Requisitos de Hiperdocumentos de suporte ao domínio de Engenharia Reversa de Software

Valéria Delisandra Feltrim
Renata Pontin de M. Fortes
{vfeltrim, renata}@icmsc.sc.usp.br

Departamento de Ciências da Computação eEstatística
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo

Resumo

Este trabalho apresenta os requisitos funcionais identificados no processode Engenharia Reversa de Software que possam ser suportados por um SistemaHipertexto. Por meio da modelagem conceitual e navegacional do domíniode informações relativas ao método de engenharia reversaFusion-RE/I, foram estabelecidos os requisitos funcionais de um aplicativohipermídia de suporte ao método, de forma a nortear o engenheirode software responsável pelo processo de engenharia reversa e possibilitaro acompanhamento da evolução desse processo.

Palavras-Chave: Requisitos do Processo de Engenharia Reversade Software, Modelagem de Hiperdocumentos, Projeto de Hiperdocumentos.

Abstract

This paper presents the functional requirements of the reverse engineeringprocess in order to be supported by hypertext systems. These requirementswere defined by a conceptual and navigation modelling of the informationdomain related to a reverse engineering method called Fusion-RE/I. Thus,the software engineer responsable for the reverse engineering process hasthe specific guidelines to be follow and these guidelines can be used duringthe process evolution.

Keywords: Requirements of the Reverse Engineering Process, HyperdocumentsModelling, Hyperdocument Project.

1. Introdução

As atividades de engenharia de software em geral têm sido apoiadaspor ferramentas que buscam explorar os conhecimentos dos engenheiros desoftware de forma a possibilitar mais garantias de qualidade durante aexecução das atividades. Além disso, as ferramentastentam explorar novas tecnologias visando incorporar maior agilidade durantea realização do processo todo. A Engenharia de Requisitoscontribui muito neste sentido, pois permite fornecer as definiçõesque nortearão a construção de tais ferramentas demaneira flexível a possibilitar sua evolução [Staa97].

Notadamente, tanto a atividade de manutenção como o reusode componentes de software requerem uma análise do sistema alvoe uma recuperação do seu projeto para um melhor entendimentodo sistema por parte do engenheiro de software. As informaçõesrecuperadas através dessa atividade também servem como umafonte de informações à coleta de dados, uma das principaisatividades da fase de elicitação de requisitos do sistema.Tal tarefa de recuperação não é trivial, poismuitas vezes a documentação disponível é poucaou nenhuma, ou ainda desatualizada, e o código fonte nãoretrata explicitamente decisões tomadas durante a etapa de projetodo sistema. Nesse ponto, como auxílio às tarefas de recuperaçãodas informações de projeto e análise do sistema alvo,surgem os métodos e técnicas de engenharia reversa, comoferramenta à recuperação de informaçõesrelevantes e à uma compreensão adequada do sistema em si.

Em [Costa 97] encontramos engenharia reversa como "o processo de examee compreensão do sistema existente, para recapturar ou recriar osrequisitos atualmente implementados pelo sistema, apresentando-os em umgrau ou nível mais alto de abstração. Não envolvemudanças no sistema ou criação de um novo sistema".Ainda, engenharia reversa de software é um processo de examinação,não um processo de mudança ou duplicação [Chikofsky90].

Em termos comparativos, tendo definido engenharia progressiva como umprocesso que inicia-se na análise dos requerimentos, passa peloprojeto até a implementação do sistema, a engenhariareversa é vista como o oposto da engenharia progressiva.

Figura 1. Processo de Eng. Progressiva x Eng. Reversa

O processo de engenharia progressiva é caracterizado por umaseqüência de atividades que podem ser descritas atravésde um modelo de ciclo de vida. Na literatura encontramos diferentes formalizaçõesdesses modelos, sendo que embora todos passem por atividades básicasque apresentam-se em todas as formalizações, a existênciade interações entre as partes de cada modelo diferem, bemcomo a existência de atividades e produtos específicos a cadaformalização.

Na engenharia progressiva, o sistema é o resultado do processode desenvolvimento. Na engenharia reversa, o sistema geralmente éo ponto inicial do processo [Chikofsky 90]. A Figura 1, adaptada de [Chikofsky90], representa esquematicamente, uma forma comparativa das direçõesinversas por quais se orientam as etapas envolvidas em cada uma das engenharias.O processo de engenharia reversa caracteriza-se pelas atividades retroativasdo ciclo de vida, que partem de um baixo nível de abstraçãopara um alto nível de abstração.

Embora exista a caracterização de um processo para a engenhariareversa, não existe a formalização de um "modelo deciclo de vida" para as atividades envolvidas na engenharia reversa de software.Existem, por outro lado, métodos de engenharia reversa que impõemuma ordem na execução das atividades propostas por cada método,bem como estabelecem os produtos resultantes da realizaçãode tais atividades.

Neste contexto, este trabalho apresenta os requisitos funcionais identificadosno processo de engenharia reversa que possam ser suportados por um sistemahipertexto. Dessa forma, os requisitos especificados podem ser qualificadoscomo requisitos internos do sistema alvo, pois neste trabalho érealizado um processo de *feedback* para a obtençãodo suporte de hiperdocumento ao processo de engenharia reversa. Ou seja,para obtermos os resultados da Engenharia reversa de um sistema hipertexto,estaremos usando o próprio sistema hipertexto para registrar o processo.Sendo assim, foi realizada a modelagem do domínio de informaçõesde um método de engenharia reversa, resultando em um projeto deaplicação hipertexto (cujo hiperdocumento representa o domínioda aplicação [Isakowitz 95]) de apoio às atividadesdesenvolvidas durante o processo; trata-se justamente de uma tentativade formalizar o relacionamento dessas atividades, seus subprodutos e interações.Para tal, nos apoiamos no método de engenharia reversa Fusion-RE/I[Costa 97], base para o estabelecimento das atividades envolvidas e dosmodelos gerados ao final do processo.

Para essa modelagem foi adotado o método OOHDM – *Object OrientedHypermedia Design Methodology*, um método para modelagem de aplicaçõeshipermídia orientado a objetos. Uma vez que o método de engenhariareversa utilizado, o Fusion-RE/I, também é orientado a objetos,a utilização do OOHDM se apresenta mais natural. Deve-seressaltar também que os requisitos funcionais aqui apresentadospela modelagem serão o ponto de partida para a realizaçãodo exercício de engenharia reversa sobre o ambiente hipertexto SASHE- *S*istema de *A*utoria e *S*uporte *H*ipermídiapara *E*nsino [Nunes 97], uma vez que o processo a se realizar seráuma instância do referido modelo proposto.

Na próxima seção são discutidas as atividadesanteriores a modelagem, dentro do processo de engenharia de requisitos,para a aplicação hipertexto proposta. Na Seção3 é apresentada a modelagem OOHDM do domínio do processode engenharia reversa de software segundo o método Fusion-RE/I.Finalmente na Seção 4 são comentados as conclusõese os próximos passos a serem continuados nesta pesquisa.

2. Engenharia de Requisitos do Pontode Vista de Hipertextos

A Engenharia de Requisitos estabelece a definição de requisitoscomo um processo no qual o que deve ser feito pelo sistema alvo éelicitado, modelado e analisado. Tal processo deve lidar com diferentespontos de vista, e usar uma combinação de

métodos,ferramentas e pessoal [Leite 90].

Na fase de elicitação, o engenheiro procura capturar osrequisitos do software, buscando obter conhecimento do domínio doproblema [Turine 96]. Em particular, para este trabalho, por se basearem um sistema hipertexto existente, para suportar a engenharia reversado próprio sistema hipertexto, muitos dos requisitos do sistemaproposto já encontravam-se pré-estabelecidos; uma vez queas características inerentes ao próprio sistema hipertextosão características desejáveis ao novo sistema. Assimtambém, os aspectos de usabilidade, desempenho e interface, comorequisitos do processo de engenharia reversa com o suporte de hipertextos,já estão pré-definidos, pois já é pressupostaa utilização do sistema hipertexto SASHE já existente.Quanto ao processo de engenharia reversa propriamente dito, os requisitosreferentes a interface, são apresentados na forma do modelo contextual(Figura 5), que mostra a navegação pelos documentos a seremmanipulados.

Como a tecnologia de sistemas hipertexto oferece um grau de facilidadesignificativo para o usuário de sistemas que visam a busca e o fornecimentode informações, as premissas de interface simples e quaseintuitiva, bem como a liberdade de escolha na busca de informaçõessão requisitos necessários de qualquer sistema aplicativoque considere os hipertextos para suporte [Fortes 96].

O hipertexto em questão deve suportar documentos resultantesda aplicação do método de engenharia reversa Fusion-RE/I.Dessa forma, um estudo detalhado do método foi realizado, semprena tentativa de visualizar as ligações existentes entre osdiversos documentos, para que a representação efetiva dessasligações dentro do hiperdocumento viesse auxiliar o engenheirode software na realização da engenharia reversa.

Sendo assim, este artigo visa apresentar o produto resultante da fasede modelagem dos requisitos de um sistema hipertexto de apoio àengenharia reversa de software, de forma a representar o domínioconceitual das informações com o qual o software trabalhará.Por tratar-se de um hiperdocumento, optamos por um método de modelagemconceitual voltado a aplicações hipermídia, que apresentasseos recursos necessários para tal modelagem.

Desta forma, a modelagem conceitual das informações produzidasdurante o processo de engenharia reversa, por meio de uma aplicaçãohipermídia, visa atender as necessidades de evoluçãodos requisitos funcionais que o processo requer. Com o suporte de um hiperdocumentode estrutura previamente definida, com o conjunto de nós e linksreferentes ao domínio de engenharia reversa, tem-se um guia dosrequisitos sobre o qual o engenheiro de software deve se apoiar duranteo processo de recuperação de informações. Assim,esse engenheiro de software atua também como autor do hiperdocumentoque representa as informações recuperadas no processo deengenharia reversa. Os pontos de vista então devem ser conciliadosno hiperdocumento a ser elaborado que se configura como um espaçode informações dos requisitos. A Figura 2 ilustra a etapade modelagem conceitual do domínio do processo de engenharia reversa,contribuindo para a obtenção dos requisitos a serem incorporadosno hiperdocumento. A área pontilhada da figura apresenta de formasimplificada o cenário de interação do engenheirode software com o sistema hipertexto (SASHE, no caso), e a modelagem conceitualdo método de engenharia reversa que estará sendo registradano hiperdocumento.

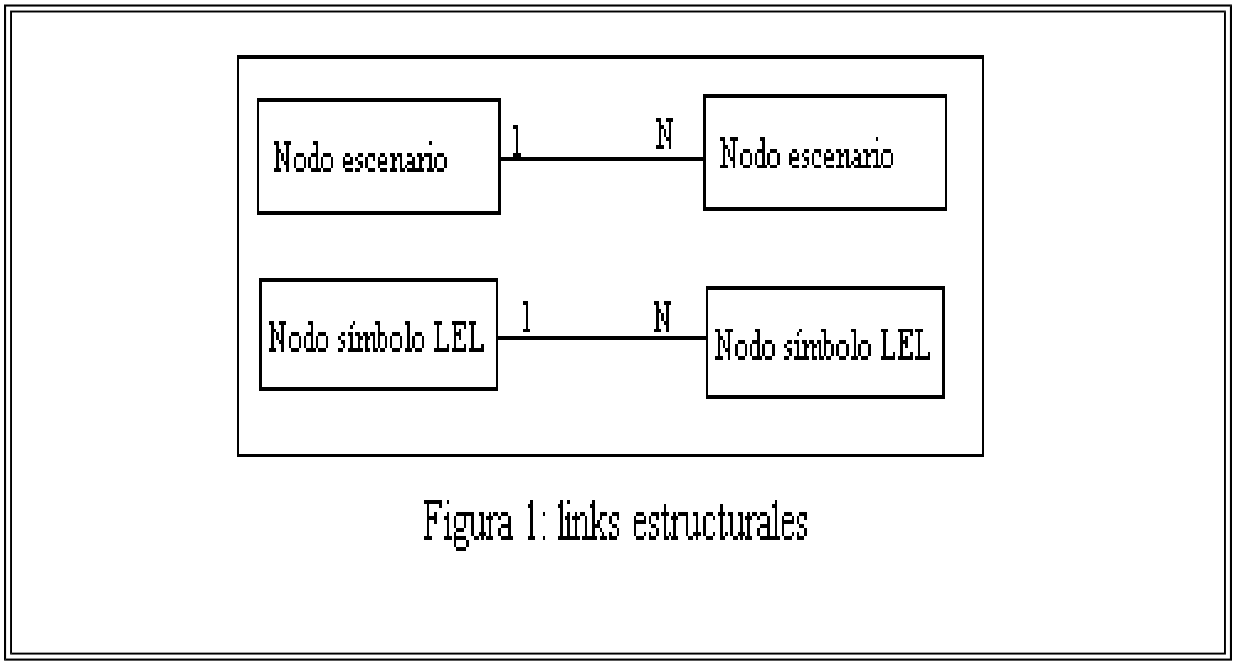

Figura 1: links estructurales

Figura 2. Elicitação de requisitos funcionais dehiperdocumento de apoio à engenharia reversa

De fato, a partir do entendimento dos requisitos inerentes aos sistemashipertexto mais a modelagem conceitual do hiperdocumento temos o quadrogeral de requisitos funcionais internos ao sistema final, que orientarãoo processo de apoio a engenharia reversa por meio de hiperdocumentos.

A modelagem de um hiperdocumento tem o objetivo de organizar de formacoerente as estruturas do hiperdocumento, ou seja, as ligaçõesentre os nós, de forma a permitir um melhor entendimento pelo leitordo hiperespaço planejado pelo autor e auxiliar o autor no própriodesenvolvimento do hiperdocumento.

Os modelos encontrados na literatura que se apresentam como principaisreferências para a modelagem conceitual de um hiperdocumento, comrecursos para definição de esquemas conceituais de dados,são: HDM *(Hypertext Design Model)* [Garzotto 93], RMM *(RelationshipManagement Methodology)* [Isakowitz 95], EORM *(Enhanced Object RelationshipModel)* [Lange 94] e OOHDM *(Object Oriented Hypermedia Design Methodology)*[Rossi 96].

O OOHDM é um método voltado para o desenvolvimento deaplicações hipermídia que descreve as tarefas a seremexecutadas desde a análise de domínio da aplicaçãoaté a sua implementação [Schwabe 96]. Ele éum descendente direto do HDM [Garzotto 93], incorporando uma sériede novos conceitos vindos sobretudo do paradigma de orientaçãoa objetos. A metodologia propõe que o desenvolvimento de aplicaçõeshipermídia seja um processo dividido em quatro etapas: modelagemconceitual, modelagem da navegação, projeto abstrato da interfacee implementação.

3. Modelagem do Domíniode Processo de Engenharia Reversa de Software

Efetivamente, o processo de engenharia reversa se constitui de procedimentossobre um software acompanhados da elaboração de diversosdocumentos que possibilitem a restauração de uma representaçãodesse software em um nível de abstração maior. Dessaforma, a modelagem dos requisitos funcionais de tais documentos, que constituemo hiperdocumento alvo são apresentados pela modelagem OOHDM do métodoFusion-RE/I nas subseções seguintes.

3.1. Esquema Conceitual

O modelo de classes do processo de engenharia reversa (PER) foi derivadodo estudo desse processo sob a perspectiva do método de engenhariareversa Fusion-RE/I, segundo demonstrado na Figura 3. Os elementos presentesno modelo podem ser agrupados em duas hierarquias principais: uma hierarquiade tarefas de engenharia reversa e uma hierarquia de produtos gerados.

As classes *Pessoa* e *Tarefa de Engenharia Reversa* (TER)definem a parte do modelo correspondente ao gerenciamento de projeto. ComoTER corresponde a atividade de recuperação de informaçõesrelevantes à geração de uma visão do sistema,por isso o relacionamento um-para-um entre TER e *Visão*. *Pessoa*está relacionada com TER apesar dessa tarefa poder ser vista comomais de uma atividade. Na modelagem, entretanto, foi considerado TER comouma tarefa que pode ser desenvolvida por uma única pessoa e nãoexiste a necessidade de se controlar a execução da TER pormais de uma pessoa, daí o relacionamento muitos-para-um entre TERe *Pessoa*. São as TER que conectam a parte de projeto e osprodutos gerados. Uma TER pode ser uma: *Consulta a Documento Existente*,*Análise da Interface*, ou *Análise do CódigoFonte*. Essas tarefas são descritas pelo método Fusion-RE/I,sendo que, segundo o método, apresentam uma escala de prioridadepara a realização de cada uma delas. No decorrer do processo,no entanto, essa prioridade pode mudar de acordo com as necessidades doengenheiro reverso, e certas tarefas podem ser novamente realizadas.

Uma *Visão* define uma interpretação do sistemasob engenharia reversa. Essa interpretação corresponde aum nível de abstração, sendo que pode ser estrutural,funcional ou de domínio. O Fusion-RE/I propõe recuperaçãode visões tanto ao nível estrutural como funcional. Assim,a representação de um sistema é composta por um conjuntode visões, e isso é expresso pelo relacionamento um-para-muitosentre *Software* e *Visão*.

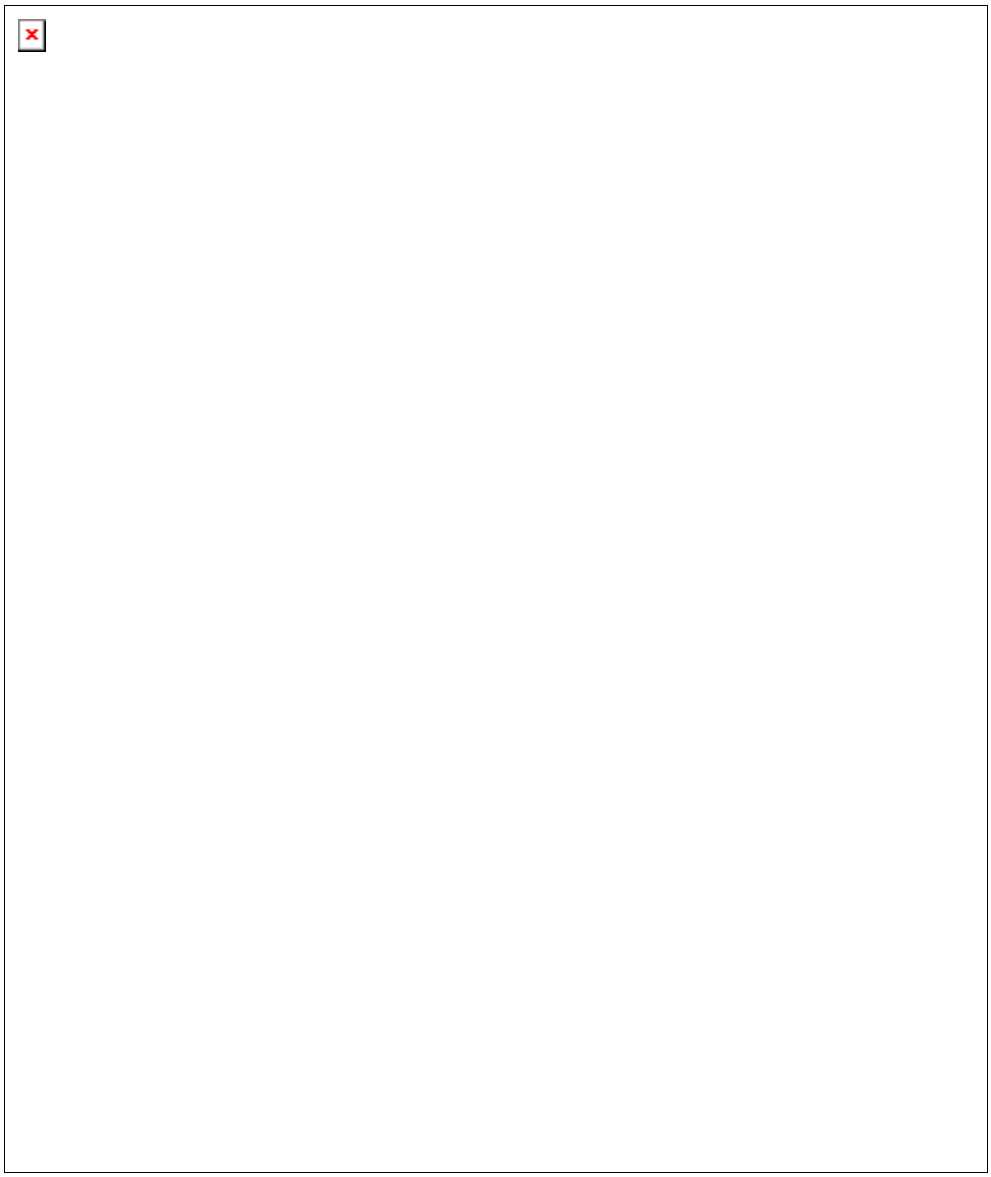

Figura 3. Modelo Conceitual do PER

Uma *Visão* é um agregado de documentos e dos relacionamentosexistentes entre esses documentos. Um documento é um *DocumentoExterno* que descreve informações recuperadas por umaTER. *Relacionamento com outros Docs* descreve uma ligaçãoentre um documento e seus documentos relacionados.

Documento Externo é um dos modelos ou quadros propostoscomo resultado do Fusion-RE/I. No esquema conceitual (modelo de classes),todos os documentos são implicitamente relacionados atravésde seus atributos. As operações contidas no *Modelo deCiclo de Vida* são as mesmas que compõem o *Modelo deOperações*. Da mesma forma, cada operaçãodescrita em *Operação* e constituinte do *Modelo deOperações* e é parte de um *Objeto*. *Modelode Objetos* é um agregado de objetos relacionados sendo que oatributo *Tema* indica o tema que inspirou o *Modelo de Objetos*em questão. O *Quadro de Operações-Procedimentosde Implementação* é um agregado de *Item*,onde cada *Item* compõe uma entrada do quadro. As operaçõescitadas nesse modelo também são as mesmas descritas nos modelosmencionados anteriormente. Da mesma forma, os procedimentos presentes emcada *Item* também são descritos em *Procedimento*.A agregação de *Procedimento* compõe os *Quadrosde Chamadas*, cujo atributo *arquivo* indica o arquivo onde seencontra o código que implementa *Procedimento*. O relacionamentoexplícito entre *Operação* e *Procedimento*se faz necessário uma vez que, apesar de existir o relacionamento (uma operação é implementada por um procedimento),não existem atributos relacionados. O relacionamento um-para-muitosse explica por uma operação ser implementada por um ou maisprocedimentos.

3.2. Esquema de ClassesNavegacionais

Como pode ser visualizado na Figura 4, o esquema de classes navegacionaisé derivado do esquema conceitual descrito anteriormente, e écomposto pelas classes que serão efetivamente navegadas, ou seja,serão visualizadas pelo usuário do aplicativo.

Dentro do processo descrito anteriormente, as classes navegáveisserão os produtos gerados pelas TER (modelos e quadros). Com exceçãodo *Modelo de Ciclo de Vida* e do *Modelo de Objetos*, todosos outros produtos são uma agregação de outros objetos.Dessa forma, as classes que compõem a agregação éque serão navegados, e não a classe agregada.

Assim, as classes componentes do esquema navegacional são *Operação*,*Objeto*, *Procedimento*, *Item*, *Modelo de Objetos*e *Modelo de Ciclo de Vida*. Nesse esquema tornam-se explícitosos relacionamentos que no esquema conceitual estão implícitospor meio dos atributos contidos nas classes. Esses relacionamentos sãoexpressos pelas setas rotuladas. O direcionamento das setas foi decididoconsiderando os objetos *Operação*, *Objeto*, *Procedimento*,*Modelo de Ciclo de Vida* e *Modelo de Objetos* como possíveisentradas para o início da navegação. A partir dissoos outros objetos foram relacionados de forma que todos pudessem ser alcançadose os relacionamentos fossem coerentes.

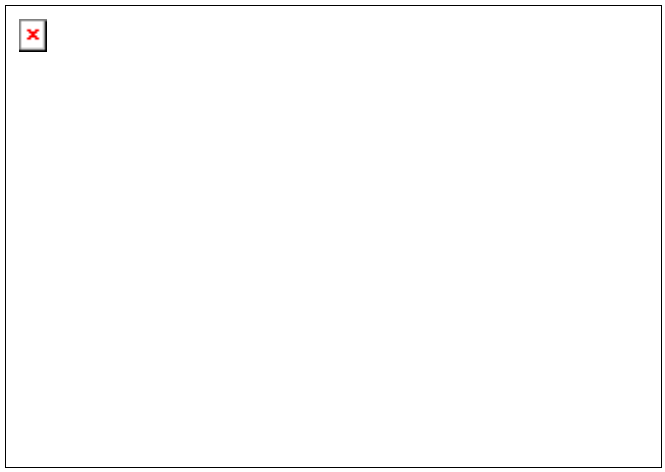

Figura 4. Modelo de Classes Navegacionais

3.3. Esquema Contextual

Os contextos navegacionais são geralmente induzidos pelas classesnavegacionais e expressam a estrutura navegacional geral da aplicação,enquanto as classe navegacionais especificam os objetos que serãovistos pelo usuário [Rossi 96].

Os contextos são um conjunto de instancias relacionadas, como,por exemplo, todos procedimentos em um determinado arquivo, todos os procedimentoscom uma determinada data, etc. No esquema contextual as classes navegáveissão mostradas juntamente com os contextos em podem aparecer na aplicação.Os contextos são representados por caixas pontilhadas que aparecemdentro da classe a qual o contexto pertence. Para simplificaçãodo diagrama, os contextos podem ser agrupados em quadros pontilhados maioresquando conveniente. As setas indicam navegação ou mudançade contexto. Quando uma seta chega a um agrupamento de contexto, isto indicaque qualquer um dos contextos pertencentes ao agrupamento estarádisponível para a navegação, sendo que os contextosda classe que não estiverem no agrupamento não serãovisíveis. Os índices também aparecem com caixas pontilhadasentre as setas de navegação.

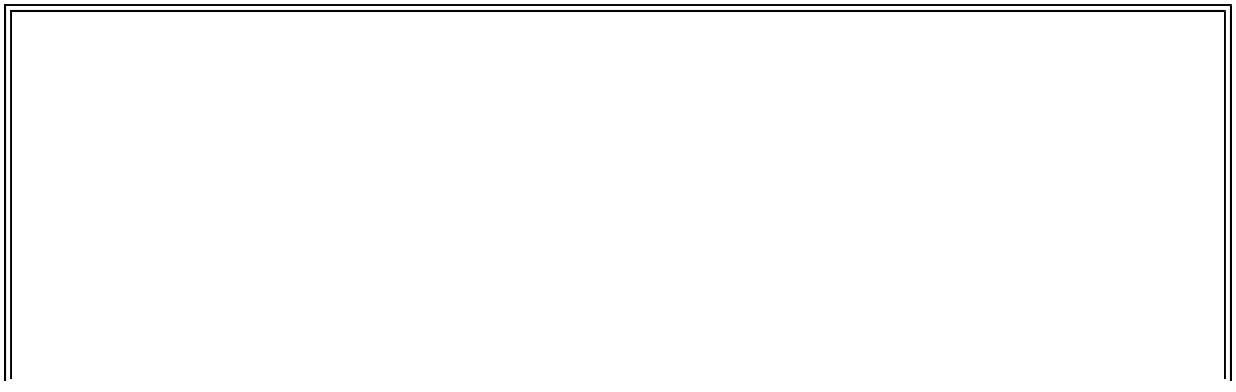

Figura 5. Modelo Contextual

No esquema contextual apresentado na Figura 5, o índice àesquerda é o principal, a partir do qual podem ser acessados diversosíndices intermediários. Cada classe pode então sernavegada nos contextos mostrados na figura. Um exemplo de navegaçãoem diferentes contextos pode ser visualizado na classe *Procedimento*.Uma vez que os procedimentos são acessados através do índicede procedimentos, cada instância da classe *Procedimento* podeser navegada tanto no contexto *por data* como no contexto *porarquivo*. No entanto, quando a classe *Procedimento* é acessadaatravés da classe *Operação*, a classe *Procedimento*passa a poder ser navegada no contexto *por operação*,uma vez que uma operação pode ser implementada por mais deum procedimento. A partir desse contexto, a classe *Procedimento*pode ser navegada também nos contextos *por data* e *porarquivo*. Dessa forma, o contexto *por operação*só é visível quando a classe é acessada atravésda classe *Operação*, e não aparece quando aclasse é acessada através do índice.

A navegação mostrada nesse esquema é a mesma doesquema de classes navegáveis, porém em maior nívelde detalhamento. Neste diagrama são visualizados os contextos emque cada classe pode ser navegada.

4. Conclusões

Neste trabalho foi apresentada uma modelagem conceitual e navegacionaldo domínio de Engenharia Reversa de Software utilizando o métododo projeto de aplicações hipermídia OOHDM [Rossi 96].Tal modelagem foi utilizada como forma de elicitação dosrequisitos funcionais do processo de engenharia reversa suportado por umhiperdocumento. A modelagem conceitual foi representada por um diagramade classes e a modelagem navegacional foi composta pelo diagrama navegacional,contendo as classes do domínio que serão efetivamente navegadas,e pelo diagrama contextual, que especifica o contexto de navegaçãodessas classes.

Para tal modelagem, nos apoiamos nas etapas descritas pelo métododo engenharia reversa Fusion-RE/I [Costa 97], um método para recuperaçãode projeto orientado a objetos desenvolvido no ICMC-SC. Por ambos os métodosserem orientados a objetos, a escolha do OOHDM nos pareceu mais adequada.De fato pode-se observar que apesar do domínio de informaçõesem questão ser abstrato, por meio da abordagem de orientaçãoa objetos os resultados da modelagem se mostraram representativos dos principaisdocumentos produzidos pelo Fusion-RE/I, bem como seus relacionamentos.

Na linha dos próximos passos desta pesquisa, esta modelagem serviráde apoio ao projeto de adequação do ambiente protótipoSASHE ao domínio da engenharia reversa de software. Dessa forma,o modelos apresentados serão validados em relaçãoaos documentos gerados durante o processo de engenharia reversa, pois osmesmos serão utilizados na manutenção do SASHE, poroutros engenheiros de software.

Referências Bibliográficas

[Chikofsky 90] Chikofsky, E.J.; Cross II, J.H. *ReverseEngeneering and Design Recovery: A Taxonomy*. IEEE Software, v.7, n.1,p. 13-7, 1990.

[Costa 97] Costa, Rejane M. *Um método de EngenhariaReversa para Auxiliar a Manutenção de Software*. Dissertaçãode mestrado, ICMC–USP, São Carlos-SP, 1997.

[Fortes 96] Fortes, R.P.M. *Análise e Avaliaçãode Hiperdocumentos: Uma Abordagem Baseada na RepresentaçãoEstrutural*. Tese de Doutorado, IFSC-USP, São Carlos-SP, 1996.

[Garzotto 93] Garzotto, F.; Paolini, P.; Schwabe, D. *HDM- A Model-Based Approach to Hypertext Application Design*. ACM Transactionson Information Systems, v.11, n.1, p.1-26, January 1993.

[Lange 94] Lange, D. B. *An Object-oriented design methodfor hypermedia information systems*. In Proc. of the 27thAnnual Hawaii Int. Conf. on System Sciences, p. 366-375,1994.

[Lehman 96] Lehman, M. M. *Process Improvement - TheWay Forward*. Simpósio Brasileiro de Engenharia de Software.São Carlos, 1996.

[Leite 90] Leite, J.C.S.P. *Validaçãode requisitos: o uso de pontos de vista.* Revista Brasileira de Computação,v.6, n.2, RBC, outubro/dezembro, p.39-52, 1990.

[Nunes 97] Nunes, M.G.V.; Hasegawa, R.; Vieira, F.M.C.;Santos, G.H.R.; Fortes, R.P.M.. *SASHE: Sistema de Autoria e SuporteHipermídia para Ensino*. Notas, n.33, ICMC–USP, São Carlos-SP,1997.

[Isakowitz 95] Isakowitz, T.; Stohr, E.A., Balasubramanian,P. *RMM: A Methodology for Structured Hypermedia Design*. Communicationsof the ACM, v.38, n.8, p.34-44, August 1995.

[Rossi 96] Rossi, G. *Um método orientado a objetospara o projeto de aplicações hipermídia*. Tesede Doutorado. Departamento de Informática Pontifícia UniversidadeCatólica, Rio de Janeiro, 1996.

[Staa 97] Staa, A. V.; Serrano, M. A. B. *Elicitaçãodos Requisitos de Meta-Ambientes de Engenharia de Software atravésda Análise de Negócios*. XI Simpósio Brasileirode Engenharia de Software, *Anais.* Fortaleza, 1997, p. 33-48.

[Schwabe 96] Schwabe, D.; Rossi, G.; Barbosa, S.D.J. *SystematicHypermedia Application Design with OOHDM*. In: Hypertext'96, WashingtonDC, USA, March 1996. *Proceedings.* New York, ACM Press, 1996.p.116-28.

[Turine 96] Turine, M.A.S.; Masiero, P.C. *Especificaçãode Requisitos: uma introdução.* Relatório Técnico,n.39, ICMC - USP, São Carlos - SP, 1996.

[Wang 92] Wang, B.; Hitchcock, P.; Holden, T. *An ObjectOriented Database Approach for Supporting Hypertext*. LNCS, p.601-28,1992.