ERACE-TOOL - UMA FERRAMENTA BASEADA EM CENÁRIOS PARA À ENGENHARIA DE REQUISITOS

João Caldas Júnior
FIL- Fundação Paulista de Educação e Tecnologia
Paulo C. Masiero
ICMC - Universidade de São Paulo
masiero@icmsc.sc.usp.br

Abstract

A tool to support requirements trading is presented. The Tool supports the ERACE approach. This approach starts from the system's requirement document and proposes to specify interactions between the system and its agents (scenarios), and then the scenarios are specified in detail. Heuristics to evolve from the requirements model to the analysis are also presented. An example to illustrates the approach is also presented.

Keywords: Engineering Requirements, Scenarios, Use Cases and UML

1. Introdução

A principal função da engenharia de requisitos é gerar especificações que descrevam de forma não ambígua, consistente e completa o comportamento do universo do domínio de um problema. Criar ferramentas que apoiem tal fase é uma forma de apoiar algumas atividades da engenharia de requisitos, o que contribui para aumentar a produtividade e a qualidade dos sistemas produzidos. Para que uma ferramenta suporte com êxito a fase de engenharia de requisitos, ela deve considerar que as especificações de requisitos são geradas a partir da análise das necessidades do sistema após consultas ao usuário e da redação de um documento de requisitos. A satisfação dos requisitos especificados pelo usuário é pré-condição básica para o sucesso de um software. A Erace-Tool é uma ferramenta baseada no processo ERACE, que utiliza um conjunto de diagramas para retratar, a partir da visão do usuário, os caminhos possíveis na utilização do sistema. Na próxima seção é apresentado o processo ERACE. Na Seção 3 é apresentada a ERACE-tool que suporta a criação dos modelos do processo que são exemplificados por intermédio de um estudo de caso. Na Seção 4 são apresentadas uma série de heurísticas de passagem para construção de um modelo de análise a partir dos modelos do processo ERACE. Finalmente na Seção 5 são apresentadas as conclusões deste trabalho.

2. Engenharia de Requisitos Apoiada por CEenários (ERACE)

Atualmente existem diversas técnicas de implementação da fase de engenharia de requisitos com base na construção de cenários [7][8][9][10]. Segundo Jacobson [7], existe uma grande lacuna entre os cénários e o desenho e implementação do modelo de objetos. Para preencher essa lacuna é necessário a identificação dos objetos e a solução de problemas como concorrência e consistência entre os cenários. Infelizmente a maioria dos métodos não apóia essa fase de transição. Outro problema com relação aos cenários é que mesmo apresentando uma boa forma de abstração funcional e comportamental, nenhum dos métodos atuais descrevem uma forma concreta de integração entre os vários cenários criados. Além disso, não existe uma verificação formal bem definida para a checagem dos vários cenários. De uma forma geral, todos os métodos são incompletos nos pontos considerados como críticos durante a construção de cenários[6]. Entretanto, as abordagens existentes proporcionam uma análise objetiva, que possibilita a composição de uma solução a partir da fusão dos pontos positivos das várias técnicas revisadas.

A proposta do projeto ERACE é fundir os pontos críticos das fases de engenharia de requisitos dos principais métodos que utilizam a técnica de construção de cenários, a fim de propor um processo para elicitar e modelar requisitos por meio de cenários e desenvolver uma ferramenta, baseada nesse processo, que apoie a elicitação e modelagem de requisitos. O processo ERACE[6], mostrado na Figura 1, é composto de duas fases: fase de Especificação das Interações e fase de Modelagem dos Cenários. Na fase de Especificação das Interações são feitas especificações das interações que ocorrem no sistema em um diagrama que contém informações sobre os atores e os cenários de uso. A base de informação para a construção desse diagrama é derivada de um documento inicial de requisitos de software, que deve conter todos os requisitos funcionais e de qualidade do software, incluindo capacidades do produto, recursos disponíveis e os benefícios e os critérios de aceitação. Mais informações sobre a construção de um documento de requisitos podem ser obtida no relatório técnico publicado por Turine e Masiero [3]. Todas as informações presentes no diagrama da fase de Especificação das Interações devem ser armazenadas em um glossário.

O processo ERACE possui características semelhantes às do método OOSE[7] e a notação da UML[1]. Em ambos são construídos diagramas (use cases diagram) bastante parecidos com os construídos na fase de especificação da ERACE e a

partir deles são feitas descrições mais específicas dos cenários. No entanto, existem singularidades especificadas em [6], que os tornam diferentes.

Figura 1 – Processo ERACE

No Processo ERACE, uma cena de uso pode ter a participação de mais de um ator e podem existir dois estilos de interação possíveis entre atores e cenários: interação principal, na qual os atores a ele relacionados iniciam a interação com seus cenários e interação participante, na qual os atores a ele relacionados participam, supervisionam ou gerenciam a interação de seus cenários. Cenários que envolvem mais de um ator durante seu desenvolvimento apresentam interações específicas para cada ator que nelas interage, por isso relacionar apenas um ator a uma determinada cena (como é feito no OOSE) e não especificar o tipo de envolvimento do ator com a cena (como é feito no UML), pode limitar a especificação da mesma.

No Processo ERACE cada detalhamento de uma cena de uso descreve o comportamento do sistema na visão de apenas um ator e estes detalhamentos contém condições que definem um contexto para sua iniciação. Um detalhamento de uma cena de uso completo tem um estado inicial que satisfaz seu contexto de iniciação, retornando no final do detalhamento ao mesmo estado no qual iniciou. Essas definições limitam o grau de complexidade dos cenários, aumentam sua completitude e possibilitam uma possível integração entre os vários cenário.

Um boa solução para evitar alguns problemas típicos em descrições de requisitos que utilizem linguagem natural é a criação de glossários com o objetivo de detalhar os termos usados na descrição dos requisitos. Com a formação deste léxico do domínio, todo vocabulário usado nas especificações e detalhamentos dos cenários pode ser sintetizado e unificado, facilitando o entendimento dos diagramas.

3. A Ferramenta

A ERACE-Tool é uma ferramenta baseada no processo ERACE que utiliza um conjunto de diagramas para retratar, a partir da visão do usuário, os caminhos possíveis na utilização do sistema. A ferramenta suporta as fases de Especificação das Interações e Modelagem de Cenários possibilitando assim: a criação de diagramas de interação dos cenário; diagramas abstratos detalhados dos cenários; glossários do domínio do problema e formalização dos diagramas abstratos detalhados. A primeira versão da ERACE-Tool não apoia a fase de integração do processo.

A ERACE-Tool será apresentada por meio de um estudo de caso de um sistema de auxílio a escrita de documentos técnicos. Para o desenvolvimento do estudo de caso utilizou-se o documento de requisitos do Sistema de APoio a EScrita de documentos técnicos (SAPES) que serviu como base para a elaboração do relatório técnico publicado por Turine e Masiero [3].

3.1 Construção do Diagrama de Interação dos Cenários (DIC)

A primeira atividade envolve a identificação das fronteiras de interação do sistema, a elicitação e a descrição dos atores e cenários de uso. Ainda no diagrama, são especificados os estilos de interação entre cada ator e os cenários de uso em que estes estão envolvidos, como também os relacionamentos entre os cenários de uso.

O processo de construção de um Diagrama de Interação dos Cenários é composto pelos seguintes passos: identificação dos atores; identificação de um conjunto de cenários de uso e seus relacionamentos; identificação das associações de interação entre atores e cenários de uso; e, identificação das dependências entre os cenários de uso.

Identificação dos atores: Um ator é identificado a partir de sua interação com o sistema e pelos eventos que envia para ou recebe do sistema. Nem sempre um usuário é necessariamente um ator, já que o mesmo usuário pode representar vários papéis no sistema, tendo assim mais de um ator para representá-lo. Um ator é representado por dois tipos de ícones: o de uma figura humana, para representar papéis desempenhados por usuários humanos e o de um computador, para representar papéis desempenhados por subsistemas ou outro tipo de software auxiliar.

Identificação de um conjunto de cenários de uso: Os cenários de uso são identificados a partir do documento de requisitos. O conjunto dos cenários de uso de um Diagrama de Interação dos Cenários deve cobrir as principais operações do sistema. As operações agrupadas em uma mesma cena estão relacionadas por terem um mesmo contexto de início e término, isto é, todas as operações de uma cena devem iniciar e terminar em um mesmo estado do sistema. Uma cena de uso é representada como

uma elipse contendo o nome da cena.

Identificação das associações entre atores e cenários de uso: Um ator interage com uma cena quando inicia ou participa de algum ponto da cena, fornecendo informações para o sistema ou recebendo informações. A participação de um ator em uma determinada cena é mostrada conectando-se o ícone do ator com o símbolo da cena de uso por um segmento de reta sólido. O estilo da interação é mostrado por pontas do segmento de reta: um segmento de reta com apenas uma seta na extremidade próxima à cena de uso retrata uma interação de um ator principal na cena; um segmento de reta com setas em ambas as extremidades retrata uma interação de um ator participante na cena.

Identificação de dependências entre cenários de uso: A dependência entre dois cenários é identificada quando existe um relacionamento de uso entre eles. Um relacionamento de uso de A para B indica que uma instância de uma cena de uso A também inclui o comportamento especificado em B. Um relacionamento tipo "usa" entre cenários de uso é mostrado por um segmento de reta tracejado. Na Figura 2 é ilustrado o Diagrama de Interação dos Cenários para o sistema SAPES.

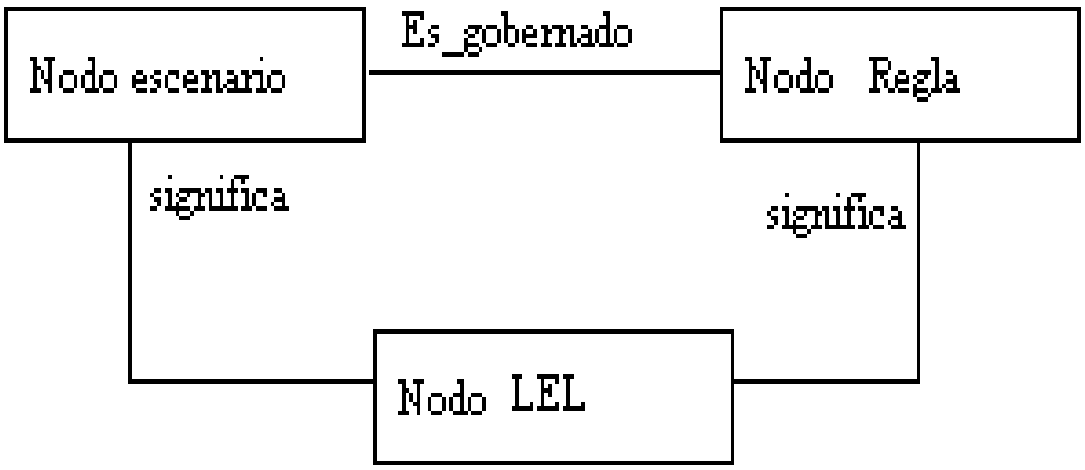

Figura 2 – Exemplo de um DIC

3.2 Glossário do diagrama de interação de cenários

No glossário do Diagrama de Interação dos Cenários, no qual é realizada a unificação inicial da terminologia do sistema, são descritos os cenários de uso identificados, os atores pertencentes ao domínio e seus atributos. Esse glossário é composto pelas colunas nome, tipo e descrição. O nome é um identificador alfanumérico único para o domínio do problema. O tipo identifica se é uma cena ou um ator. As descrições de cada elemento do glossário devem ser baseadas no documento de requisitos do sistema. O glossário do Diagrama de Interação dos Cenários construído para o sistema SAPES é mostrado na Figura 3.

3.3 Construção dos Diagramas Abstratos Detalhados (DAD)

Os Diagramas Abstratos Detalhados devem ser construídos a partir da visão de cada ator que participa de uma determinada cena, pois atores diferentes que participam de uma mesma cena provavelmente gerarão eventos diferentes, passando por estados diferentes. Por exemplo, uma cena A que contém a interação de dois atores, terá Diagramas Abstratos Detalhados diferentes, um para cada ator que com ela interage.

O ERACE-Tool define dois tipos de estados em um Diagrama Abstrato Detalhado, os estados intermediários, que representam operações do sistema que só podem ser executadas a partir de uma dada interação e os estados iniciais. Toda cena contém um estado inicial que representa o contexto de inicio da mesma; uma cena completa é composta por um conjunto de operações que iniciam no estado inicial e após um número qualquer de eventos retornam a ele. A semântica de um estado inicial é diferente da dos estados intermediários, pois ele não representa uma ação do sistema, mas sim uma pré-condição necessária para o início e término da cena.

Como já foi dito, uma mesma cena pode possuir vários Diagramas Abstratos Detalhados a ela relacionados, um para o seu ator principal e outros para cada ator participante. A única diferença existente entre os diagramas construídos para os dois

tipos de atores é que no caso do Diagrama Abstrato Detalhado do ator participante, o estado inicial e todos os estados em que o ator participante colabore devem estar relacionados a um estado intermediário presente no Diagrama Abstrato Detalhado do ator principal, visto que foi ele quem iniciou a interação da cena.

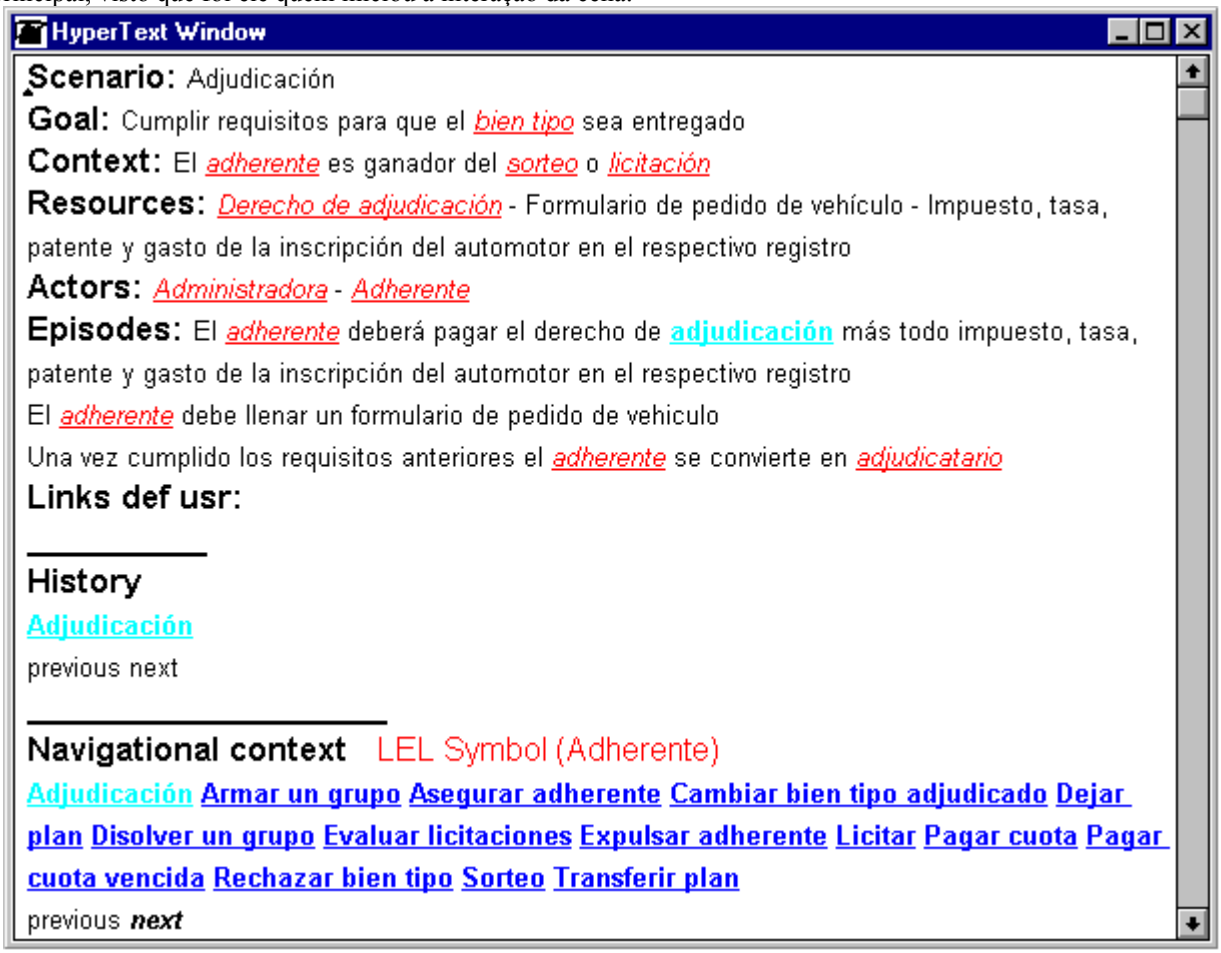

Figura 3 – Glossário do DIC

Existem dois tipos de fluxos de eventos em um Diagrama Abstrato Detalhado: um fluxo principal, que é composto de seqüências de eventos de uma operação realizada com sucesso dentro da cena e um fluxo alternativo, que são eventos que podem descrever algum tratamento de erros durante uma operação, interrupções realizadas durante uma operação ou qualquer tipo de evento que esteja fora do fluxo principal.

Cada Diagrama Abstrato Detalhado é composto por uma seqüência de estados e um conjunto de transições. Um estado intermediário é representado por um retângulo com os cantos arredondados contendo o nome do estado em seu centro. Os estados iniciais possuem notação um pouco diferente, apresentando bordas mais grossas que os intermediários. Um cena completa tem apenas um estado inicial, podendo ter *n* estados intermediários para várias operações detalhadas.

As transições são descritas como eventos que provocam a mudança do sistema para um determinado estado. Uma transição pode representar uma interação de um ator com o sistema, uma resposta do sistema a um ator ou um estímulo do sistema para a interação de um ator. Uma transição entre estados é mostrada conectando um estado a outro através de uma reta. As setas na ponta de cada reta indicam a seqüência e ocorrência dos estados. As transições do fluxo principal são representadas por retas sólidas, enquanto que as transições do fluxo secundário possuem retas tracejadas em sua representação. Em uma cena completa o número de transições principais que saem de um estado inicial deve ser o mesmo das que nele chegam. Na Figura 4 são detalhadas as operações existentes na cena "Preparação de Itens Bibliográfico". Todas as operações agrupadas na cena iniciam e terminam no mesmo estado, as transições entre os estados representam interações com sistema ou respostas fornecidas pela interface.

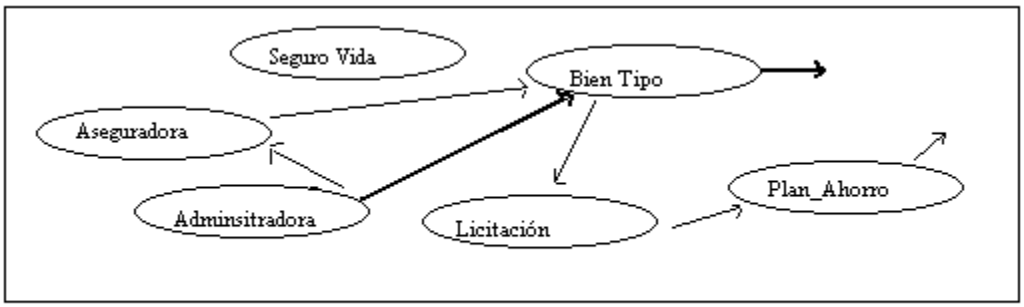

Figura 4 – DAD da cena "Preparação de Itens bibliográficos"

3.4 Formalização

A fase de formalização visa aplicar um base formal matemática a fase de modelagem dos cenários com o objetivo de obter cenários mais precisos, suscetíveis a mudanças e mantendo o usuário envolvido no processo [3]. Nesta fase são gerados cenários formais para cada um dos cenários detalhadas no Diagrama Abstrato Detalhado. Esses cenários têm a forma de pequenos programas que são gerados a partir das especificações presentes no Diagrama Abstrato Detalhado.

Um Diagrama Abstrato Detalhado é um diagrama baseado na notação das máquinas de estados finitos que representam uma visão V de uma determinada cena, sob o ponto de vista de um ator. Na fase de formalização, essa máquina de estados de V é convertida em um conjunto de sentenças de uma linguagem regular Lv, chamada **Linguagem de Detalhamento** (LD). A gramática de Lv, chamada de Gv, é uma gramática regular, definida por quádruplas (NT, T ,P,S), onde:

- NT é um conjunto finito chamado vocabulário não terminal que representa os estados do Diagrama Abstrato Detalhado para a visão V;
- T é um alfabeto finito, chamado vocabulário terminal que representa o conjunto de eventos do Diagrama Abstrato Detalhado;
- P é um conjunto finito de regras ou produções. Cada regra ou produção de P tem o formato A → B e as transições de A para B são resultado de um evento; e
- S é o axioma ou símbolo inicial de G. S é o nome da categoria sintática principal [2].

Na ERACE-Tool os diagramas abstratos de detalhados, após construídos, podem ser verificados por meio de um parser, um subsistema responsável pela etapa de análise. Cada parser de domínio é responsável pela análise de programas escritos em sua linguagem. A ERACE-Tool produz uma descrição textual do detalhamento, que é um pequeno programa escrito na linguagem gerada a partir da gramática de construção dos Diagramas Abstratos Detalhados, que é validado por um parser de domínio ERACE gerado na máquina Draco-Puc [4]. Na Figura 5 é mostrada a interface de verificação de um Diagrama Abstrato Detalhado.

A máquina Draco-Puc é um sistema de software que tem como objetivo principal implementar o paradigma Draco[5]. O paradigma está fundamentado no reuso de componentes, onde estes estão agrupados em domínios, os quais oferecem uma linguagem própria como interface. A máquina permite validar idéias do paradigma Draco, suportando não apenas a construção de domínios, mas também as principais etapas do processo de desenvolvimento de software, ao longo do seu ciclo de vida. A máquina Draco, possibilita a produção automática de um programa executável a partir da especificação na linguagem do domínio da aplicação. Dada uma aplicação, o processo para sua transformação é iniciado pela verificação de suas sintaxe pelo *parser* domínio. A ERACE-Tool utiliza as aplicações de análise do ambiente Draco, para validar os programas produzidos com base nos diagramas.

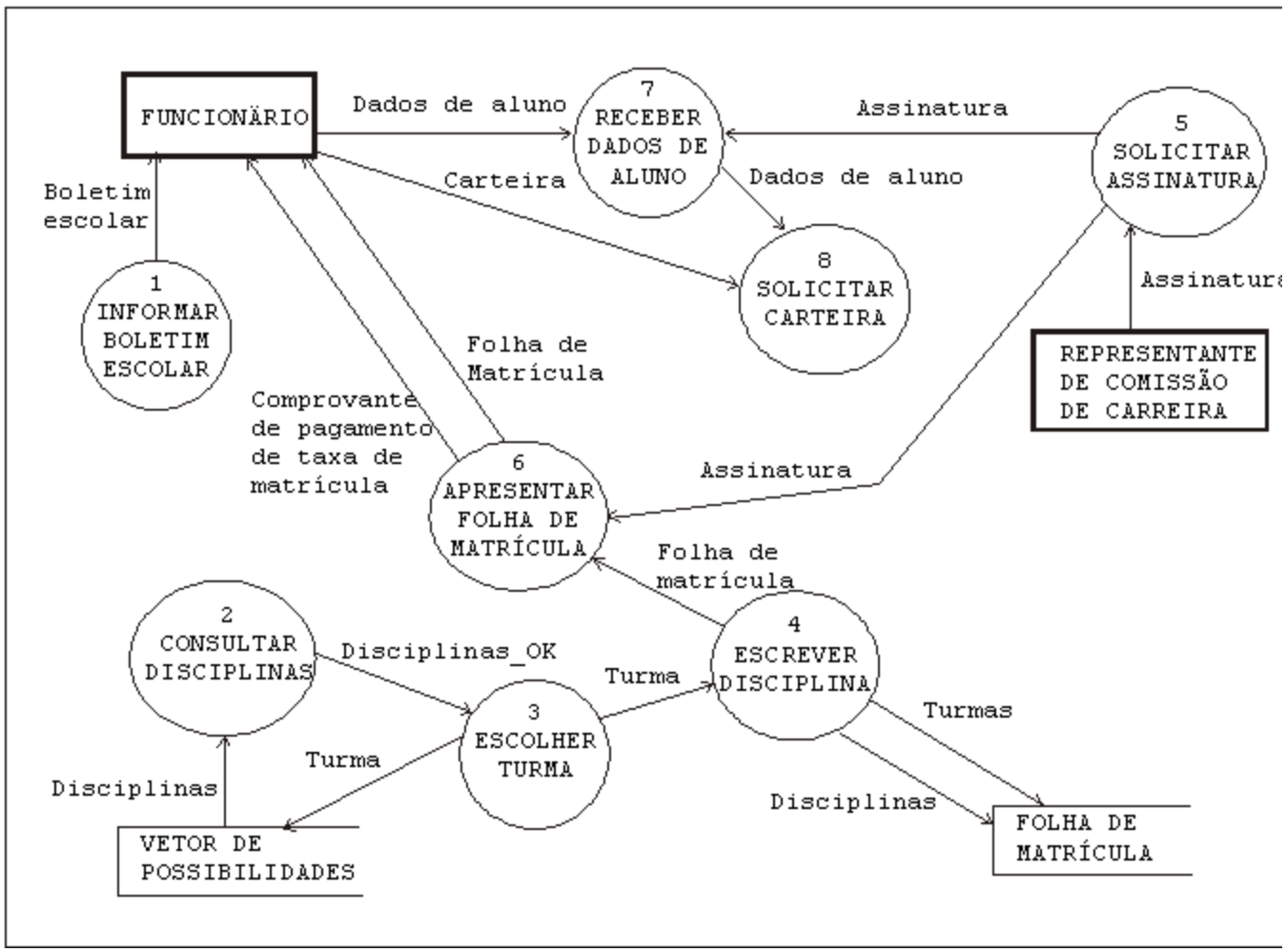

Figura 5 – Interface de verificação de um DAD

3.5 Extensão dos Glossários

Na fase de criação dos Diagramas Abstratos Detalhados o glossário que unifica a terminologia do sistema é estendido para descrever as novas informações especificadas nestes diagramas. Novos glossários que unificam as informações coletadas na fase de Modelagem dos Cenários são construídos com base nos Diagramas Abstratos Detalhados feitos para os vários cenários e atores do sistema. Cada Diagrama Abstrato Detalhado terá um glossário correspondente para descrever suas informações. Estes glossários são compostos pelas colunas nome, tipo, operação, argumento e descrição. O nome é o identificador alfanumérico de um elemento presente no Diagrama Abstrato Detalhado. O tipo identifica se é um estado ou uma transição. A operação identifica a qual seqüência de estados pertencentes à cena está relacionado aquele elemento do diagrama. Toda cena contém um conjunto de operações com o mesmo contexto de início e término que podem ser realizadas durante seu desenrolar. O argumento identifica os objetos do domínio que são usados durante a operação. A descrição contém um texto livre e sucinto com as informações que descrevem o contexto e os objetivos de um estado ou uma transição do diagrama

4. Passagem para o Modelo de Análise

Uma vez que os modelos de requisitos do processo ERACE tenham sido elaborados, pode-se iniciar a fase de passagem para um modelo de análise. Os diagramas presentes no processo ERACE não coincidentemente são bastante semelhantes aos diagramas de casos de uso e diagramas de estados descritos na *Unified Modeling Language* [1]. A notação aqui introduzida serve como processo que auxilia na construção dos modelos de análise da UML. Esses diagramas são utilizados para representar diversos aspectos da estrutura do sistema. Para a produção dos modelos de análise devem ser aplicados ao modelo de requisitos produzidos na ERACE-Tool um conjunto de heurísticas que ajudam a definir os componentes desses modelos:

a) Os Diagramas de Estados são construídos a partir dos Diagramas Abstratos Detalhados.

Os Diagramas de Estados da UML podem ser obtidos a partir dos Diagramas Abstratos Detalhados construídos para os cenários de uso identificadas no Diagrama de Interação dos Cenários.

b) Os Atores e os Argumentos são considerados candidatos a classes. Os métodos são as operações.

Baseado nos atores presentes no glossário do Diagrama de Interação dos Cenários e nos argumentos dos estados presentes no glossário do Diagrama Abstrato Detalhado são identificadas as classes candidatas do sistema. As operações presentes no glossário do Diagrama Abstrato Detalhado são indicativas dos métodos que compõem as classes. Praticamente todos os atores e argumentos podem dar origem a uma classe. Entretanto, para que seja incluído, o item deve estar relacionado a um conceito importante ao entendimento do domínio. Os métodos identificados com base nas operações são normalmente associados aos argumentos que mais aparecem no campo operação do glossário.

c) As associações e as cardinalidades em geral são obtidas nos glossários.

Com as informações registradas nos dois glossários construídos no processo ERACE são identificadas associações entre as classes candidatas. Cada associação está relacionada com os argumentos listados em cada cena. Consultas ao documento de requisitos e aos diagramas construídos também auxiliam na nomeação e na identificação das cardinalidades das associações. Se ao final da aplicação da terceira heurística restarem classes candidatas que não possuam associações, tais classes podem ser atributos de outras classes ou itens não relacionados a um conceito importante para o entendimento do domínio. Neste caso essa classe deverá ser excluída.

d) O número de cenários sugere o número de submodelos de análise para a representação.

As notações gráficas podem se tornar complicadas de entender caso os diagramas fiquem muito grandes e dispersos. A notação de um modelo de análise não é uma exceção a essa regra; para lidar com este problema, é possível dividir um diagrama em vários subdiagramas. A separação por assuntos pode ser utilizada para dividir um problema em partes coesas e independentes.

e) Construção do Diagrama de Classes do Sistema.

Com as informações coletadas por intermédio das heurísticas **b** e **c** é iniciada a construção do Diagrama de Classes do Sistema, delimitando os limites do sistema e separando as classes e as associações que não são necessárias para a funcionalidade do sistema. Essa delimitação é feita com a aplicação de estereótipos da notação UML às classes identificadas. Os estereótipos UML que podem ser utilizados são: de **ator**, substituem as classes candidatas que interagem com o sistema; de **fronteira**, usados para classes candidatas com função de comunicação ou manipulação (protocolos); de **entidade**, usados para modelar objetos de negócios e controle, usados para conectar objetos de fronteira com objetos de entidade e para manusear uma seqüência de operações dentro do sistema.

f) Diagramas de Seqüência e os Diagramas de Colaboração são construídos a partir das operações identificadas no glossário do DAD.

Os Diagramas de Seqüência são construídos a partir de cada operação presente nos glossários do Diagrama Abstrato Detalhado. Os objetos representados neste diagrama são identificados a partir das classes especificadas no Diagrama de Classes do Sistema construído com base na heurística **e**. As mensagens são obtidas nas transições do Diagrama Abstrato Detalhado da operação especificada. Os rótulos são obtidos nas descrições dos estados e transições da operação.

5. Conclusões

A construção da ERACE-Tool visou principalmente a criação de uma ferramenta que apoiasse a elicitação e modelagem de requisitos possibilitando ainda a verificação dos modelos criados através de formalismos. Muitas das ferramentas que atualmente apoiam métodos orientados a objetos utilizam notações para coleta de requisitos de forma não integrada, o que prejudica a verificação e validação dos requisitos durante o desenvolvimento de software.

A ERACE-Tool é um ambiente gráfico simples para construção dos diagramas descritos no processo ERACE para a fase de modelagem de requisitos. Todos os glossários criados durante a modelagem podem ser obtidos na forma de relatórios, o que fornece uma documentação para outras fases do sistema modelado. Além disso, a ERACE-Tool respeita a organização de todos os modelos do processo (diagramas e glossários), facilitando a verificação dos requisitos do sistema. A verificação dos cenários na fase de formalização impõe um maior rigor na criação dos modelos, permitindo avaliar a correção e completitude

de possíveis modificações nos requisitos. A verificação ainda possibilita a produção de cenários formais completos e fornece informações úteis em outras fases do ciclo de vida de desenvolvimento de software.

A busca da integração e compatibilidade entre os diagramas produzidos na ERACE-Tool e a notação da UML através das heurísticas de passagem, propiciou inúmeras vantagens na etapa de evolução dos requisitos para a construção do modelo de análise. Algumas dessas vantagens são: redução da atividade de construção do diagrama de classes, melhor apoio para a construção do diagramas de classes do sistema, do diagrama de estados, do diagrama de seqüência e do diagrama de colaboração. A construção da ferramenta proporcionou não somente uma interface para a construção dos modelos propostos no processo, mas um ambiente computacional para a verificação e integração destes vários modelos. Além disso, a ferramenta utiliza em sua atividade de verificação o sistema transformacional Draco-Puc, que abriu uma gama de possibilidades quanto à tradução entre o domínio construído para a ferramenta e outros linguagens de domínio. A integração com o Draco-Puc ainda garantiu uma série de informações adicionais que aumentam a formalidade da descrição dos cenários de uso, dando origem a documentos que podem ser utilizados na fase de teste de software.

Bibliografia

[1] Rational Software Corporation, Informações disponíveis via worldwide web: http://www.rational.com

[2] Rodrigues, S. *Introdução a Gramática e Linguagens*. Notas do ICMC, Série Computação.

[3] Turine, M.;Masiero P.C. *Especificação de Requisitos: Uma introdução*. Relatório técnico do ICMC, Série Computação, Março,1996.

[4] Prado, A. F. *Estratégia de Re-Engenharia de Software Orientada a Domínios*. PhD thesis, Pontifícia Universidade Católica do Rio de Janeiro, 1992

[5] Neighbors, M. J. *The Draco approach to constructing using components*. Phd thesis , University of California at Invine, 1980.

[6] Caldas, J. *Uma Ferramenta Baseada em Cenários para Elicitação e Modelagem de Requisitos*. Dissertação de Mestrado, ICMC-USP, Maio 1998