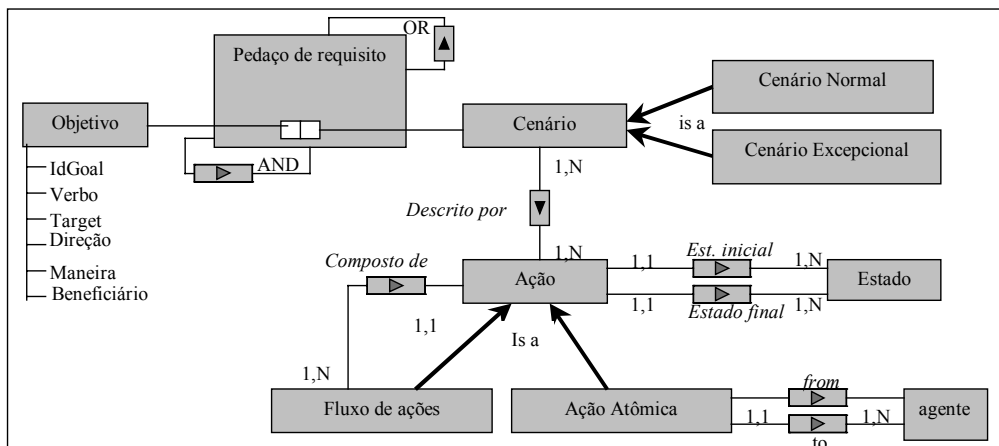


Fig. 2: Modelo de Produto da abordagem CREWS- L' Ecrivitoire.

2.2 Processo UORE

Em [15] é proposto um processo de Engenharia de requisitos como uma melhoria da Análise Dirigida a caso de uso [11], estendendo-a com uma fase de síntese onde os casos de uso são integrados em um modelo de uso sintetizado que captura os requisitos funcionais e aspectos de uso do sistema.

O processo UORE (ver fig. 3) consiste de duas fases: análise e síntese. Na fase de análise ocorre a identificação, descrição dos atores e casos de uso e uma unificação da terminologia dessas descrições em dicionário de dados.

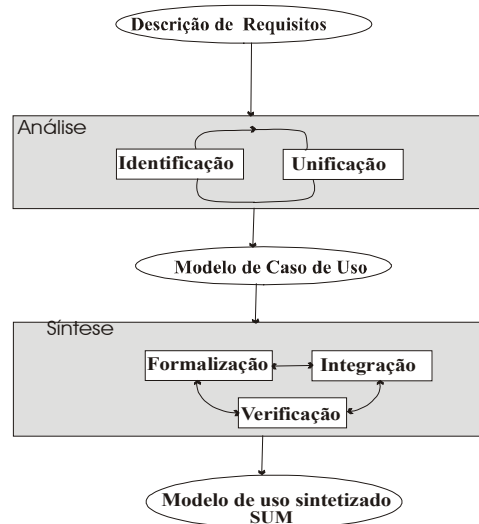


Fig. 3: Processo UORE.

A fase de síntese formaliza os casos de uso, integra-os e cria o modelo de uso sintetizado, constituindo-se de três atividades: formalização dos casos de uso, integração dos casos de uso e verificação.

A atividade de formalização destina-se a produzir especificações de casos de uso formais para cada caso de uso identificado na fase de análise. Essa coleção de especificações de casos de uso são representados em uma linguagem gráfica que expressa a ordem temporal de estímulos dos usuários, respostas do sistema e operações atômicas.

A atividade de integração propõe-se a realizar a união de diferentes especificações de casos de uso e produzir um modelo de uso sintetizado. Para isso, cada especificação de caso

de uso é transformada em um cenário abstrato de uso, desenhado como uma seqüência de ações do usuário e do sistema, interconectados por transições que representam as mensagens resultantes de cada ação. Em seguida, todos os cenários abstratos de uso produzidos para um ator específico são unificados resultando no modelo de uso sintetizado, que consiste em uma visão de uso do sistema para um ator.

E, por último a atividade de verificação destina-se à obtenção de um modelo de uso sintetizado completo e consistente.

2.3 Modelo de use case hierárquico

Regnell et al [16] propõe um modelo de caso de uso hierárquico com representação gráfica organizados em três níveis: ambiente, estrutura e evento. No nível ambiente, o ambiente de cada caso de uso é descrito associando-o com atores relacionados, serviços e objetivos.

O nível estrutura descreve cada caso de uso como um grafo de episódios, representando um fluxo coerente e demarcado de eventos. Neste nível também são definidos operadores de seqüência, alternativa, repetição, exceção e interrupção, assim como as pré e pós-condições que restringem o escopo do caso de uso.

No nível evento, os episódios são descritos em detalhes adicionais em termos dos eventos que ocorrem em cada episódio. O nível evento ordena os eventos da mesma forma como os episódios são ordenados no nível estrutura.

Os principais conceitos utilizados, mostrados no modelo conceitual (ver fig.4), são sumarizados como segue. Um use case descreve uma situação de uso do sistema. Um contexto é utilizado para expressar as pré e pós-condições que restringem o escopo do caso de uso. Uma pré-condição é definida como propriedades do sistema e seu ambiente que precisam ser satisfeitas para invocar o caso de uso. Uma pós-condição descreve o estado do sistema após o término do caso de uso. Um ator representa um conjunto de usuários que tem algumas características comuns com respeito a porque e como é utilizado o sistema. Um serviço é descrito por um conjunto de casos de uso. Objetivos são utilizados para categorizar usuários em atores.

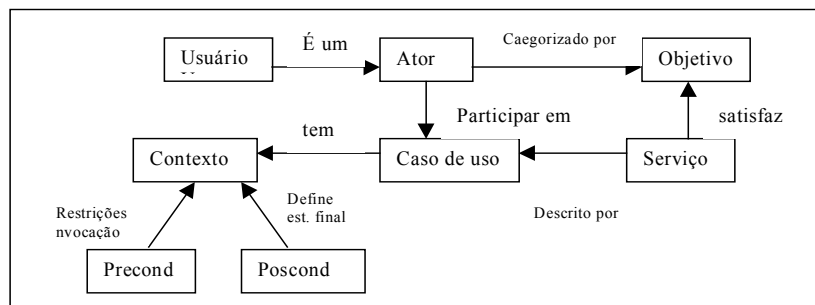


Fig 4: Modelo conceitual da abordagem de Regnell et al.

2.4 Método GBRAM

Em [4,5] é apresentado o Método GBRAM (Método de Análise de Requisitos Baseados em Objetivos) que pressupõe que os objetivos não tenham sido previamente documentados ou explicitamente elicitados a partir dos stakeholders. Assim, o analista deve trabalhar a partir de todas as fontes de informação disponíveis como fluxos de processos ou de informação, declarações textuais de necessidades, e/ou fontes adicionais de informação como transcripts de entrevistas com os stakeholders para determinar os objetivos do sistema desejado.

O Método GBRAM descrito na figura 5 envolve duas fases: a fase de análise de objetivos e a fase de refinamento de objetivos, produzindo como saída o DRS (Documento de

Requisitos do Software), que provê uma comunicação não ambígua entre stakeholders², além de suportar evolução e validação de requisitos.

As atividades de análise de objetivo podem ser sumarizadas como segue:

- Exploração da documentação existente para a identificação inicial de objetivos;
- Identificação de objetivos, stakeholders e seus agentes³ responsáveis;
- Organização de objetivos de acordo com as relações de dependência e classificação dos objetivos conforme as condições alvo.

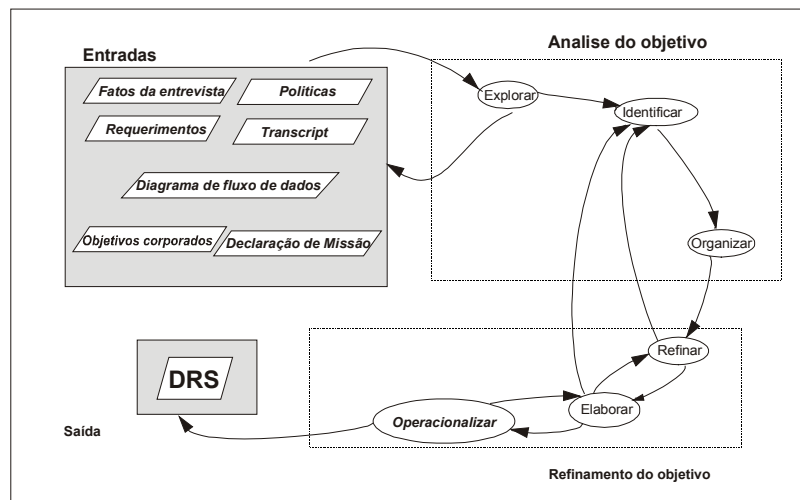


Fig. 5: Atividades do Método GBRAM.

No que concerne às atividades de refinamento do objetivo, estas podem ser sumarizadas como segue:

- Refinamento de um conjunto objetivo com a diminuição do tamanho do conjunto objetivo;
- Elaboração de cenários para descobrir objetivos e requisitos escondidos;
- Operacionalização dos objetivos em requisitos operacionais.

Na fase de elaboração dos objetivos são identificados os obstáculos dos objetivos, considerando as possíveis maneiras de falhas dos objetivos, e como estes podem ser bloqueados facilitando a antecipação de casos excepcionais. Os obstáculos são identificados por meio de perguntas como " De quais outros objetivos ou condições este objetivo depende?" "Pode o agente responsável pela falha do objetivo realizar o objetivo?" " Pode a falha de outro objetivo completamente causar o bloqueio deste objetivo?" " Se este objetivo é bloqueado, quais são as conseqüências?" "Quais outros objetivos ou condições este objetivo depende?" ou através das palavras-chaves não, entretanto e quando.

Uma vez especificados os obstáculos de objetivos, os engenheiros de requisitos devem considerar os cenários possíveis que são prováveis para cada obstáculo. Cenários são descrições comportamentais de um sistema e seu ambiente que surgem a partir de situações restritas, e oferecem uma maneira natural para descrever circunstâncias excepcionais, especiais, auxiliando os engenheiros de requisitos a descobrirem objetivos escondidos ou aspectos necessários a uma resolução adicional que poderia de outra forma negligenciada.

² São as pessoas envolvidas no processo de engenharia de requisitos, ou seja, clientes, atores e proprietários.

³ Agentes são responsáveis pela realização dos objetivos dentro de uma organização ou sistema.

Para identificar cenários, são sugeridas por este método as seguintes perguntas " O que acontece se este objetivo não é executado?" "Por que este objetivo não foi executado?" " Quais as circunstâncias sob a qual este obstáculo ocorre?" "Por que este obstáculo ocorre?".

Realizada a especificação de objetivos, esta informação deve ser operacionalizada e traduzida em expressões de linguagem natural de requisitos no DRS (Documento de Requisitos de Software). Durante a operacionalização, as ações descritas pelos stakeholders e extraídas a partir da documentação disponível são relacionadas de volta aos objetivos.

Assim, os objetivos operacionalizados, agentes responsáveis, stakeholders, restrições, obstáculos e cenários são mapeados em ações firmadas em um conjunto de esquemas objetivo (modelos que especificam os relacionamentos entre objetivos e agentes em termos de eventos que causam uma mudança de estado).

O conjunto de esquemas objetivo é mapeado para um documento de requisitos de software, incorporando toda a informação adquirida durante a análise e elaboração dos objetivos.

3. Método proposto para elicitación e modelagem de requisitos

O método proposto para elicitación e modelagem de requisitos se destina a acoplar objetivos e casos de uso numa decomposição top-down em diferentes níveis de abstração. Ele combina o uso de dois conceitos: casos de uso e objetivo em um pedaço de requisitos (RC). Um RC é um par <objetivo, caso de uso>, relacionados através de relacionamentos AND, OR e de refinamento. Esses três tipos de relacionamentos entre RCs conduzem a uma organização hierárquica de RCs em três níveis de abstração: o funcional, o comportamental e o físico.

O nível funcional focaliza-se nos serviços e casos de uso associados, onde um objetivo de serviço é acoplado ao conjunto de casos de uso necessários para executar um serviço, formando o RC funcional. Cada um dos casos de uso são adicionalmente detalhados pelos RCs comportamentais no nível comportamental.

O nível comportamental se destina a identificar os objetivos dos casos de uso descobertos no nível anterior com a operacionalização dos casos de uso através de cenários estruturais. Dessa forma o RC comportamental acopla um objetivo funcional a um cenário estrutural, onde o objetivo funcional expressa uma maneira de prover uma funcionalidade identificada no nível anterior e, portanto estabelece um link hierárquico com o pedaço funcional. O cenário estrutural descreve o fluxo de ações do usuário e do sistema necessários para executar o objetivo funcional.

Já o nível físico detalha uma maneira possível em que o sistema pode internamente executar uma interação identificada no cenário estrutural. O RC físico associa um objetivo do sistema a um cenário interno do sistema, onde o objetivo do sistema expressa uma maneira de executar uma ação identificada no cenário estrutural e o cenário interno do sistema descreve o fluxo de interações entre os objetos do sistema para executar o objetivo do sistema.

Os relacionamentos definidos entre os RCs podem ser relacionamentos AND; relacionamentos OR e relacionamentos de refinamentos explicados a seguir.

Os relacionamentos AND são utilizados entre modelos de casos de uso formando um pacote de serviços no nível funcional e, entre os cenários estruturais complementares no nível comportamental descrevendo um serviço e, por ultimo entre os cenários internos que compõem um cenário estrutural.

Os relacionamentos OR representam maneiras alternativas de executar o mesmo objetivo, sendo expressos entre RCs dentro do mesmo nível de abstração representando variações dos cenários normais ou cenários excepcionais no nível comportamental ou no nível físico.

Relacionamentos de refinamento relacionam RCs em diferentes níveis de detalhe. Ele é utilizado para refinar um caso de uso do nível i em cenários estruturais no nível $i + 1$. Além disso, ele pode ser também empregado para especificar(refinar) os cenários estruturais em cenários internos do sistema. Os relacionamentos de refinamento são definidos a partir de um RC funcional para um RC comportamental e, de um RC comportamental para um RC físico.

A numeração utilizada pelos RCs refere-se à colocação de índices ou um subitem, de acordo com o tipo de relacionamento. Para os relacionamentos alternativos, é acrescentada à notação de RCs um expoente que inicia em 1 e vai sendo incrementado à medida que novos RCs alternativos vão sendo descobertos. Assim, no nível comportamental para o RC1.1 temos um RC cujo cenário é normal e ligados a ele tem-se vários RCs cuja numeração será RC1.1¹, RC1.1², e assim por diante. Para os relacionamentos de composição o último algarismo do RC é incrementado. Assim para o RC 1.1, teremos a ele ligados pelo relacionamento AND, RC1.2 e RC1.3, e assim por diante.

No caso dos relacionamentos de refinamento é acrescentado um novo nível à numeração e, portanto uma nova numeração. Dessa forma para o 1º nível funcional temos RC1, onde por meio da estratégia de refinamento obtém-se o RC1.1 no nível comportamental e o RC1.1.1 no nível interno. Para o RC2, teremos o RC2.1 no nível comportamental e o RC2.1.1 no nível interno. Os relacionamentos AND e OR ocorrem dentro do mesmo nível, enquanto o relacionamento de refinamento ocorre de um nível para o outro.

3.1 Fases do Método Proposto para elicitación e modelagem de requisitos

O método proposto consiste de um modelo de processo e um modelo de produto representando o processo de desenvolvimento do produto. O mapa do processo (ver figura 6) mostra várias fases: Elicitar um objetivo, especificar use cases, elaborar cenários, reorganizar objetivos e cenários, refinar objetivos e cenários e operacionalizar objetivos e cenários.



Fig. 6: Modelo de processo do Método Proposto.

A seguir são explicadas cada uma das fases, numeradas de acordo com a seqüência que devem ser executadas, assim como as estratégias relacionadas.

1 - Elicitação de objetivos

Nessa fase são identificados, inicialmente, os objetivos de serviço por meio da estratégia de identificação dos objetivos iniciais e/ou estratégia de perguntas específicas, definidas a seguir:

- ✓ Estratégia de identificação dos objetivos iniciais: consiste em extrair os objetivos, a partir da análise da documentação existente ou das entrevistas realizadas entre clientes, usuários e engenheiro de requisitos. Essa análise da documentação existente é realizada examinando-se transcrições de entrevistas com os usuários à procura por palavras (verbos) de ação, uma vez que os usuários tendem a expressar seus requisitos através de operações, descrições de processos, fluxogramas, enfim, quaisquer documentos (normalmente informais) que existam acerca do domínio da aplicação onde o sistema será desenvolvido.
- ✓ Estratégia de perguntas específicas: concerne em estabelecer perguntas específicas de maneira a identificar os objetivos para o primeiro nível de nossa abordagem.

Além dessas duas estratégias definidas acima, outras estratégias que são utilizadas para elicitación de objetivos são:

- ✓ Estratégia dirigida a template: propõe uma template para elicitación dos objetivos, onde essa template consiste no nome do objetivo, identificação do objetivo e N° do RC à qual ele pertence.
- ✓ Estratégia dirigida a estrutura do objetivo: a aplicação dessa estratégia consiste em definir valores alternativos para os diversos parâmetros, combinando-os resultando em objetivos possíveis a serem escolhidos pelo usuário para elicitación. A estrutura do objetivo adotada consiste em um verbo + parâmetros, onde parâmetros:: objeto + maneira +[de]/ [para] [fonte] [destino], sendo fonte e destino parâmetros opcionais.
- ✓ Estratégia lingüística: visa à fase de formalização posteriormente definida dentro da abordagem.

2 – Especificar casos de uso

Essa etapa é realizada para cada um dos objetivos de serviço identificados na fase anterior. Para especificar um caso de uso são propostas três estratégias:

- ✓ Estratégia dirigida a template: propõe especificar casos de uso por meio de uma template, composta pelo nome do ator⁴, código do caso de uso, nome do caso de uso, [identificação do ator] e código do objetivo.
- ✓ Estratégia gráfica: os casos de uso são construídos conforme a notação de Jacobson [7], utilizando o conceito de caso de uso definido em Regnell et al [10], onde a descrição de casos de uso não permite múltiplos atores, existindo uma visão única para cada ator envolvido no caso de uso.
- ✓ Estratégia baseada em ator: propõe identificar os atores do sistema como um meio de identificar os casos de uso

Após especificados os casos de uso, devem ser identificados os objetivos funcionais, correspondentes ao segundo nível de abstração de nossa abordagem.

3 – Elicitación de objetivos

Nesse caso, podem ser consideradas três estratégias para obtenção dos objetivos funcionais, a seguir descritas.

⁴ Ator, dentro do método proposto, significa alguém que executa interações com o caso de uso.

- ✓ Estratégia de descoberta do refinamento: essa estratégia é aplicada sobre os casos de uso originando os objetivos funcionais, que são operacionalizados pelos cenários estruturais.
- ✓ Estratégia de descoberta alternativa: essa estratégia permite obter variações para os objetivos de serviço no nível funcional.
- ✓ Estratégia de composição: permite identificar os objetivos complementares para um dado objetivo e auxilia a identificar um pacote de serviços para um dado sistema.

4 – Especificação de cenários

Durante essa fase, os cenários são escritos segundo algumas regras de estilo e conteúdo definidas em [2] seguindo a estratégia dirigida a template, composta pelo código do cenário, agente, pré e pós-condições, ação e código do objetivo.

Observa-se que existe um elemento da template Ator/Agente onde se faz necessário algumas explicações. Para o nível funcional, utiliza-se o “Ator” definido como “Alguém que executa interações com o caso de uso”. Para o nível comportamental e físico, utiliza-se o termo agente, definido como “aquele que é responsável pela execução do cenário”. No caso do nível interno do sistema, o agente terá por default o sistema, uma vez que este nível descreve as interações entre os objetos do sistema para executar um objetivo do sistema.

As pré-condições tem um formato que consiste em uma estrutura normalmente condicional ou iterativa. A primeira é realizada com if, como por exemplo, if código de acesso do usuário = válido; if senha do usuário = válida; if código do cardápio= válido. No caso da iterativa, é utilizado o while, como, por exemplo, enquanto código do cardápio for válido.

As pós-condições consistem em um estado obtido após a realização do cenário. Possuem um formato de um **DO** ação como, por exemplo. DO status do cardápio = " recebido".

5 – Elicitação de novos objetivos

No nível comportamental, a elicitação de novos objetivos é realizada utilizando três diferentes estratégias:

- ✓ Estratégia-descoberta-alternativa: permite identificar todos os objetivos alternativos para um dado objetivo.
- ✓ Estratégia da descoberta da composição: essa estratégia permite identificar os objetivos complementares para um dado objetivo.
- ✓ Estratégia de refinamento: no nível comportamental, durante a realização dessa estratégia são identificados novos objetivos do sistema, pertencentes ao nível físico, oriundos das interações ocorridas no cenário estrutural. Assim, observa-se que ela ocorre a partir de um RC comportamental para um RC físico.

6 – Reorganização de objetivos e cenários

A fase de reorganização de objetivos e cenários caracteriza-se por serem estabelecidas relações de dependência entre objetivos e cenários, que são reorganizados de modo a satisfazer os critérios definidos.

Uma das relações de dependência utilizada entre objetivos foi a relação de dependência por precedência, conforme definida em [4,5]. Outra dependência de precedência adotada entre objetivos é a dependência de contrato entre objetivos G1 e G2, já definida anteriormente e encontrada em [5].

Neste trabalho, extendemos a dependência de contrato entre objetivos para

dependência de contrato alternativa entre objetivos G1, G2, G3 e G4, onde se G1 ocorre então G2/G3/G4 ocorrem, sendo expresso por $G1 \rightarrow G2/G3/G4$. Nesse tipo de dependência, se G1 for realizado, então G2/G3/G4/.../ Gn é também automaticamente realizado. É importante observar que vários agentes podem executar um mesmo cenário em diferentes tempos, mas somente um agente pode executar um cenário em um tempo t, não existindo portanto a execução paralela de objetivos. Dessa forma, os objetivos G1, G2, G3,..., Gn podem ser acionados automaticamente, mas somente um será executado em um tempo t.

Com relação aos agentes, o método proposto permite a execução paralela de agentes, desde que sejam cenários diferentes. Outro aspecto que deve ser ressaltado é que um agente pode executar um mesmo cenário em diferentes tempos.

Outra extensão proposta à dependência de contrato é dependência condicional exclusiva, onde objetivo G1 ocorre se pelo menos um dos objetivos G2/G3/G4/.../ Gn \rightarrow G1. Nesse caso, quando algum dos objetivos condicionais ocorrem, os demais são bloqueados e G1 é realizado.

Objetivos podem também compartilhar dependências entre agentes. Assim, para um agente completar um objetivo, outro agente pode ter que primeiro completar outro objetivo (indicando uma pré-condição). Essa dependência se destina a facilitar a produção de um modelo integrado a ser construído durante a fase de integração da abordagem proposta.

7 – Refinamento de objetivos e cenários

Após estabelecidas relações de dependências entre objetivos e, conseqüentemente entre cenários, inicia-se a fase de refinamento que utiliza uma estratégia de refinamento que consiste em um processo manual realizado pelo engenheiro de requisitos com o propósito de eliminar objetivos duplicados, unificar sinônimos, descobrir inconsistências, redundâncias e incomplezas internas nos cenários e objetivos. No caso desse processo ser semi-automatizado, faz-se necessário primeiramente formalizar os objetivos e cenários para que estes possam então serem verificados, por meio de uma estratégia de verificação, utilizada na fase de verificação da abordagem proposta.

São eliminadas redundâncias e reconciliadas objetivos sinônimos eliminando o objetivo cujo conteúdo é descrito por outro. É preferível que sejam identificados redundâncias e objetivos sinônimos após os objetivos terem sido ordenados de acordo com as relações de precedência, uma vez que estes objetivos se encontram adjacentes um com o outro, em um conjunto ordenado refletindo suas relações de precedência comum compartilhadas.

Incentiva-se também a participação dos usuários e clientes neste processo de refinamento, de modo que algumas discrepâncias sejam trazidas para sua atenção permitindo-os indicar qual nome do objetivo é mais apropriado.

8 – Operacionalização de objetivos e cenários

Na fase de operacionalização de objetivos e cenários, estes são descritos em mais detalhes e mapeados para seus respectivos esquemas. Dessa forma, possibilita a tradução da informação dos objetivos, use cases e cenários em um documento de requisitos de software por meio da tradução dos resultados do esforço das fases anteriores em um conjunto de esquemas.

Embora existam uma grande variedade de representações disponíveis, o método proposto emprega um estilo informal similar ao Método GBRAM. Durante essa fase, a estratégia utilizada consiste na estratégia dirigida a esquema, que se baseia na definição de esquemas-objetivos, esquemas-casos-de-uso e esquemas-cenários a seguir descritos.

O esquema-objetivo especifica os relacionamentos entre objetivos e cenários. A seguir, é definida a sintaxe para especificar esquemas de objetivos e seus atributos associados, bem como uma explicação de como objetivos podem ser expressos em uma série de esquemas

de objetivos que podem ser mapeados para um documento de requisitos de software e organizados de acordo com a topografia objetivo.

O modelo de objetivo e cenário incorpora toda a informação adquirida durante a análise de objetivo e elaboração de cenários. Ela é expressa em uma série de esquemas. Deve haver um esquema para cada objetivo e para cada cenário. A sintaxe do esquema para modelos objetivo utilizados para especificar objetivos consiste de um N° de RC, Nome do objetivo, Identificação do objetivo, Nível, Tipo, descrição, Agente/Ator, Código do cenário/Código do caso de uso, [pré-condições], [pós-condições], código do obstáculo e [subobjetivos].

É importante notar que o nível pode ser funcional, comportamental ou físico (interno) e que o tipo de objetivo pode ser definido como serviço, funcional e interno. Obstáculos referem-se a condições ou objetivos que oferecem impedimentos à realização do objetivo, sendo mecanismos que auxiliam na descoberta de cenários excepcionais.

Cada caso de uso é expresso na forma de um esquema de caso de uso. Existe pelo menos um esquema caso de uso para cada objetivo; na maioria dos casos, vários esquemas casos de uso devem ser providos para cada objetivo. A semântica de cada cláusula em um esquema caso de uso é composta pelo n° do RC, nível, tipo, nome do ator, código do caso de uso, descrição, nome do caso de uso e código do objetivo.

No caso do esquema para modelos de cenários e ações utilizados para especificar cenários e ações, a sintaxe do esquema consiste em: código do cenário, nível, tipo, descrição, ator/agente,[pré-condição], [pós-condição], ação, código do objetivo e [subobjetivos/episódios].

Percebe-se então que cada cenário é expresso na forma de um esquema de cenário. O nível do cenário pode ser funcional, comportamental ou interno e o tipo pode ser normal, excepcional. Uma ação envolve tanto ações do usuário como ações do sistema. As cláusulas opcionais estão colocadas entre colchetes.

Cada ação de um cenário também é expressa na forma de um esquema ação, salientando-se que existe pelo menos um esquema ação para cada cenário. Na maioria dos casos, vários esquemas ações devem ser providos para cada cenário. A seguir descreve-se a semântica de cada cláusula em um esquema ação: ação, tipo, reads, changes, N° da seqüência, código do cenário.

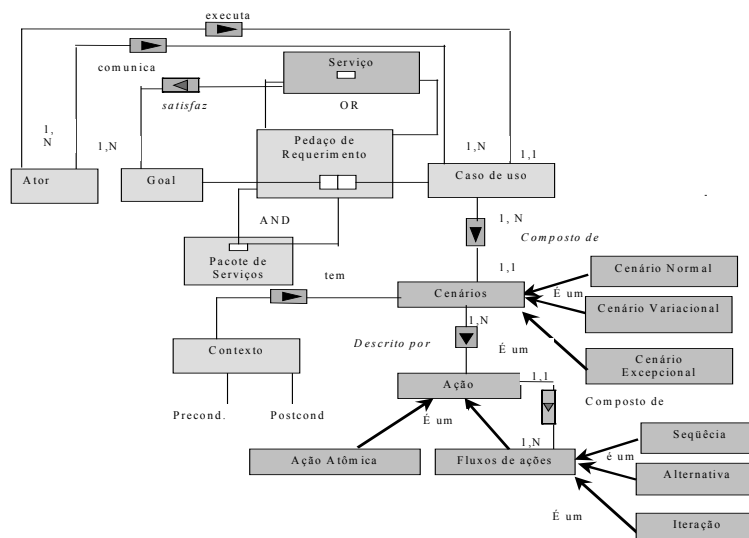


Fig. 7: Modelo do produto do Método Proposto.

Ressalta-se que o agente e as pré e pós-condições não foram colocados no esquema ação por serem os mesmos do esquema cenário. A notação utilizada pelos esquemas

propostos não é uma notação formal, sendo muito similar a uma modelagem de dados. Assim objetivos, cenários e ações são especificados utilizando os modelos esquemas apresentados.

O modelo de produto combina os conceitos de serviços, pacotes de serviços, atores, casos de uso, objetivos e cenários em uma única estrutura, conforme ilustra a figura 7.

4. Estudo de caso

Visando à verificação e posterior validação do método proposto, foi escolhido aplicá-lo sobre um domínio de restaurantes. Inicialmente, através da aplicação da estratégia de identificação dos objetivos iniciais e da estratégia de perguntas específicas, são realizadas as entrevistas e a análise da documentação existente, de modo a elicitare os objetivos de serviço.

Para a exemplo de restaurantes, foram encontrados os objetivos de serviço G1: Controlar cardápio; G2: Efetuar cotação; G3: Comprar produtos; G4: Distribuir alimentos e G5: Controlar qualidade. Estes objetivos são acoplados aos seus respectivos modelos de casos de uso para formar os RCs funcionais RC1, RC2, RC3, RC4 e RC5, como ilustra a fig.8.

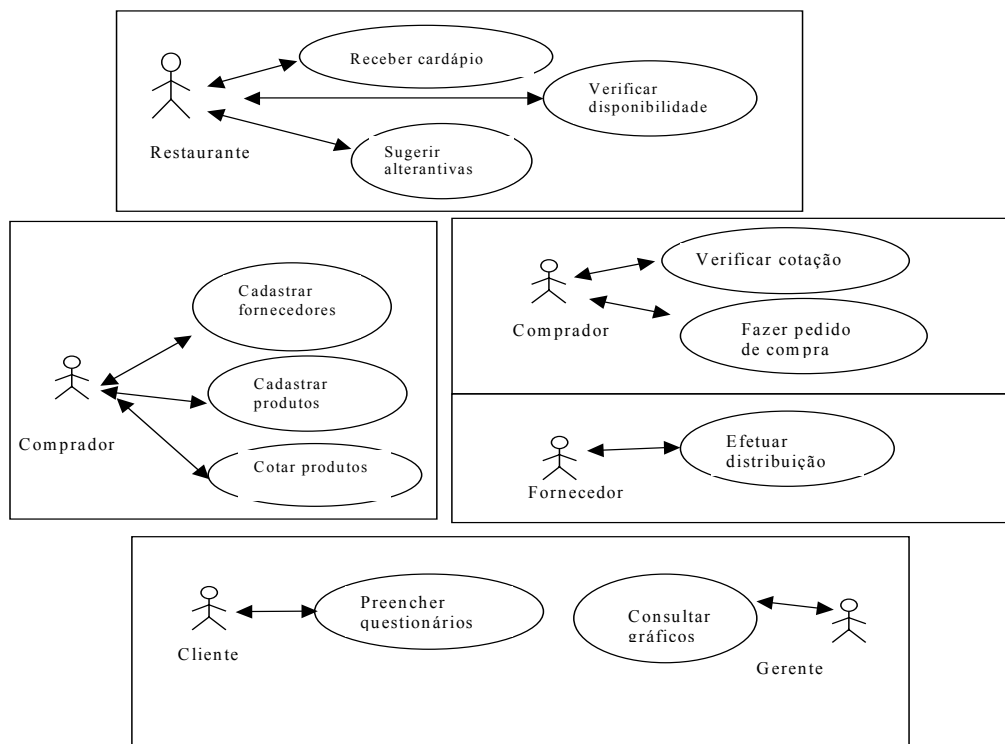


Fig. 8: Nível Funcional (RC1, RC2, RC3, RC4 e RC5).

Cada caso de uso do nível funcional origina um objetivo funcional do nível comportamental. Dessa forma, por exemplo, o caso de uso "Receber cardápio" origina no nível comportamental o G1.1: Receber cardápio de uma maneira normal" operacionalizado pelo Sc1.1. Abaixo exemplificamos uma parte do nível comportamental e o físico para o RC1. Por falta de espaço não será possível colocá-lo em sua totalidade.

G1.1: Receber cardápio de uma maneira normal

Agente/Ator: restaurante

Pré-condições: If código de acesso do usuário = válido

If Senha do usuário = válida

If Código do cardápio = válido

SC1.1:

1. O usuário entra com código de acesso
 2. O sistema valida código de acesso..
 3. O usuário entra com senha
 4. O sistema valida senha
 5. O usuário entra com código do cardápio
 6. O sistema valida código do cardápio.
 7. O usuário confirma recebimento
 8. O sistema exibe mensagem " Cardápio recebido com sucesso" para o usuário.
- Pós-condições: Do Status do cardápio = "recebido"

RC1.1¹

G1.1¹: Receber cardápio com código do cardápio inválido

Agente/Ator: restaurante

Pré-condições: If código de acesso do usuário válido

If Senha do usuário válida

If Código do cardápio inválido

SC1.1¹:

1. O usuário entra com código de acesso
 2. O sistema valida código de acesso..
 3. O usuário entra com senha
 4. O sistema valida senha
 5. O usuário entra com código do cardápio
 6. O sistema verifica código do cardápio.
 7. O sistema exibe mensagem " Código do cardápio inválido" para o usuário.
 8. O usuário entra com novo código do cardápio.
 9. O sistema valida código do cardápio
 10. O usuário confirma recebimento
 11. O sistema exibe mensagem " Cardápio recebido com sucesso" para o usuário.
- Pós-condições: Do Status do cardápio = "recebido"

RC1.1.1 (Nível físico)

G1.1.1: Verificar validade do código de acesso.

Pré-condições: If Código de acesso digitado

Agente/Ator: sistema

Sc1.1.1:

1. Abrir tabela de usuário no BD.
2. Pesquisar código de acesso na tabela.
3. Se código de acesso encontrado, então mostrar mensagem "código de acesso válido" para o usuário.
4. Senão exibir mensagem "código de acesso inválido" para o usuário.
5. Fechar tabela.

Pós-condição: DO Código de acesso validado

RC1.1.2

G1.1.2: Verificar validade da senha.

Pré-condições: IF senha digitada

Agente/Ator: sistema

Sc1.1.2:

1. Abrir tabela de usuário no BD.
2. Pesquisar senha por código de acesso na tabela.
3. Se senha encontrada, então mostrar mensagem " Faltam x dias para expirar sua senha"
4. Senão exibir mensagem "Senha não confere" para o usuário.
5. Fechar tabela.

Pós-condição: DO senha verificada

RC1.1.3

G1.1.3: Verificar validade do código de cardápio.

Pré-condições: IF Código do cardápio digitado

Agente/Ator: sistema

Sc1.1.3:

1. Abrir tabela de cardápio no BD.
2. Pesquisar código do cardápio na tabela.
3. Se código do cardápio encontrado, então mostrar mensagem " código do cardápio válido" para o usuário.
4. Senão mostrar mensagem "código do cardápio inválido" para o usuário.
5. Fechar tabela.

Pós-condição: DO código do cardápio verificado.

Foram realizadas as fases de reorganização e refinamento de objetivos e cenários, não demonstradas aqui por serem extensas para o escopo deste artigo, passamos então para a fase de operacionalização de objetivos e cenários. Na fase de operacionalização de objetivos e cenários, estes foram descritos em mais detalhes e mapeados para seus respectivos esquemas, conforme ilustram alguns esquemas-objetivos, esquemas-cenários e esquemas-ações do exemplo proposto.

- **Esquemas-objetivos**

Nº do RC: RC1

Código do objetivo: G1

Nome do objetivo Controlar cardápio **Nível funcional**

Nível: funcional

Tipo: serviço

Descrição:

Agente/Ator: restaurante

Cod-cenário/Cod-use case: MUC1.1

Nº do RC: RC1.1

Cod-obj: G1.1

Objetivo Receber cardápio de uma maneira normal

Nível: comportamental

Tipo: funcional

Descrição: **Nível comportamental**

Agente/Ator: restaurante

Cod-cenário/Cod-use case: SC1.1

Nº do RC: RC1.1.1

Cod-obj: G1.1.1

Objetivo Verificar validade do código de acesso

Nível: interno

Nível físico

Tipo: sistema
Descrição:
Agente/Ator: sistema
Cod-cenário/Cod-use case: SC1.1.1

- **Esquema-cenário**

Cod-cenário: SC1.1
Nível: comportamental
Tipo: normal
Descrição:
Pré-condições: If código de acesso do usuário válido
If Senha do usuário válido
If Cod- cardápio válido
Pós-condições: Do status do cardápio = " recebido"
Agente: restaurante
Ação:

1. O usuário entra com código de acesso
2. O sistema valida código de acesso..
3. O usuário entra com senha
4. O sistema valida senha
5. O usuário entra com código do cardápio
6. O sistema valida código do cardápio.
7. O usuário confirma recebimento
8. O sistema exibe mensagem " Cardápio recebido com sucesso" para o usuário.

- **Esquema-ação**

Ação: O sistema valida código de acesso
Tipo: sistema
Reads: código
Changes:
Nº da sequência: 1
Cod-cenário: SC1.1

5. Comparação com outras abordagens

A Abordagem CREWS-L'Ecritoire tem como um dos seus principais aspectos a utilização de um acoplamento bidirecional cenário-objetivo, onde à medida em que é descoberto um objetivo, este é operacionalizado por um cenário. Na direção reversa, i.e., de cenários para objetivos, a abordagem guia o processo de elicitação de requisitos descobrindo novos objetivos através da análise de cenários textuais. Na abordagem proposta observa-se um relacionamento bidirecional entre objetivo e caso de uso, onde ocorre a descoberta de casos de uso a partir de objetivos de serviço, assim como a geração de novos objetivos a partir dos casos de uso, sendo operacionalizados através de cenários estruturais cujas ações do sistema geram novos objetivos concretizados pelos cenários internos do sistema.

Ambas as abordagens adotam o RC como construtor básico para organização de cenários, sendo que na abordagem CREWS-L'Ecritoire são definidos os níveis contextual, interação do sistema e interno do sistema. Na abordagem proposta é adotado o conceito de caso de uso de acordo com [15, 16] que o define como uma maneira de utilizar o sistema descrito a partir da perspectiva de um único ator organizados nos níveis funcional,

comportamental e físico.

Uma das diferenças pode ser encontrada na geração de novos objetivos do nível físico, onde eles se originam das ações do sistema pertencentes às descrições dos cenários estruturais no nível comportamental. Na abordagem CREWS-L'Écritoire, a cada interação ocorrida no nível funcional origina-se um objetivo do sistema.

O Método GBRAM (Método de Análise de Requisitos baseados em objetivos) envolve duas fases até a produção do documento de software (DRS): a fase de análise dos objetivos e a fase de refinamento dos objetivos.

Na fase de análise dos objetivos, são identificados os objetivos, os stakeholders e os agentes responsáveis pela execução do objetivo. Objetivos são classificados em objetivos de manutenção e objetivos de realização. São estabelecidas relações de dependência entre os objetivos e construída a topografia de objetivos.

A abordagem proposta engloba as fases de elicitação de objetivo, especificação de casos de uso, elaboração de cenários, reorganização de objetivos e cenários, refinamento de objetivos e cenários e operacionalização de objetivos e cenários. Durante a aplicação do método proposto são identificados os objetivos, os casos de uso, os cenários dentro dos três níveis estabelecidos: funcional, comportamental e físico. Dessa forma, os objetivos são classificados em objetivos de serviço, objetivos funcionais e objetivos do sistema. São mantidas as relações de dependência do Método GBRAM e são criadas novas relações de dependência entre os objetivos, através da extensão da dependência de contrato para as variantes dependência de contrato alternativa e dependência condicional exclusiva.

Ambas as abordagens (GBRAM e a proposta) possuem uma fase de refinamento de objetivos onde são eliminados os objetivos redundantes e unificados os objetivos sinônimos. Durante a segunda fase do Método GBRAM que corresponde à fase de refinamento do objetivo, são identificados os obstáculos como meios possíveis de falhas dos objetivos, enquanto os cenários oferecem uma maneira natural para descrever circunstâncias excepcionais, permitindo a consideração de alternativas possíveis de operacionalizações de objetivos.

No método proposto adotou-se o conceito de obstáculos da mesma forma como foi adotado pelo Método GBRAM, ou seja, como um mecanismo de auxílio na descoberta de cenários excepcionais. No entanto, dentro da nossa abordagem, os cenários descrevem tanto situações excepcionais, como normais ou variacionais. No Método GBRAM, são considerados apenas os cenários excepcionais; outra diferença que pode ser visualizada refere-se à atividade de operacionalização onde objetivos podem ser expressos em uma série de esquemas objetivos mapeados para o documento de requisitos do software e organizados de acordo com a topografia de objetivos. No caso específico da nossa abordagem, existem esquemas-objetivo, esquemas-cenários, esquemas-casos de uso e esquemas-ações, dependendo do nível de detalhamento.

6. Conclusão

Esse trabalho teve como objetivos estudar algumas abordagens para Engenharia de Requisitos direcionados a casos de uso, cenários, objetivos e apresentar uma proposta de um método para a fase de Engenharia de Requisitos. A abordagem proposta propõe combinar objetivos, casos de uso e cenários como um mecanismo para elicitar os requisitos de um sistema. Ela é proveniente da integração da abordagem CREWS-L'Écritoire [17] com o trabalho de Regnell et al [15,16] e o Método GBRAM[4,5], consistindo em um modelo de processo e um modelo de produto, aplicado em um estudo de caso para elicitar os requisitos de um sistema para restaurantes e um para bibliotecas, não demonstrado aqui por falta de espaço.

Com a integração desses métodos, observa-se que o modelo RC da abordagem

CREWS-L'Ecritoire foi estendido para <objetivo, casos de uso>. Casos de uso são descritos a partir da visão de um único ator, permitindo que cenários normais, excepcionais e variacionais sejam integrados em um modelo de uso sintetizado. Dessa forma, a obtenção de cenários origina-se da aplicação de uma estratégia de refinamento sobre os casos de uso.

No que concerne ao Método GBRAM, foram estendidas as relações de dependência de contrato entre objetivos para dependência alternativa e dependência condicional exclusiva, assim como foi introduzida a técnica de caso de uso, conforme descreve Regnell et al em [15,16] e estendida a noção de cenários para caracterizar situações não só excepcionais, como as normais e variacionais dos objetivos.

No momento, estamos desenvolvendo um protótipo de uma ferramenta semi-automatizada de suporte ao desenvolvimento do método proposto. Sugerem-se como trabalhos futuros a extensão desse método para as fases de integração, formalização, verificação e validação dos objetivos e cenários. A fase de integração concentra-se em unificar os vários cenários escritos de acordo com os critérios selecionados, que podem ser atores, casos de uso, promovendo dessa forma uma visão geral do comportamento do sistema com uma visão única por ator. A integração de cenários deve ser realizada somente no nível comportamental a partir de critérios estabelecidos no nível funcional. Assim, depois de elaborados os objetivos e cenários, estes podem ser formalizados ou integrados, seja através de uma estratégia de formalização ou integração a serem definidas futuramente. Posteriormente à formalização destes cenários, estes podem ser submetidos à verificação utilizando uma estratégia de verificação e correção para que possam assim serem validados. A fase de validação consiste em juntamente com o usuário, validar os objetivos e cenários produzidos de maneira à obtenção de um documento de requisitos de software que satisfaça aos clientes, seja não ambíguo e consistente.

7. Referências

- [1] ACHOUR, C. Ben, Tawbi, Mustapha, SOUVEYET, Carine. Bridging the gap between users and requirements engineering: the scenario-based approach. Submitted to Computer Systems Journal, 1999.
- [2] ACHOUR, C. Ben. Guiding scenario authoring. Submitted to the Euro-Japanese conference, 1998.
- [3] ACHOUR, C. Ben, ROLLAND, Collete, SOUVEYET, Carine. A proposal for improving the quality of scenario collections. Proceedings of the Fourth International Workshop on Requirements Engineering: Foundations of Software Quality, REFSQ'98, Pisa, Italy Presses Universitaires de Namur (eds, E. Dubois, A. L. Opdhal, K. Pohl), pp. 29-42, 1998.
- [4] ÁNTON, Annie. Goal Identification and Refinement in the specification of Information Systems. Ph.D. Thesis. Georgia Institute of Technology, June 1997.
- [5] ÁNTON, Annie. Goal Based Requirements Analysis. Proceedings of the 2nd International Conference on Requirements Engineering' 96. Pp. 136-144, 1996.
- [6] COCKBURN, A. Structuring use cases with goals. Technical report. Human and Technology, 7691 Dell Rd, Salt Lake City, UT 84121, HaT. TR.95.1. Available at <http://members.aol.com/acockburn/papers/usecases.htm>, 1995.
- [7] DANO, B., BRIAND, H., BARBIER, F. A use case driven requirements engineering process. Third IEEE International Symposium on Requirements Engineering ' 97, Antapolis, Maryland, IEEE Computer Society Press, 1997.
- [8] DARDENNE, A., Lamsweerde, Van A.. and S. Fickas. Goal-directed Requirements Acquisition. Science of Computer Programming, 20 (1-2): 3-50, April 1993
- [9] GREEN, S. Goal-Driven Approaches to Requirements Engineering. Technical Report DOC TR 93-42 1994, Imperial College of Science, Technology and Medicine, Department of Computing Technical Report, London, UK, 1994.

- [10] HSIA, P, SAMUEL, J., GAO, J. et al. Formal approach to scenario analysis. IEEE Software, pp. 33-41, 1994.
- [11] JACOBSON, CHRISTERSON, M, JOHSSON. Object Oriented Software Engineering: a Use Case Driven Approach, Addison-Wesley Publishing Company, 1992.
- [12] LEITE, Júlio César Sampaio do Prado et al. Enhancing a Requirements Baseline with scenarios. In Third IEEE International Symposium on Requirement Engineering RE' 97, antapolis, Maryland, IEEE Computer Society Press, pp. 44-53, 1997.
- [13] POTTS, C.,Takahashi, K., Anton, A . Inquiry-Based Requirements Analysis. ,IEEE Software, 11(2): 21-32, March 1994.
- [14] RALYTÉ, Jolita, ROLLAND, Collete, PLIHON, Véronique. Method enhancement with scenario based techniques. To appear in: Proceedings of CAISE'99, 11th Conference on Advanced Information Systems Engineering Heidelberg, Germany June 14-18, 1999.
- [15] REGNELL, B., KIMBLER, Kristofer, WESSLÉN, A. Improving the use case driven approach to Requirements Engineering. Proceedings of Second International Symposium on Requirements Engineering,, York, UK March 1995.
- [16] REGNELL. B., ANDERSSON, M., BERGSTRAND, JOHN. A Hierarchical Use Case Model with Graphical representation. IEEE International Symposium and Workshop on Engineering of Computer-Based Systems, Germany, March 1996.
- [17] ROLLAND, C., SOUVEYET, C. ACHOUR, C. Ben. Guiding goal modelling using scenarios. IEEE Transactions on Software Engineering, special issue on Scenario Management, 1998.
- [18] ROLLAND, C., Grosz, G and Kla, R.. Experience with Goal-Scenario Coupling in Requirements Engineering. CREWS Report Series 98- 32. Fourth IEEE International Symposium on Requirements Engineering (RE'97). University of Limerick, Ireland. 7-11 June 1999.
- [19] SUTCLIFFE, A G. and N.A.M. Maiden. Bridging the Requirements Gap: Policies, Goals and Domains. In Proceedings of the Seventh International Workshop on Software Specification and Design, Redondo Beach, Califórnia, December 1993.
- [20] TAWBI, M., Souveyet, C. and Rolland, C. L'Ecritoire a tool to support a goal-scenario based approach to requirements engineering. CREWS Report Series. Submitted to Information and Software Technology Journal (ed. M. Shepperd) Elsevier Science Publishers. 1998.