

JULIANA PILATIS CURY

UTILIZAÇÃO DO MÉTODO *FEATURE-ORIENTED DOMAIN ANALYSIS*
NA IMPLANTAÇÃO DE PRODUTOS DE SOFTWARE EM
TELECOMUNICAÇÕES

Monografia apresentada à Escola
Politécnica da Universidade de São Paulo
para conclusão do curso MBA em Gestão
de Tecnologia de Informação.

Área de concentração: Engenharia de
Software.

Orientador: Professor Doutor Paulo Sérgio
Muniz Silva

São Paulo

2009

MBA/TJ
2008
C-949 u

DEDALUS - Acervo - EPEL



31500020871

FICHA CATALOGRÁFICA

Cury, Juliana Pilatis

**Utilização do método Feature-Oriented Domain Analysis na
implantação de produtos de software em telecomunicações /
J.P. Cury. -- São Paulo, 2009.**

61 p.

**Monografia (MBA em Tecnologia da Informação) - Escola
Politécnica da Universidade de São Paulo. Programa de Educa-
ção Continuada em Engenharia.**

**1. Engenharia 2. Engenharia de software 3. Reúso de
software I. Universidade de São Paulo. Escola Politécnica. Pro-
grama de Educação Continua- da em Engenharia II. t.**

AGRADECIMENTOS

Ao Professor Paulo Sérgio Muniz Silva pelo tempo despendido durante a produção deste texto, pela orientação e pelo constante estímulo em todas as etapas do trabalho.

Aos amigos e professores do PECE que durante os dois anos cursados sempre se mostraram prestativos e atenciosos, auxiliando a enriquecer meu conhecimento pessoal e profissional, e a todos que colaboraram direta ou indiretamente na realização deste trabalho.

RESUMO

Este trabalho descreve a aplicação de um método de elicitação de requisitos em projetos de implantação de um produto de software de mediação na área de Telecomunicações, com o propósito de estudar o reuso de requisitos na implantação de personalizações do produto. Dos métodos de elicitação de requisitos descritos, adota-se o *Feature-Oriented Domain Analysis*, um método maduro e cujo enfoque trata explicitamente o reuso de requisitos e aplica-se à implantação de produtos de software. No trabalho, são empregados apenas dois domínios definidos pelo método *Feature-Oriented Domain Analysis* para a identificação dos requisitos potencialmente reutilizáveis em um fragmento de projeto de implantação do produto de software de mediação em questão: análise de contexto e modelagem de domínio.

Palavras-chave: Elicitação de Requisitos. Reuso de Requisitos. Telecomunicações. Software. Implantação de Software

ABSTRACT

This work describes the application of a requirement elicitation method in a project implementation of a mediation software product in the Telecommunication field, with the proposition of study reuse of requirements on product personalization implementation. Above all studied methods in this work it is adopted the Featured-Oriented Domains Analysis method, a mature method whereas the main focus explicitly deals with reuse of requirements and it is applied to software product implementation. In this work, two domains of the Featured-Oriented Domains Analysis method are applied with the purpose of indentify potential reusable requirements in an element of a mediation software product implementation project: context analysis and modeling domain.

Key words: Requirements Elicitation. Telecommunication. Software. Software Implementation.

LISTA DE ILUSTRAÇÕES

Figura 1 - Modelo paralelo do método de processo de requisitos	5
Figura 2 - <i>Misuse</i> representado como um caso de uso ao inverso	10
Figura 3 - Visão dos produtos na Análise de Domínio.....	15
Figura 4 - Fases e produtos da análise de domínio.....	16
Figura 5 - Visão geral de um sistema de mediação.....	20
Figura 6 - Representação da utilização de re-uso de requisitos.....	22
Figura 7 - Sistema Externo	24
Figura 8 - Visão geral do software de mediação	25
Figura 9 - Fases e produtos da análise de domínio.....	28
Figura 10 - Diagrama de contexto	29
Figura 11 - Diagrama de relacionamento de entidades.....	30
Figura 12 - Diagrama de características portal de entrada.....	31
Figura 13 - Diagrama de características rota de coleta	32
Figura 15 – Diagrama de características plug-in	39

LISTA DE TABELAS

Tabela 1 – Mapeamento de problemas comuns durante levantamento de requisitos...	6
Tabela 2 – Catálogo de características tipo de dado versus Central Telefônica	34
Tabela 3 – Dicionário de terminologia	37
Tabela 4 – Catálogo de características tipos de dados e sistemas de upstream Projeto X.....	40
Tabela 5 – Catálogo de características tipos de dados e sistemas de upstream Projeto Y	40

LISTA DE ABREVIATURAS E SIGLAS

ARM	<i>Accelerated Requirements Method</i>
CORE	<i>Controlled Requirements Expression</i>
FODA	<i>Feature-Oriented Domain Analysis</i>
UDR	<i>Usage Detail Records</i>
CDR	<i>Call Detail Record</i>
ERB	Estação Rádio Base

SUMÁRIO

1. Introdução.....	1
1.1 Motivação	1
1.2 Objetivo.....	2
1.3 Organização da monografia	2
2. Elicitação e métodos de elicitação de requisitos.....	4
2.1 Elicitação e Análise de Requisitos.....	4
2.2 Métodos para Elicitação de Requisitos.....	8
2.2.1 Controlled Requirements Expression.....	8
2.2.2 Misuse Cases	9
2.2.3 Accelerated Requirements Method	12
2.2.4 Feature-Oriented Domain Analysis	13
2.3 Reuso de Requisitos.....	16
3. O software mediador e suas funções.....	19
3.1 Descrição geral.....	19
3.2 Necessidades do reuso de requisitos na implantação de produtos de softwares em mediação	21
4. Aplicação do Método FODA em um Modelo Conceitual da Implantação do Software de Mediação	23
4.1 Fluxo de dados do modelo conceitual da implantação	23
4.2 Aplicação do método FODA na implantação do software de mediação	26
4.2.1 Informações sobre o domínio	27
4.2.2 Análise de Contexto.....	28
4.2.3 Modelagem de domínio	29
4.2.4 Modelo de características	30
4.2.5 Modelo Funcional	35
4.2.6 Dicionário de terminologia do domínio.....	36
4.3 Reuso de requisitos na implantação	38
5. Considerações Finais	42
Referências Bibliográficas	43
Referências consultadas.....	44

APÊNDICE A – Atributos das entidades	45
--	----

1. Introdução

1.1 Motivação

A evolução da indústria de software nos últimos tempos trouxe impacto direto na evolução das diversas áreas envolvidas na sua produção, aprimorando ferramentas de apoio para o seu desenvolvimento para atender ao crescente mercado de software e a demanda pela agilidade na entrega destes produtos. Ao longo dessa evolução criaram-se métodos para o auxílio ao desenvolvimento e implantação de projetos de software, de maneira a aperfeiçoar as técnicas para implantação destes produtos. Tudo isso influenciou de maneira definitiva o modo em que os novos projetos de software passaram a ser desenvolvidos e implantados.

Com o aumento da competitividade, a exigência cada vez mais rigorosa dos compradores e os prazos para entrega de projetos cada vez mais restritos, foi identificada a necessidade do reuso de parte do material previamente desenvolvido em projetos anteriores, de maneira a garantir agilidade na entrega do projeto com precisão e cumprimento das expectativas do cliente.

Por esses motivos, a elicitação de requisitos é vista como ponto crítico no desenvolvimento e na implantação de projetos de software, pois uma definição incorreta dos requisitos do projeto pode levá-lo ao insucesso de sua implantação.

Como em todas as áreas de tecnologia, o desenvolvimento de softwares direcionados à indústria de telecomunicações passou e tem passado por diversas mudanças. A própria inovação tecnológica que vem ocorrendo nesta área com a modernização de centrais telefônicas e a busca por métodos mais seguros e eficientes de tarifação, auditoria e combate a fraudes, provocou o crescimento da produção de produtos de software voltados para atender essas necessidades.

Durante a implantação dos produtos de software responsáveis por atender essas inovações, pode-se observar que o tempo despendido na etapa de elicitação de requisitos, freqüentemente, não é utilizado da melhor forma possível. Em muitos

casos grande parte do tempo de duração de um projeto de implantação é gasto no desenvolvimento de requisitos, em retrabalhos, ou consumido no desenvolvimento em duplicidade de componentes entregues previamente em outros projetos.

Pode-se observar também, que o grupo de fornecedores de equipamentos de grande porte de telefonia é um grupo reservado e que, na grande maioria dos casos, as operadoras de telefonia utilizam equipamentos similares, devido à restrição de fornecedores. Como consequência, softwares projetados para realizar atividades como, por exemplo, mediação, acabem apresentando requisitos similares nas implantações. A reutilização de tais requisitos pode desta maneira contribuir com o avanço de melhorias como utilização do tempo e redução de recursos durante a implantação de produtos de software em diversos projetos.

1.2 Objetivo

O objetivo do presente trabalho é a aplicação parcial do método de elicitação de requisitos denominado *Feature-Oriented Domain Analysis* (Análise de Domínio Orientada a Características - *FODA*), na implantação de um componente de produto de software de mediação desenvolvido para atender a área de telecomunicações, visando estudar características que permitam o reuso de requisitos na implantação deste software. A aplicação será realizada em um modelo conceitual simplificado do produto de software, empregando dados simulados de contextos reais de implantação.

1.3 Organização da monografia

Nesse capítulo é feita uma breve explanação sobre o que será tratado na monografia apresentada. No segundo capítulo, serão apresentados alguns conceitos para a fundamentação conceitual do tema de estudo do presente trabalho.

No terceiro capítulo será apresentado o software objeto de análise desta monografia e a necessidade da utilização de um método de elicitação e reuso de requisitos em projetos de implantação do produto de software em questão.

No quarto capítulo será apresentada uma simulação da aplicação de um método de elicitação de requisitos em um modelo conceitual de projeto de implantação do produto de software analisado. E será apresentado o uso do método para identificação de requisitos passíveis de reutilização em um projeto de implantação.

2. Elicitação e métodos de elicitação de requisitos

Neste capítulo se descreve de maneira abreviada o conceito de análise e elicitação de requisitos e qual a sua função em projetos de implantação de produtos de software. São apresentados de maneira sucinta quatro métodos de elicitação de requisitos existentes no mercado. E também é exposta a importância do reuso de requisitos e seu emprego em projetos de implantação de produtos de software.

2.1 Elicitação e Análise de Requisitos

Sendo considerada a primeira atividade técnica no desenvolvimento de um produto de software, a análise de requisitos tem como objetivo principal a definição detalhada das funções que o sistema deverá realizar, não sendo objetivo definir como o sistema será desenvolvido ou como a implantação do produto de software deverá ser administrada. A análise de requisitos é em muitas vezes descrita como uma série de atividades, que podem ser vistas da seguinte forma: elicitação, modelagem, triagem, especificação, verificação¹, tem-se (HEISKARI, Juho, 2008):

- a. Elicitação, fase em que se toma conhecimento e é feita a extração, pesquisa e descoberta das necessidades do cliente, usuários e de potenciais *stakeholders*;
- b. Modelagem, atividade de criação e análise dos modelos dos requisitos, com metas como o aumento da compreensão da solicitação e a busca por lacunas e inconsistências;
- c. Triagem, verificação se um determinado grupo de requisitos é apropriado para ser entregue e endereçado em uma versão específica do produto;
- d. Especificação, toda a documentação referente às funcionalidades do produto.

¹ Não existe um consenso na convenção dos nomes das atividades da Análise de Requisitos.

- e. Verificação, nessa atividade é determinada a consistência do produto e se o mesmo atende todos os requisitos solicitados com ausência de defeitos.

Todas as atividades da etapa de análise de requisitos devem ser consideradas de extrema importância na implantação de um software, pois serão utilizadas como base em tudo que será desenvolvido nas etapas seguintes.

Entre as atividades de análise de requisitos tem destaque a elicitação de requisitos, devido a sua importância e influência geral no projeto. Estudos comprovam que erros na definição dos requisitos durante a fase inicial do projeto são responsáveis por grande parte dos atrasos e falhas na entrega da implantação de projetos de software (DAVEY, Bill; COPE, Chris, 2008), tendo-se em vista que solicitações erroneamente compreendidas se estenderão às etapas seguintes do projeto, compreende-se porque uma incorreta definição inicial acarreta em perdas para todo o projeto, impactando diretamente na sua entrega.

A Figura 1, foi elaborada com base em estudos comparativos sobre os impactos das atividades de análise de requisitos e suas influências no decorrer do projeto. Através da Figura 1 verifica-se a importância da elicitação de requisitos dentre todas as atividades da análise de requisitos.

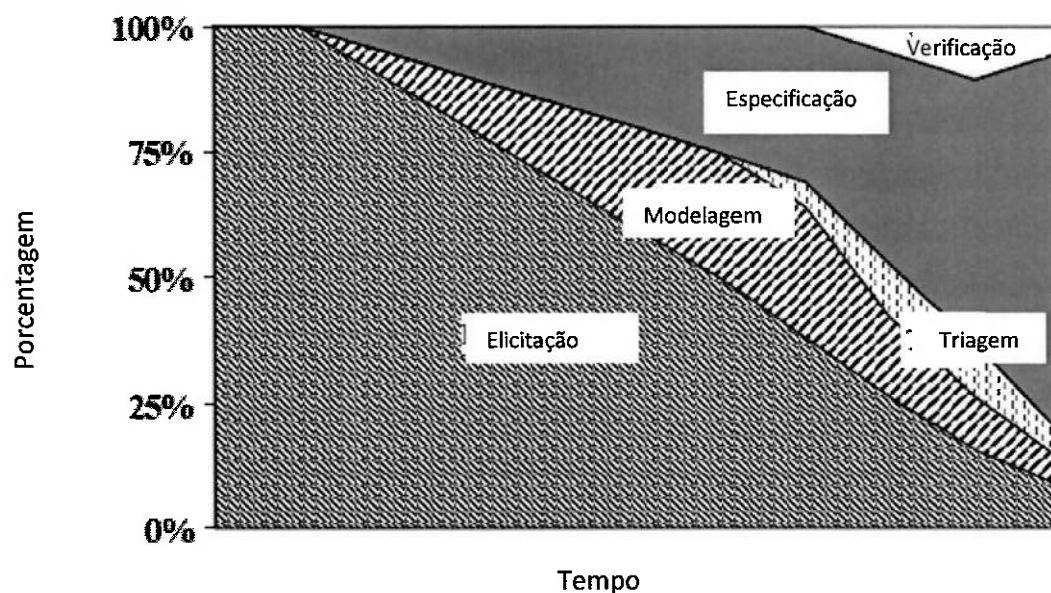


Figura 1 - Modelo paralelo do método de processo de requisitos

(adaptação, MERWE, Alta van der; CRONJE, Johannes; KOTZÉ, Paula, 2005
apud HIKEY; DAVIS, 2003)

Análises demonstram que 90% das falhas em projetos são causadas por definições incorretas dos requisitos (DAVEY, Bill; COPE, Chris, 2008 apud FULLER; TREMBLEY; BRENT), influenciando muito mais em falhas de projetos do que tecnologias de qualidade precária, atrasos na entrega ou problemas em gerenciamento de mudanças. Muitas vezes o custo dessa falha é extremamente alto, tendo em vista que, não somente a ocorrência da falha por si só no projeto aumenta os custos, mas também o tempo gasto corrigindo as falhas originadas por requisitos especificados incorretamente, a etapa de elicitação de requisitos quando bem desenvolvida pode trazer até 75% de economia nos custos de um projeto (DAVEY, Bill; COPE, Chris, 2008 apud URQUHART, 1999). A Tabela 1 enumera diferentes obstáculos mapeados na fase de elicitação de requisitos.

Tabela 1 – Mapeamento de problemas comuns durante levantamento de requisitos (adaptação (DAVEY, Bill; COPE, Chris, 2008 apud TSUMAKI e TAMAI, 2006)

Requisitos incompletos	Compreensão incompleta das necessidades
Conhecimento incompleto	Pouca colaboração dos usuários
Pouca atenção aos requisitos assumidos	Requisitos incorretos
Requisitos ambíguos	Termos homônimos e sinônimos
Requisitos não passíveis de teste	Considerações desnecessárias de design
Requisitos inconsistentes	Falta de clareza nas intenções dos solicitantes
Visões diferentes dos usuários	Aceitação contínua de requisitos adicionais
Requisitos em excesso	Fontes de informação desorganizadas

Muitos solicitantes	Promessas acima do possível de entrega pela equipe de vendas
---------------------	--

Esta lista cobre problemas que ocorrem geralmente por falhas de comunicação durante um projeto. No entanto, a quantidade de problemas também aumenta porque as necessidades das organizações mudam com o tempo e sempre existe a possibilidade de que as pessoas somente percebam essas mudanças depois que os requisitos já foram estabelecidos. A maioria das teorias de sistemas de informação pressupõe que os requisitos sejam estáveis e ignoram que a realidade é dinâmica e necessidades são criadas assim que novas oportunidades aparecem.

Ao longo do tempo foram criados vários métodos de auxílio para o processo de elicitação de requisitos, cada um com características distintas e que se adequam a diferentes tipos de necessidades em projetos de implantação, para segmentos distintos. Nas seções seguintes descrevem-se brevemente de forma imparcial alguns métodos existentes no mercado. Vale ressaltar que cada método tem pontos fracos e fortes, cabendo ao analista do projeto utilizar-se de técnicas para escolha mais adequada de qual método utilizar, não sendo objetivo desse trabalho discutir tais técnicas.

Como visto anteriormente, a seleção apropriada, a análise e a documentação de requisitos no início de um projeto de desenvolvimento de software são consideradas atividades de preparação importantes para o sucesso de um projeto. Tais atividades acompanham o projeto desde sua concepção inicial até fases posteriores à sua implantação. Esse conceito pode ser aplicado inclusive na atividade de elicitação de requisitos, que deve ser revista em diferentes etapas do projeto para garantir o alinhamento das necessidades do projeto com as suas atividades de execução. Também se deve levar em conta que a priorização dos requisitos durante o planejamento de implantação de um novo produto de software é fundamental para manter o foco nas reais necessidades do projeto e também para garantir a escolha correta do produto que será adquirido.

2.2 Métodos para Elicitação de Requisitos

A descrição de métodos para a elicitação de requisitos existentes deveria ser precedida do estabelecimento de critérios bem definidos para a seleção dos que seriam descritos. Neste trabalho, que pode ser caracterizado como estudo exploratório inicial, adotou-se um critério bem simples: os métodos devem apresentar características bem distintas, ter uma maturidade reconhecida, poderem ser empregados nos mais diversos campos de aplicação, sobretudo a implantação de produtos de software. Dentre os vários métodos existentes e em uso no mercado que atendem a esses critérios, foram selecionados apenas quatro métodos. Suas descrições serão feitas, neste capítulo, de forma geral e simplificada.

2.2.1 *Controlled Requirements Expression*

O método de elicitação de requisitos conhecido como *Controlled Requirements Expression* (Expressão Controlada de Requisitos – CORE) foi desenvolvido pelo instituto de pesquisas aeroespaciais britânico em 1979, (MOULDING, M.R.; SMITH, L.C. 1992). O método CORE é baseado em aproximações de funções decompostas e é fundamentado no modelo conhecido como *Viewpoint* (Ponto de Vista). Métodos orientados pela modelagem *Viewpoint* são reconhecidos por colecionarem informações sobre o sistema percebido por diferentes solicitantes, sobre os problemas que os atingem, sobre o ambiente ou sobre o domínio em que são coletados de uma perspectiva particular. Os *Viewpoints* permitem melhor entendimento das necessidades dos usuários mediante cada uma das atividades do processo (DA SILVA, S. M. A.; BONIN, Marcos R.; PALUDO, Marco A., 2006 apud FINKELSTEIN *et al.*, 1992).

O método de elicitação de requisitos CORE abrange tanto a análise como a especificação de requisitos, esclarecendo a visão dos usuários fornecidos pelo sistema proposto e as limitações impostas pelo ambiente operacional do sistema.

Conjugando níveis de desempenho e investigação de confiabilidade, o CORE provê métodos e notações para cada fase da elicitação e análise de requisitos, resultando em um fluxo estruturado de especificações. O CORE é considerado um método maduro e apresenta uma variedade de guias definidos que expõem como se deve aplicar o método. Esse método tem uma aproximação flexível, permitindo que seja aplicado em diferentes problemas, incluindo a implantação de produtos de software. O CORE encoraja a contribuição de diversas comunidades no desenvolvimento dos requisitos e delinea as tarefas dos membros de suas comunidades e estrutura a comunicação entre esses grupos. Uma análise incremental do fluxo de informação pode ser realizada utilizando o CORE, onde cada passo anterior proverá a base de fundação para as especificações do passo presente

O CORE está relacionado a dois elementos da modelagem *Viewpoint*: limites e definições. Limites, no modelo *Viewpoint*, são entidades externas que afetam o sistema de alguma maneira. Definições, na modelagem *Viewpoint*, são sub-processos do sistema.

Em um primeiro nível, o método CORE compreende todas as entidades que interagem com o sistema e provê orientação para identificar *viewpoints* funcionais e não funcionais. O segundo nível faz a distinção dos *viewpoints* que limitam ou definem o sistema. Neste caso, pode-se definir um *viewpoint* para a implantação de produtos de software que requerem, por exemplo, personalizações particulares a determinados clientes.

2.2.2 Misuse Cases

O *Misuse Case* (Caso de Mau Uso) tem como princípio básico demonstrar o inverso do que se representa em um caso de uso clássico (SINDRE, Guttorm ; Opdahl, ANDREAS L., 2001), representando tudo que o sistema não deve permitir que ocorra. Adota-se aqui a definição usual de caso de uso: uma seqüência completa de interações e ações entre atores e sistema que agregam valor ao sistema. O ator é um *stakeholder* (parte interessada) que usa o sistema. Em contra partida pode-se definir

Casos de uso – e casos de mau uso (*misuse cases*) - mostram-se úteis para a elicitação, comunicação e documentação de requisitos. Alguns estudos mostram que projetos de desenvolvimento em que a fase inicial de análise se concentra na utilização de casos de uso, em vez de requisitos transcritos em forma textual, são mais eficazes para capturar as necessidades dos usuários (SINDRE, Guttorm ; Opdahl, ANDREAS L., 2001). Porém, somente a utilização de casos de uso não consegue prever ações indesejáveis ao projeto de implantação. “Uma coleção de casos de uso não é o melhor substituto para a especificação de requisitos” (adaptação, SINDRE, Guttorm ; Opdahl, ANDREAS L. apud ANTÓN, DEMPSTER e SIEGE. 2000).

Apesar de apresentar inúmeros pontos favoráveis, em algumas situações à utilização exclusiva do caso de uso como ferramenta para elicitação de requisitos pode fazer com que sejam feitas suposições relacionadas ao domínio do problema, tendendo em alguns casos a seguir adiante para a etapa de implantação do projeto prematuramente. Ocultando pontos de fraqueza do produto de software em implantação, não expostos de maneira ideal de modo a serem compreendidos por todo o grupo de implantação. Um caso de uso típico deve descrever algumas funções que o sistema precisará realizar. Desta maneira, casos de uso são boas ferramentas para trabalhar os chamados requisitos funcionais, mas não são necessariamente eficientes para a representação de requisitos não funcionais, como por exemplo, segurança.

A utilização do método *Misuse Case* facilita o entendimento e localização de problemas externos potenciais que podem interferir de maneira negativa no projeto do sistema (SINDRE, Guttorm ; Opdahl, ANDREAS L., 2001). Com a sua utilização, é possível identificar a vulnerabilidade dos requisitos funcionais do sistema e formular novos requisitos, que protegem o sistema de falhas internas causadas por utilização incorreta dos usuários ou de ataques externos, que visam invadir um sistema e com isso fazer o uso inapropriado das informações do cliente.

2.2.3 Accelerated Requirements Method

O *Accelerated Requirements Method* (Método acelerado de Requisitos – ARM) é um método facilitador da eliciação de requisitos, constituído das seguintes fases (HUBBARD, R.; MEAD, N.; SCHROEDER, C., 2000):

- a. Fase de preparação;
- b. Fase facilitadora das atividades;
- c. Fase de fechamento dos entregáveis.

Também são definidas várias outras atividades sucessoras a estas em conformidade com as necessidades do cliente e com o cronograma estipulado.

Durante a fase de preparação, são realizados o planejamento e a preparação do projeto de implantação, visando garantir uma etapa eficaz. Durante a atividade de captura de objetivos, em que também o escopo dos esforços do projeto é inicialmente definido, definem-se os fatores de avaliação da eficácia do projeto, identificam-se os participantes-chave, e desenvolve-se o cronograma preliminar.

A fase de preparação prepara o projeto para a fase seguinte, que é a fase facilitadora das atividades. São seis os passos que definem a fase de preparação (HUBBARD, R.; MEAD, N.; SCHROEDER, C., 2000):

- a. Definição de metas, objetivos e critérios de sucesso do projeto;
- b. Definição dos objetivos e o escopo preliminar das reuniões;
- c. Estabelecimento da divisão estrutural do projeto e identificação de seus participantes;
- d. Determinação dos aspectos ambientais e logísticos;
- e. Estabelecimento das expectativas dos participantes;
- f. Comunicação com os participantes.

Pode-se utilizar de formulários de pesquisa de opinião e entrevistas para obter essas informações.

Durante a fase facilitadora de atividades, um líder facilitador neutro seleciona os participantes do projeto através de um processo estruturado de coleta de requisitos funcionais do projeto em consideração. O processo facilitador emprega escopos

definidos, *brainstorming* e técnicas de priorização e explanação. Os seis passos dessa fase são:

- a. Recebimento dos comentários dos investidores do projeto;
- b. Fechamento do escopo;
- c. *Brainstorm*, organização e nomeação;
- d. Detalhamento;
- e. Priorização;
- f. Pesquisa de opinião dos participantes.

Durante a fase de encerramento, os entregáveis-chave, e os próprios requisitos do projeto são revisados, publicados e distribuídos, e todas as atividades sucessoras são planejadas.

2.2.4 Feature-Oriented Domain Analysis

O método *Feature-Oriented Domain Analysis* (Análise de Domínio Orientada a Características - *FODA*) é dirigido pela análise de domínio que pode ser organizada para possibilitar o reuso de requisitos.

A análise de domínio coleta e representa informações de softwares de sistemas que compartilham um conjunto comum de funcionalidades e dados. A existência de três fases básicas no processo de análise de domínios é sugerida por esse método (KANG, Kyo C. et al., 1990):

- a. Análise de contexto: definição dos limites do domínio em análise;
- b. Modelagem de domínio: descrição dos problemas relacionados ao domínio que será endereçado ao software;
- c. Modelagem da arquitetura: criação de arquiteturas de software que programam soluções aos problemas no domínio.

O texto que segue foi inteiramente baseado em estudos realizados por pesquisadores da área e tendo como referência (KANG, Kyo C. et al., 1990).

Na análise do contexto, os analistas de domínio interagem com os usuários e os especialistas do domínio para demarcar as fronteiras e estabelecer um escopo

apropriado para a análise. O analista também coleta as fontes de informação para análise de desempenho.

Durante a modelagem de domínio, o analista de domínio utiliza as fontes de informação e produtos da análise de contexto para apoiar a criação da modelagem de domínio. Esse modelo é revisto pelo usuário final, o especialista do domínio e o analista de requisitos.

Utilizando os elementos produzidos pela modelagem de domínio, o analista de domínio produz o modelo de arquitetura de software. Esse modelo deve ser revisto pelo especialista de domínio, pelo analista de requisitos e o engenheiro de software. O usuário não participa desta revisão.

Na análise de domínio são identificados três grupos de participantes:

- a. Fontes: usuários finais e especialistas do domínio;
- b. Produtores: analistas de domínio, que coletam informação e realizam análise;
- c. Consumidores: Analistas de requisitos, projetistas de software e usuários finais.

Analistas de requisitos e projetistas de software farão o reuso dos produtos da análise de domínio quando implantando um novo sistema. Durante a implantação, esses produtos são moldados na produção da especificação e no *design* do sistema. Esse processo de modelagem fornecerá uma imagem geral para o analista de domínio e para o especialista, permitindo a modificação ou definição da análise de domínio para um futuro desenvolvimento. Essa imagem também servirá para melhorar o processo de análise de domínio, descobrindo as possíveis fraquezas no método original.

O método de análise de domínio deve fornecer representações específicas para documentar os resultados de cada atividade de análise de domínio. As representações definem o escopo do domínio, e provê arquiteturas para implantação da solução. A Figura 3 apresenta como os produtos de software interagem com as fases da análise de domínio e seus participantes.

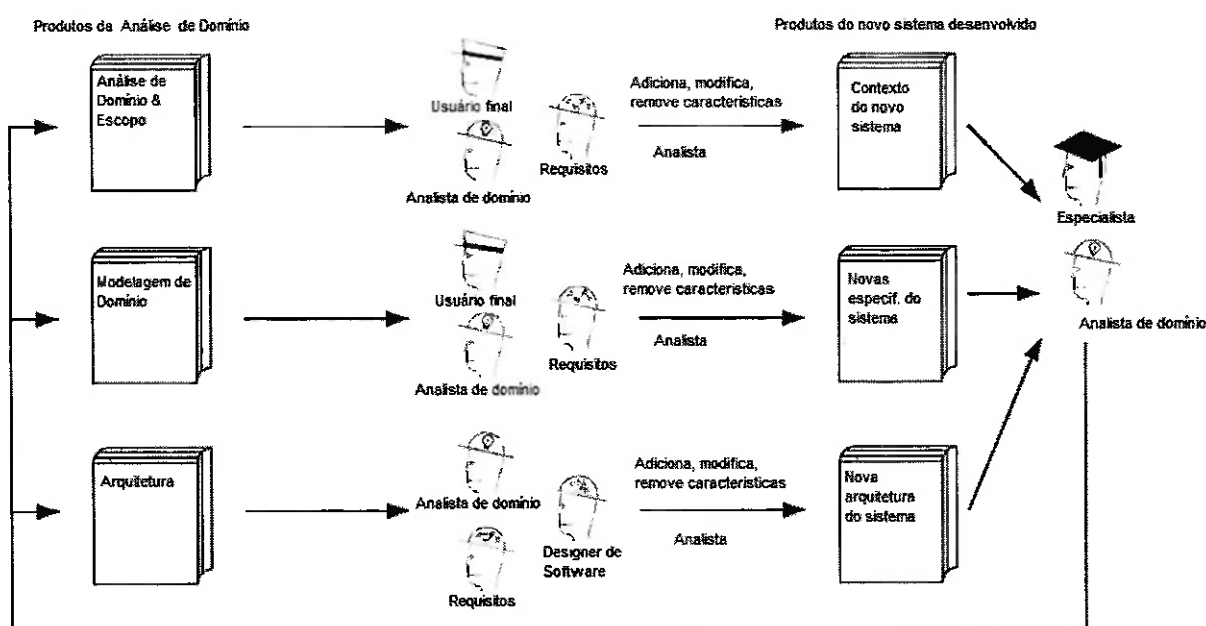


Figura 3 - Visão dos produtos na Análise de Domínio
(adaptação, KANG, Kyo C. et al., 1990)

As várias definições referentes ao conceito da análise de domínios requerem que o próprio termo domínio tenha uma definição consensual para que melhor seja compreendido. Um domínio não necessariamente é definido ou ocorre em um nível específico do software. O domínio pode ser encarado através de um conceito mais geral que pode ser ajustado para poder ser aplicado em qualquer classe potencial de sistema. Essa classe será referenciada como domínio-alvo, que por sua vez poderá apresentar domínios em níveis mais altos ou mais baixos. Por exemplo, diferentes instâncias do mesmo tipo de sistema podem ser agrupadas para definir um domínio. Ou também de modo similar, um domínio pode ser identificado em níveis de base e ainda assim ser visto como domínio-alvo. Portanto, a caracterização de um domínio como domínio-alvo não tem relação à localização deste domínio ou seu relacionamento com outras entidades, podendo ser caracterizado de acordo com objetivo da análise

O método FODA provê meios para aplicar os elementos da análise de domínio para dar suporte ao desenvolvimento de software e a implantação de produtos de software. A visão da análise de domínio mostra que seus produtos podem ser integrados em um processo mais geral para apoiar:

- a. O entendimento do domínio;
- b. A Implantação das aplicações de domínio;
- c. A criação de requisitos reutilizáveis;
- d. O suporte à criação da análise de domínio e outras ferramentas reutilizáveis.

A Figura 4 apresenta de forma ilustrativa as fases e produtos da análise de domínio

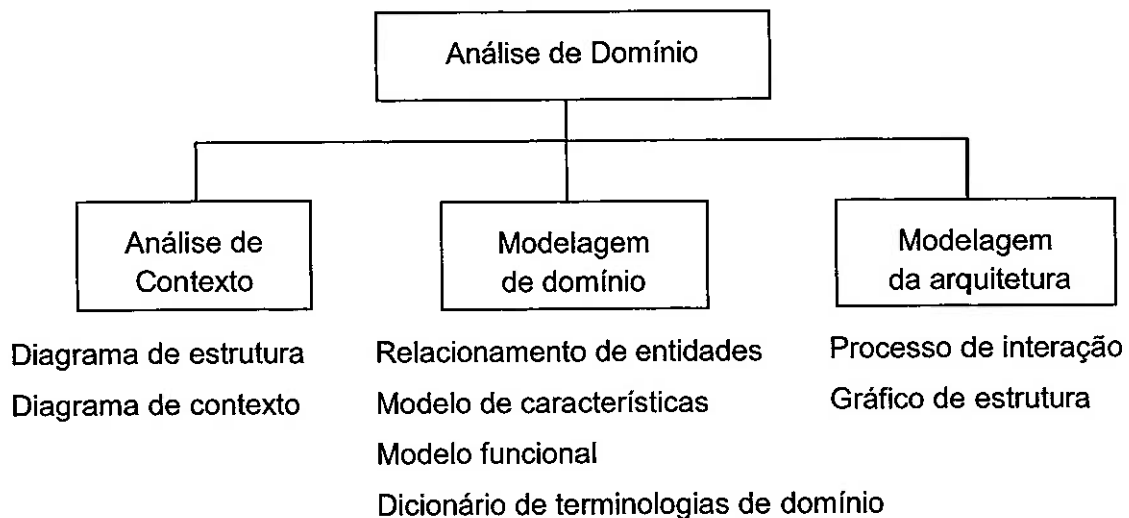


Figura 4 - Fases e produtos da análise de domínio

2.3 Reuso de Requisitos

O tema do reuso de requisitos é complexo e tem sido bastante pesquisado. Aqui são apresentadas apenas algumas linhas gerais desta questão.

Normalmente, a produção de software é controlada pelos seguintes fatores: padrões de qualidade, tempo de entrega e atendimento dos requisitos, em que podemos também considerar como requisitos as expectativas dos clientes. Visando a compreensão dos padrões de qualidade especificados no prazo contratado, as atividades de reuso de software têm sido amplamente utilizadas como uma alternativa

para solucionar diversos problemas encontrados durante a implantação de um produto de software (VILLEGAS, Oscar L.; LAGUNA, Miguel A , 2001).

O reuso em produtos de software tem como objetivo aumentar significativamente a qualidade e a produtividade dos artefatos do software. Desse modo, pretende reduzir o custo da sua implantação e o tempo empregado para a execução do projeto (HOMOD, M. Sami ; RINE, David C.,1998) garantindo a entrega correta dos requisitos do sistema. Para que a reutilização de requisitos seja realmente eficiente é preciso avaliar cada requisito reutilizável e certificar-se que os requisitos identificados serão completamente atendidos através do reuso dos componentes identificados.

O reuso na implantação de produtos de software, implica em atividades que vão além da reutilização de componentes previamente desenvolvidos. A verdadeira reutilização de requisitos de software consiste em um uso sistemático de requisitos de produtos de softwares já desenvolvidos e experimentados (VILLEGAS, Oscar L.; LAGUNA, Miguel A., 2001).

Com todos os benefícios trazidos pela reutilização de requisitos, é importante incluir dentre as atividades de elicitação de requisitos, o levantamento dos requisitos reutilizáveis, assim como todas as atividades de análise para verificação da compatibilidade dos requisitos.

As atividades de desenvolvimento para reuso, desenvolvimento com reuso e implantação com reuso são relacionadas a componentes de bibliotecas e baseadas em suporte operativo para atuar, gerenciar e usar os componentes reutilizáveis (VILLEGAS, Oscar L.; LAGUNA, Miguel A , 2001). Os artefatos devem ser enfocados através de duas perspectivas, composição e geração. Na composição, os artefatos são considerados elementos passivos que atuam como blocos de construção para compor as aplicações do software. Do outro lado, na geração, os artefatos são vistos como elementos ativos que podem ser transformados para obter produtos de software.

Os requisitos de software devem ser tratados adequadamente através da estratégia de reuso (HOMOD, M. Sami ; RINE, David C.,1998). Tanto no ambiente de composição quanto no de geração, o processo de reuso deve ser guiado pelos requisitos propostos na arquitetura do software. Esses requisitos devem ser supostamente satisfeitos pelos módulos existentes que estão armazenados em um

repositório. Os requisitos devem ser previamente classificados quando os módulos de software são criados pelos desenvolvedores.

3. O software mediador e suas funções

Este capítulo descreve, de modo sucinto, o produto de software objeto de estudo do presente trabalho. O texto deste capítulo foi desenvolvido com apoio de documentação de uso comercial de uma empresa do ramo de telecomunicações, não podendo ser explicitamente referenciada. Todas as referências são generalistas e não comprometem o produto.

3.1 Descrição geral

O software objeto de estudo desse trabalho é conhecido no mercado de telecomunicações como, Mediador. O software de mediação, ou mediador em telecomunicações, é um sistema utilizado para converter formatos de dados distintos em outros. Essa conversão normalmente se faz necessária para que os dados sejam utilizados posteriormente pelas áreas de faturamento em operadoras de telecomunicações. Estas precisam processar UDRs (*Usage Detail Records* – Registros Detalhados de Uso) gerados durante transações telefônicas, as quais, após serem processados, passam a ser chamados de CDRs (*Call Detail Record* – Registro Detalhado de Chamada) na área de telefonia.

Plataformas de mediação em telefonia receberam esse nome pelo papel que interpretam na área de telecomunicações e no processamento de dados oriundos de operadoras. Fazem a intermediação entre vários sistemas diferentes, conhecidos como sistemas de *upstream* e sistemas de *downstream*. Em um cenário típico de telecomunicações pode-se fazer as seguintes considerações para melhor exemplificar o sistema de mediação.

- a. Os sistemas de *upstream* são todos os sistemas que provêem dados para a plataforma de mediação, geralmente sistemas de centrais telefônicas;

- b. Os sistemas de *downstream* são os sistemas que recebem os dados da plataforma de mediação, geralmente realizando funções como: contagem de chamadas, auditoria dos eventos, arquivamento e geração de bilhetes de cobrança.

O sistema de mediação realiza a coleta, a manipulação de dados e prepara os dados enviados por sistemas de *upstream* para serem utilizados pelos sistemas de *downstream* que, em muitos casos, somente aceitam dados em formatos específicos. Comumente uma plataforma de mediação é utilizada para realizar as seguintes atividades, conforme ilustra a Figura 5:

- a. Coletar e validar dados de CDR;
- b. Manipular e alterar a formatação;
- c. Transformar dados de acordo com regras de negócio;
- d. Enviar dados para sistemas de destino (*downstream*).

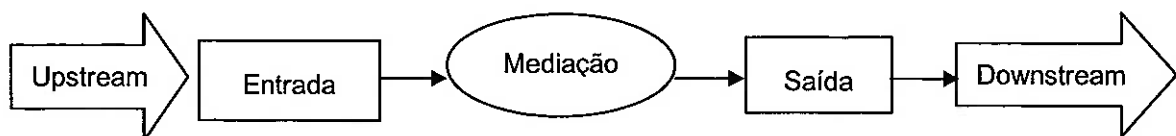


Figura 5 - Visão geral de um sistema de mediação

No cenário de faturamento de uma operadora, a mediação é o primeiro passo antes do recebimento de um CDR. Pode-se considerar genericamente que o CDR mediado é enviado para o sistema de tarifação. Este calcula os preços associados aos CDRs, para o consumo de sistemas de contagem, auditoria, arquivamento, ou geração de bilhetes de cobrança.

Apesar de inicialmente ter sido amplamente utilizado para finalidades de faturamento, auditoria e detecção de fraudes, o software de mediação também tem sido utilizado para a geração de relatórios estatísticos. Estes relatórios são muitas vezes utilizados para o planejamento de capacidade de quantidade de envio e recebimento de chamadas nos equipamentos utilizados pela rede de telefonia das operadoras.

3.2 Necessidades do reuso de requisitos na implantação de produtos de softwares em mediação

Durante a implantação de um novo projeto de mediação são criados elementos de software personalizados para cada operadora, visando a integração do sistema mediador com os diversos elementos de rede e processamento em operação.

Foi identificado que, apesar de serem entregues para operadoras distintas, muitos elementos desenvolvidos para a integração do sistema poderiam ser reutilizados, uma vez que as operadoras têm, em geral, os mesmos fornecedores mundiais de sistemas de telefonia.

Para ilustrar a questão, considera-se o seguinte cenário. Uma operadora **A** tem, entre suas centrais telefônicas, centrais do fornecedor **X**. São desenvolvidos elementos de integração para que o sistema de mediação se comunique com a central telefônica **X**, e, em se tratando de um sistema mediador, ele deverá traduzir os dados coletados da central do fornecedor **X** para um sistema de *downstream* de um fornecedor **M**. No caso de outra operadora **B** utilizar centrais telefônicas do mesmo fornecedor **X** e sistemas de *downstream* do fornecedor **M**, um novo projeto de implantação do sistema mediador poderia reutilizar os elementos desenvolvidos para a operadora **A** anteriormente. Isso poderia reduzir o tempo de desenvolvimento de novos elementos, melhorando assim o tempo total de entrega do projeto e provavelmente trazendo redução de custos do projeto.

Para que essa reutilização seja aceitável, não se pode deixar de levar em consideração a avaliação dos requisitos definidos por cada operadora. Os requisitos que orientarão a construção de novos elementos para a implantação do software devem ser avaliados de acordo com o que foi solicitado, de modo que atenda às solicitações previamente definidas pelos clientes.

A Figura 6 representa esquematicamente produtos comuns à duas operadoras que, por sua vez, proporcionam a reutilização dos requisitos em projetos de desenvolvimento e implantação:

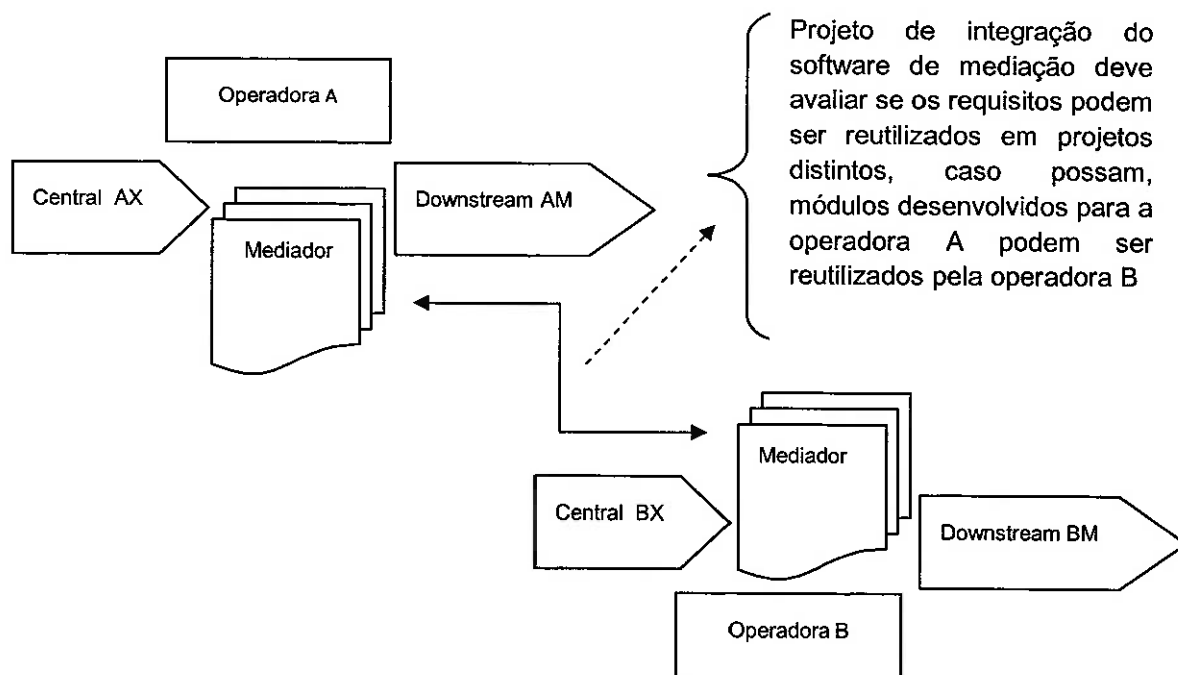


Figura 6 - Representação da utilização de re-uso de requisitos

4. Aplicação do Método FODA em um Modelo Conceitual da Implantação do Software de Mediação

Após a descrição de alguns métodos para elicitação e análise de requisitos e das linhas gerais dos benefícios proporcionados pelo reuso de requisitos na implantação de produtos de software, neste capítulo descreve-se um modelo conceitual da implantação do software de mediação e a aplicação de um método de elicitação e análise de requisitos visando o reuso de requisitos na implantação.

O método escolhido foi o *Feature Oriented Domain Analysis* - FODA, por apresentar duas características constitutivas: incorpora explicitamente noções e orientações para a implantação das aplicações de um determinado domínio; e é orientado à criação de requisitos reutilizáveis.

Também neste capítulo, apresentam-se, de modo sintético, algumas possibilidades de reuso de requisitos no modelo conceitual da implantação considerado.

4.1 Fluxo de dados do modelo conceitual da implantação

Como documentação de apoio para esta seção foi utilizado o mesmo material bibliográfico referenciado no Capítulo 3.

Como representado pela Figura 6 item 3.2, o modelo da implantação do software mediador que será utilizado como base do estudo neste trabalho pode ser decomposto de modo simplificado em três áreas:

- a. Entrada: onde arquivos de um sistema de *upstream* de uma determinada operadora são coletados periodicamente de modo manual ou automático, utilizando-se uma grande variedade de protocolos com diferentes formatações de dados. A fonte de entrada é um sistema de dados como, por exemplo, uma central telefônica, um computador, ou um dispositivo de fita;

- b. Mediação: onde os arquivos coletados são processados, para que se possa realizar conversão de dados, filtragem, coleta de estatísticas, entre outras funções. Os dados podem ser enviados a um destino externo, retidos para análise de erros ou filtrados de acordo com os parâmetros definidos pela saída;
- c. Saída: onde os arquivos processados são armazenados para posterior ou imediata entrega aos sistemas de *downstream* contratados pela operadora.

Durante a implantação do software de mediação em estudo alguns parâmetros deverão ser definidos para executar as seguintes ações:

- a. Receber ou localizar dados de um sistema de entrada externo (por exemplo, uma central telefônica);
- b. Transmitir dados para um sistema de saída externo (por exemplo, um sistema de faturamento).

O conjunto desses parâmetros é denominado de portal. Um portal pode ser dividido em duas categorias: portal de aquisição e portal distribuição. Um grupo lógico de portais de aquisição e de distribuição é denominado como um sistema externo. A Figura 7 representa a relação entre os portais e o sistema externo

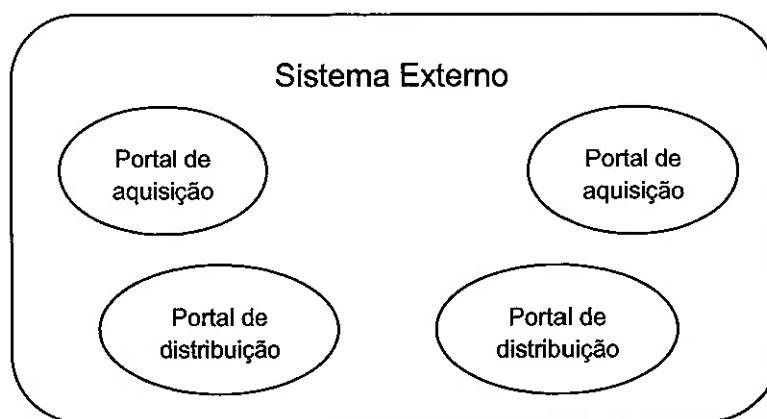


Figura 7 - Sistema Externo

Além de coletar dados e enviá-los para o sistema de *downstream*, o software de mediação tipicamente funciona como uma ferramenta para intermediação dos dados, processando os arquivos recebidos antes que sejam entregues. Este processamento

de dados, ou mediação, é realizado pelo que identificamos como o módulo lógico do negócio.

O módulo lógico é reconhecido por uma seqüência de operações e portais de distribuição, para processamento dos dados coletados, identificados e especificados durante a implantação. Operação é o evento que ocorre após a coleta do arquivo. Uma operação deve ser personalizada para realizar uma variedade de ações incluindo:

- a. Validação de dados;
- b. Encaminhamento de dados;
- c. Filtragem de dados;
- d. Transformação de dados;
- e. Coleta de estatísticas referente aos dados.

A Figura 8 apresenta um diagrama geral das operações, portais de entrada e saída de um sistema de mediação.

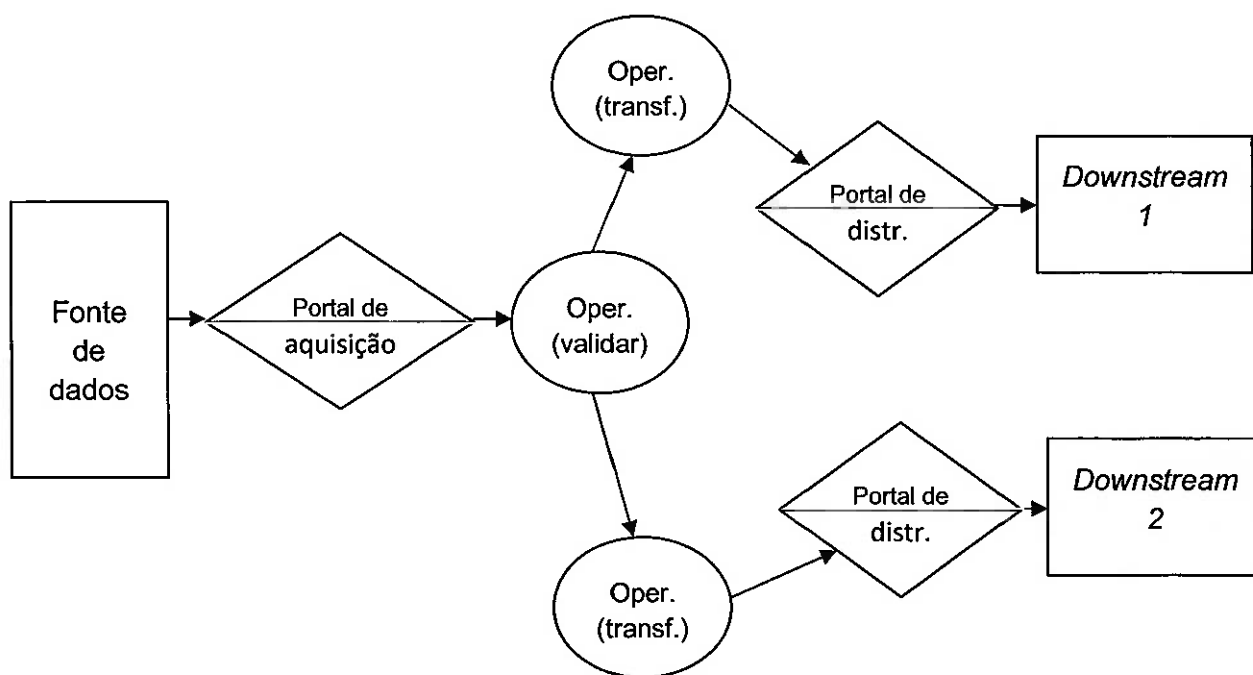


Figura 8 - Visão geral do software de mediação

Além da transformação de dados, o software mediador atua como módulo de identificação de erros gerados, sendo possível criar atividades de rotina para executar a correção de erros recorrentes. O software mediador realiza também detecção de duplicidade de CDRs. Esta detecção é feita mediante a retenção dos arquivos coletados na base do software mediador. Para cada arquivo retido o software gera uma chave de comparação criada a partir de dados definidos pelo cliente. É gerada uma chave de identificação para cada novo arquivo recebido que ficando armazenada em uma base de dados, durante o processamento para detecção de duplicidade é feita a comparação das chaves armazenadas nesta base. Quando é identificada uma chave com duplicidade, isto é, chaves iguais, o arquivo referente à chave encontrada em duplicidade é descartado.

Além dos módulos de detecção de erro e duplicidade, o software mediador apresenta um módulo para remontagem de CDRs incompletos. Em casos de chamadas de longa duração, ou chamadas nas quais o usuário final transita entre estações rádio base (ERB) o CDR gerado poderá não ser encerrado corretamente, pois o arquivo gerado pela central telefônica é fechado e enviado para o sistema de mediação antes da finalização da chamada. Eles também poderão ser encerrados por uma central diferente da central de origem, dependendo da localização do usuário em trânsito, assim como no caso da duplicidade. Os CDRs incompletos também podem ser retidos na base de dados e remontados de acordo com chaves de comparações definidas durante a implantação do software. O produto gerado para essa remontagem é desenvolvido também de acordo com os requisitos especificados pela operadora de telefonia.

4.2 Aplicação do método FODA na implantação do software de mediação

A seguir será feita a apresentação da aplicação de dois domínios do método de elicitação de requisitos FODA - descrito no capítulo 3 - na implantação do software de mediação: análise de contexto e modelagem de domínio. Na realidade, a aplicação é

feita de modo simulado em um modelo conceitual da implantação do software de mediação. Como dito no início do capítulo, o método FODA foi eleito por apresentar ferramentas de reuso de requisitos e aplicar-se à implantação de domínios de aplicação.

Todas as passagens do texto que segue, detalhando partes do método FODA não descritas no Capítulo 3, estão baseadas em (KANG, Kyo C. et al., 1990).

A análise de contexto terá como foco o diagrama de contexto e o diagrama de estrutura. A seção de modelagem de domínio descreve o diagrama relacionamento de entidades e os diversos componentes do modelo de características, ferramentas de caracterização automáticas, o modelo funcional dos fluxos de dados e o modelo de transição de dados. A idéia é utilizar essas ferramentas para detectar requisitos semelhantes em projetos distintos e desta forma poder reutilizá-los conforme a necessidade do projeto em implantação.

4.2.1 Informações sobre o domínio

As seguintes fontes foram utilizadas como entradas para a análise de domínio do software de mediação:

Especialista de domínio: pessoas com alto nível de conhecimento das interfaces de usuário, de softwares de mediação, do fluxo de dados gerados pelas centrais telefônicas.

Sistema de mediação: Software mediador

Literatura de domínio: Manual do software mediador, documentação do formato de dado gerado pela central telefônica, documentação dos formatos de dados aceitos pelos sistemas de *downstream*.

É essencial que seja primeiramente entendida a área de domínio proposta de maneira a compreender claramente o escopo do domínio. Após uma análise inicial ficou claro que devido à complexidade do sistema não será realizada a análise do domínio completa, ficando restrita aos portais de entrada do software de mediação.

4.2.2 Análise de Contexto

A definição do escopo do domínio que será analisado é feita através da fase de análise de escopo. Os dois principais produtos gerados durante essa análise são: diagrama de contexto e diagrama de estrutura, como representado na Figura 4 do Capítulo 2, reproduzida novamente aqui na Figura 9.

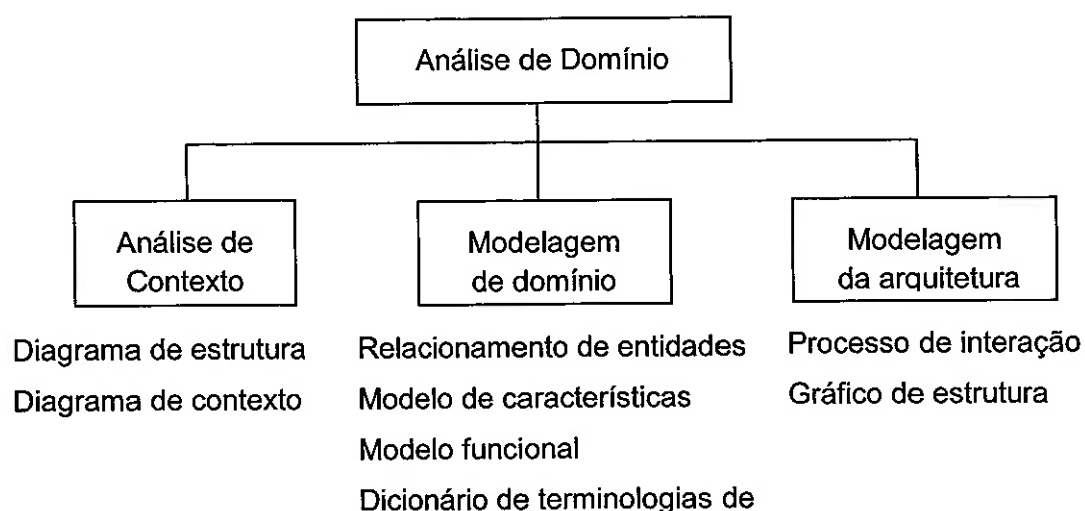


Figura 9 - Fases e produtos da análise de domínio

A proposta desses diagramas é delinear claramente qual será a análise do escopo. O diagrama de contexto mostra a posição do portal de entrada relacionado ao sistema de mediação no todo. Através desse diagrama podemos separar o conceito lógico de um portal de entrada de todos os outros itens relacionados a ele em um sistema de mediação.

O diagrama de contexto apresentado na Figura 10 representa uma visão do fluxo geral de dados das interfaces do portal de entrada como também partes significativas do software de mediação.

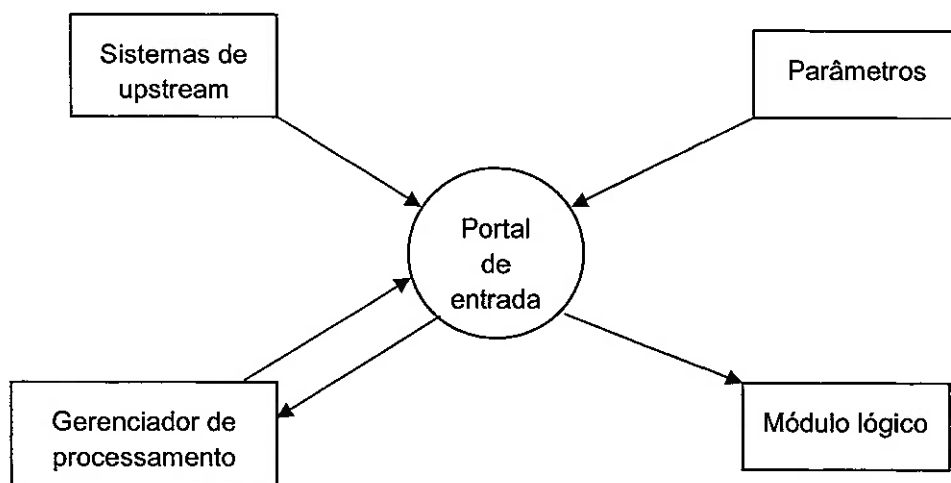


Figura 10 - Diagrama de contexto

4.2.3 Modelagem de domínio

O modelo de domínio descreve os elementos do sistema do ponto de vista do domínio da aplicação. Como apresentado na Figura 9, o modelo de domínio é compreendido por:

- a. Relacionamento de entidades;
- b. Modelo de características;
- c. Modelo funcional;
- d. Dicionário de terminologias de domínio.

O modelo de relacionamento de entidades consiste de três partes:

- a. Diagrama de relacionamento de entidades;
- b. Atributos das entidades;
- c. Ligações entre as entidades e seus relacionamentos.

A Figura 11 apresenta o diagrama de relacionamento de entidades para um portal de entrada e seus relacionamentos com outras entidades importantes do software de mediação.

As entidades apresentadas na figura seguinte, Figura 11, são:

- a. Portal de entrada;
- b. Módulo lógico;

- c. Parâmetros de configuração;
- d. Coletor de dados;
- e. Processos de execução;
- f. Operações.

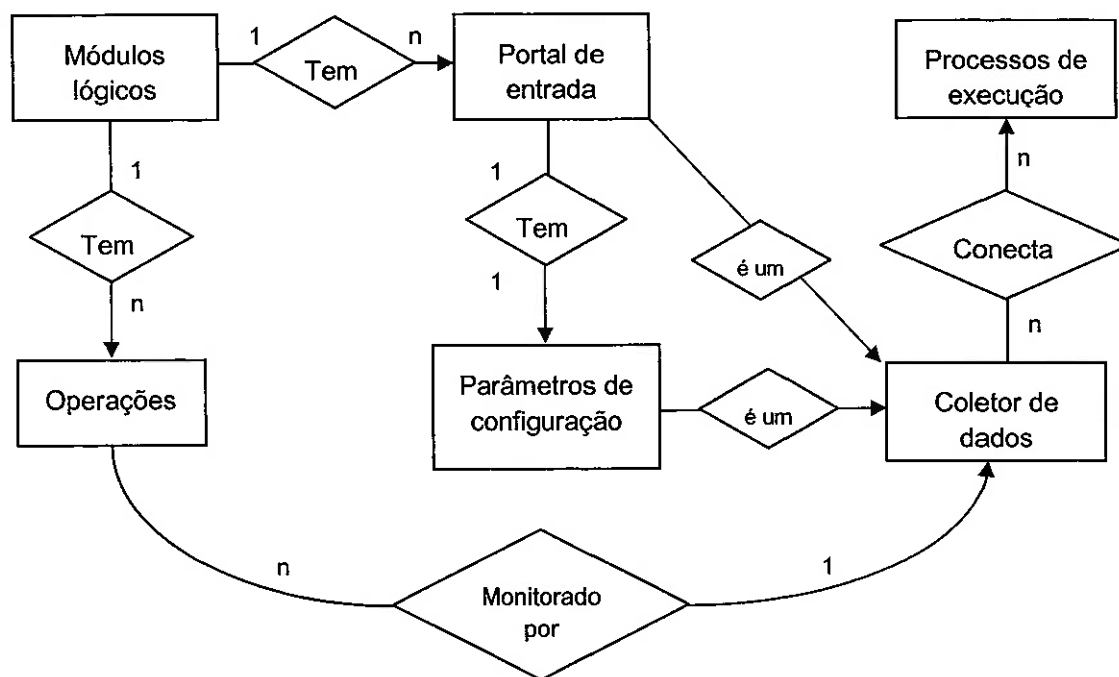


Figura 11 - Diagrama de relacionamento de entidades

Na implantação, são personalizados os Módulos Lógicos, as Operações e os Parâmetros de Configuração.

Os atributos das entidades podem ser encontrados no APÊNDICE A.

4.2.4 Modelo de características

Como visto no Capítulo 2, o método FODA é um método de análise de domínio que mantém o foco nas características do sistema de domínio. De acordo com o conceito

do método, a característica pode ser definida como um aspecto, qualidade, ou uma peculiaridade de um sistema visível ao usuário final.

O módulo de características presente na modelagem do domínio do método contém quatro elementos:

- a. Diagrama de características;
- b. Composição das regras das características;
- c. Registro dos problemas e decisões;
- d. Catálogo das características.

A implantação de uma funcionalidade pode ser considerada uma derivação do modelo de características, ou um conjunto de valores de características que descrevem uma funcionalidade característica. A capacidade de comparar as características dos requisitos de cada projeto, além de apoiar o processo de eliciação de requisitos, facilita o processo de identificação de requisitos reutilizáveis em projetos distintos.

A Figura 12 apresenta o diagrama de características de um portal de entrada em um projeto de implantação de um software mediador. Na figura 13, tem-se a decomposição de uma das características encontradas no portal de entrada, apresentada em um segundo diagrama de características da rota de coletas.

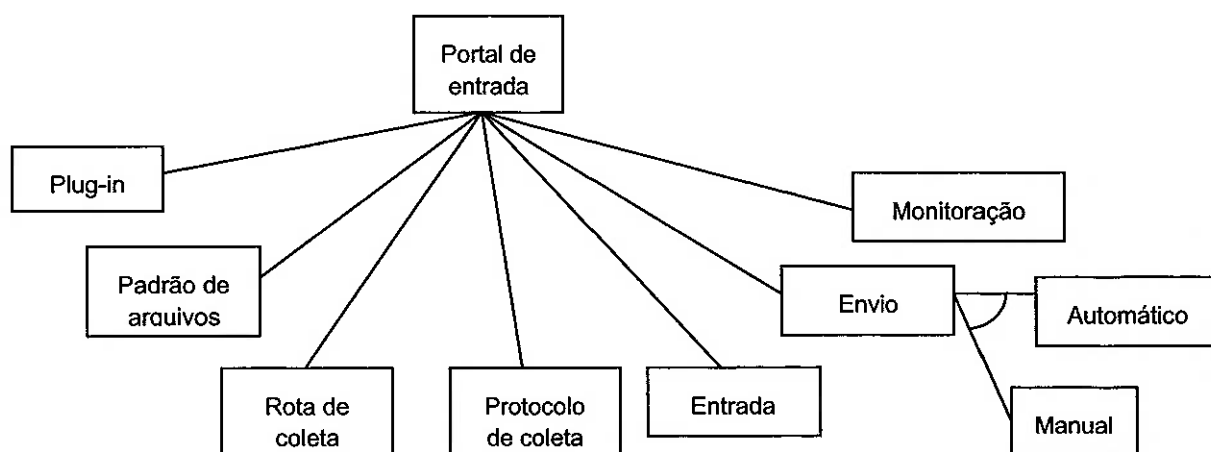


Figura 12 - Diagrama de características portal de entrada

Nas figuras 12 e 13, as linhas que conectam entidades ligadas por um arco representam alternativas de uma característica. Por exemplo, a característica Envio pode ser: Automático ou Manual.

As linhas terminadas por uma circunferência representam características opcionais do domínio-alvo. Por exemplo, a característica Rota Secundária não é uma característica fundamental para a existência da Rota de Coleta.

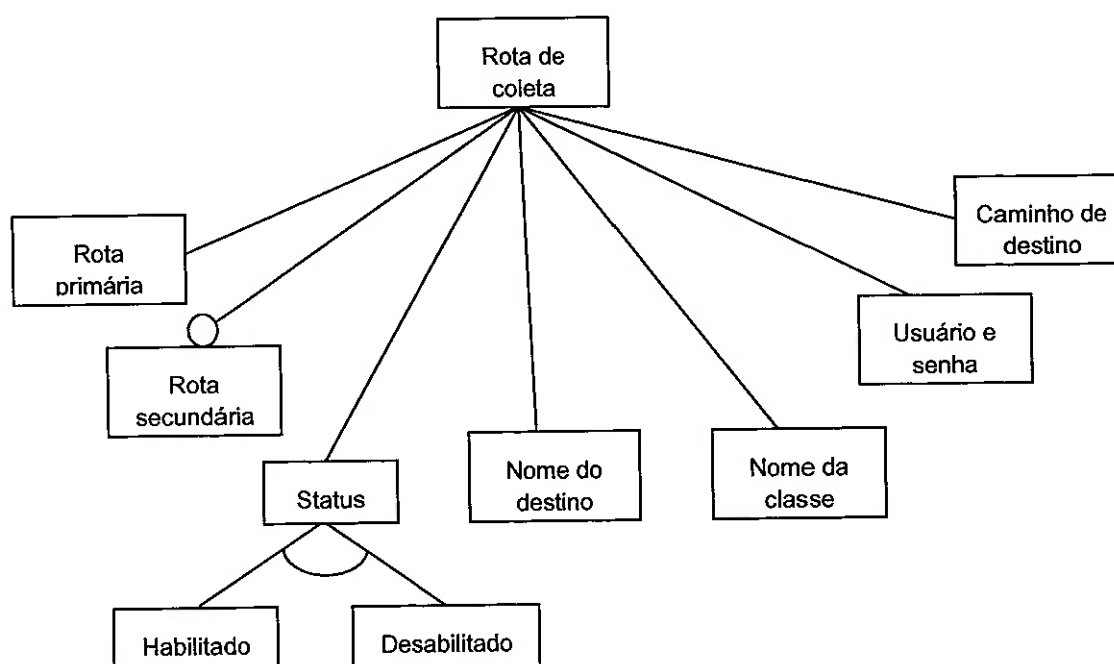


Figura 13 - Diagrama de características rota de coleta

As características são relacionadas principalmente umas às outras através de suas regras de composição, de maneira a definir os limites de cada característica. Existem dois tipos de regras de composição (1) características que requerem a existência de outra característica; (2) características que não podem coexistir com outra característica dada. A representação textual dessas regras se dá pela composição das regras das características como segue:

```
<característica1> ('requer' | 'não coexiste') <característica2>
```

Um exemplo de regra de composição utilizada em um portal de entrada é:

```
<Rota primária> 'requer' <nome de destino>
```

O grande problema da definição da composição das regras é quando devemos definir regras que não sejam percebidas imediatamente. Isso pode ser percebido quando se define as regras de composição do *plug-in*:

```
<Plug-in Name> 'requer' < Name >
<File Transfer Protocol> 'requer' < Transfer Protocol Type >
<File Access Descriptor> 'requer' < File Method >
<File Name Pattern> 'requer' < Id > and <Description> and
<Context>
<Type Output> 'não coexiste' <Input Portal>
<Data Record Type> 'requer' <Data Record Script>
```

Regras de composição geralmente são derivadas da experiência anterior que se tem dessas características em sistemas que apresentam as mesmas características do sistema em análise.

Também se devem incorporar ao modelo de características todas as definições que surgem ao longo do projeto de maneira a racionalizar as opções escolhidas no meio de diversas alternativas. Segue uma demonstração da utilidade do registro de problemas e decisões.

Problema: Espaço em disco insuficiente no sistema de *downstream*

Descrição: O *Dispatch Method* é a maneira que o portal realizará a coleta de arquivos e em consequência iniciará o processamento dos arquivos pelo sistema de mediação.

Iniciada em: Coleta de dados

Decisão A: Manual

Descrição: A Coleta somente é feita com a intervenção manual do operador do sistema.

Racionalização: O Usuário precisa monitorar a chegada de arquivos enviados pelo sistema de *upstream* e o espaço disponível no sistema de *downstream*.

Decisão B: Automático

Descrição: A Coleta é feita toda vez que houver envio de dados pelo sistema de *upstream*.

Racionalização: O Usuário precisa monitorar espaço disponível no sistema de *downstream* freqüentemente.

Decisão C: Agendado

Descrição: A Coleta é feita de acordo com o período agendado.

Racionalização: O Usuário pode adaptar sistema de *downstream* para fazer limpeza de dados antes dos períodos de envio, garantindo espaço em disco para armazenar dados processados recentemente.

A experiência trazida ao longo de outros projetos implantados é de grande valor ao projeto de implantação, pois permite o reuso. Por isso, a utilização de um catálogo de características de projetos anteriores é uma ferramenta que pode trazer muitos benefícios aos novos projetos, principalmente quando se trata de reuso de requisitos, pois é fonte de apoio às decisões que deverão ser tomadas nas futuras implantações similares. Porém, não é sempre que o sistema em implantação apresenta esse tipo de documentação. Um exemplo de catálogo de características é apresentado na Tabela 2:

Tabela 2 – Catálogo de características tipo de dado versus Central Telefônica

	Central A	Central B	Central C	Central D
ASCII	Compatível	Compatível	Compatível	Compatível
BELLTP	Compatível	Não compatível	Não compatível	Não compatível
BPLX	Somente v3.0	Compatível	Não compatível	Não compatível
DPM	Não compatível	Compatível	Não compatível	Não compatível
EBAF	Compatível	Compatível	Compatível	Não compatível
EMR	Não compatível	Não compatível	Compatível	Não compatível

RADIUSB	Compatível	Compatível	Compatível	Compatível
TRAININ	Compatível	Compatível	Compatível	Compatível
TRNICT	Somente v3.0	Não compatível	Compatível	Não compatível

4.2.5 Modelo Funcional

No método FODA o modelo funcional identifica as semelhanças funcionais da aplicação no domínio. O modelo também procura identificar e comparar diferenças entre as aplicações relacionadas. Esse modelo abstrai e representa as funções comuns e as funções distintas. Dessa maneira, a aplicação pode ser vista como uma adaptação ou refinamento do modelo.

O desenvolvimento do modelo funcional depende do modelo de características e do relacionamento de entidades. As peculiaridades do modelo de características impõem parâmetros ao modelo funcional refinando e representando funções alternativas ou opcionais. Enquanto o modelo de características é utilizado para comunicar os requisitos ao analista e usuário, o modelo funcional em conjunto com o modelo de características provê comunicação entre o analista e o projetista de software. O modelo de relacionamento de entidades, por sua vez, tem papel de guia na especificação de dados da visão funcional. A visão operacional do modelo funcional pode ser baseada em técnicas de representação de fluxo de dados. Os fatores que esta visão estabelece são: funções e aplicações do domínio, suas entradas e saídas, dados internos e estrutura de dados e fluxo de dados entre funções. O modelo funcional de um portal de entrada é representado na Figura 14:

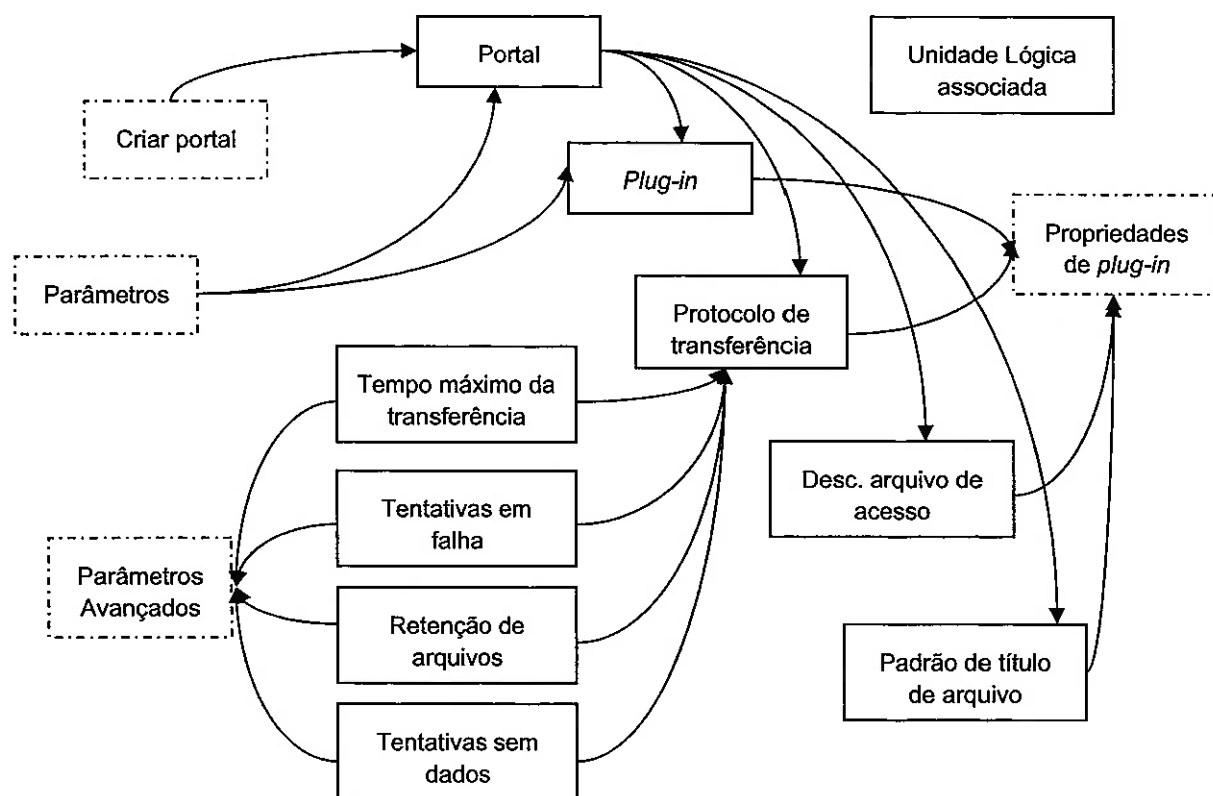


Figura 14 - Modelo Funcional - Fluxo de dados

4.2.6 Dicionário de terminologia do domínio

As terminologias de domínio são utilizadas no método FODA para padronizar a terminologia que descreve o domínio. O dicionário é especialmente crítico em um domínio novo, em que os padrões não sejam preexistentes. O dicionário resulta de informações coletadas durante todas as fases de análise. As características identificadas devem ser nomeadas e qualquer conflito deve ser solucionado. Os sinônimos das características devem ser registrados no dicionário de terminologia do domínio. Na Tabela 3 apresenta-se o dicionário de terminologia do domínio, para os termos mais utilizados quando analisando um portal de entrada a ser implantado.

Tabela 3 – Dicionário de terminologia

Aquisição	Referência aos portais de entrada
Código de erro	Código gerado pela aplicação que é incluído nos arquivo de erro da plataforma
Distribuição	Referência aos portais de saída
Função	A função define os softwares utilizados para coletar ou enviar dados de entrada ou saída
Intervalo de retenção de arquivos	Período definido onde os arquivos serão retidos no mediador
Módulo lógico	Padrão definido pelo usuário para controlar a mediação
Operação	Função baseada em linguagem C que suporta individualmente a conversão de dados, filtragem, validação de dados, tributar e enviar arquivos.
Portal	Conjunto de parâmetros definidos pelo usuário que o mediador utiliza para receber ou coletar dados de um sistema de entrada externo
Protocolo	Conjunto de regras de comunicação presentes numa camada de comunicação
Protocolo de transferência de dados	Protocolo TCP/IP utilizado para transferência de arquivos entre sistemas
Saída	Transmissão de dados já processados e acumulados no mediador
Sistema de <i>downstream</i>	Termo genérico para um sistema externo que recebe dados do mediador
Sistema de <i>upstream</i>	Termo genérico para um sistema externo que envia dados ao mediador
<i>Throughput (bytes/segundo)</i>	Número médio de <i>bytes</i> transferidos através de uma linha de comunicação por segundo
Tipo de dado/registro	O valor do tipo de dado determina o tipo padrão

	de dados utilizados para leitura e escrita de um <i>plug-in</i>
--	---

4.3 Reuso de requisitos na implantação

O objetivo principal do trabalho é estudar algumas possibilidades da reutilização de requisitos em projetos de implantação de produtos de software, em especial do software de mediação, empregando um método de elicitação de requisitos para identificar os requisitos candidatos ao reuso. Como justificado no início deste capítulo, o método FODA traz uma análise detalhada de um domínio definido do produto de software, viabilizando a análise aprofundada de vários aspectos de um mesmo requisito, o que lhe permite incorporar uma orientação ao reuso de requisitos.

Em termos gerais, aplica-se o método FODA para definir junto ao cliente os requisitos de um novo projeto de implantação de um sistema de mediação. Depois de feita essa primeira análise, no passo seguinte o analista de domínio e a equipe de desenvolvedores e especialistas do produto analisam quais requisitos são coincidentes entre outros projetos já entregues e também quais partes desenvolvidas anteriormente podem ser reutilizadas.

A identificação de requisitos reutilizáveis será apresentada através do exemplo de escolha de um *plug-in*. Nesse exemplo, utilizam-se somente duas ferramentas do método FODA na modelagem do domínio: o diagrama de características e o catálogo de características.

Na visão do software mediador, um *plug-in* é a combinação de elementos que descreve a transferência de dados na entrada e na saída, incluindo:

- a. Dados/tipos de registro;
- b. Método de acesso aos arquivos;
- c. Convenção de nome de arquivos;
- d. Softwares binários para a transferência de dados;
- e. *Scripts* para acesso aos registros.

Essencialmente *plug-ins* são parâmetros utilizados por portais de entrada e saída para referenciar dados de entrada e de saída. Todo portal tem um *plug-in* associado, que habilita o portal e determina protocolos de comunicação, rotulação de arquivos e rota.

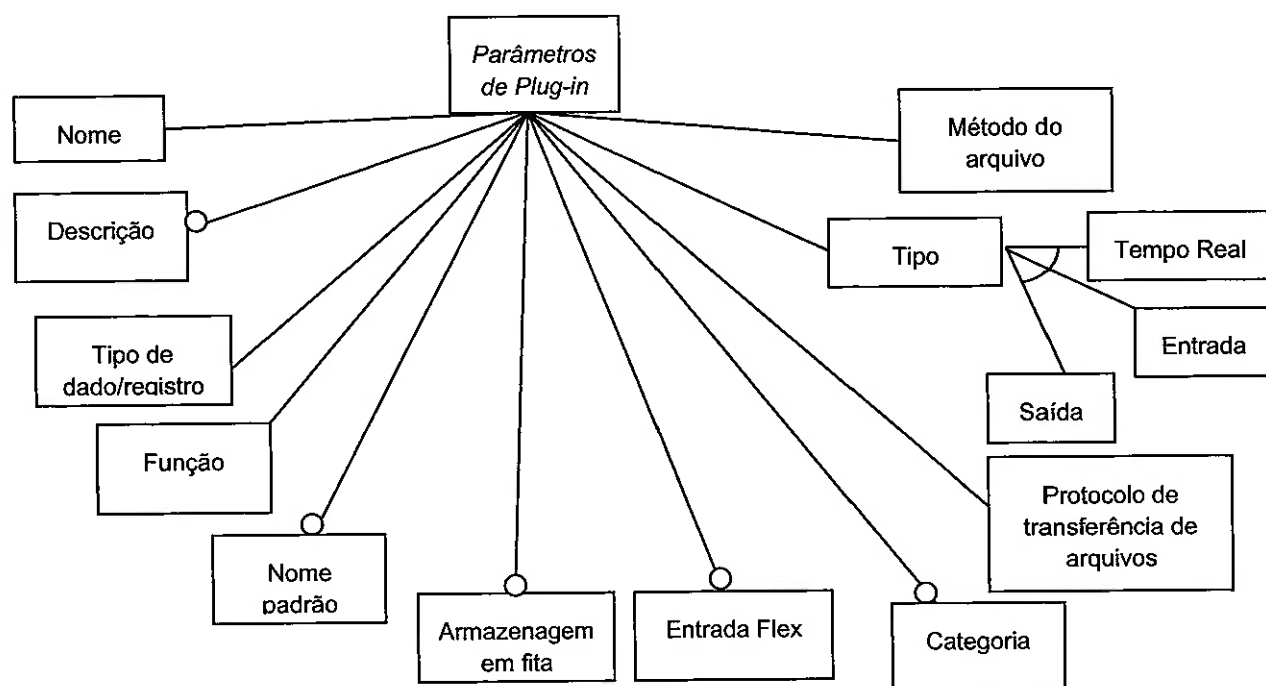


Figura 15 – Diagrama de características plug-in

No diagrama da Figura 15 podem-se identificar características de um *plug-in* necessárias para a sua configuração. Como representado na Figura 12, um *plug-in* é uma característica fundamental na definição de um portal de entrada. Portanto, pode-se considerar que tanto o diagrama da Figura 12 e o diagrama da Figura 15 serão os mesmos para qualquer projeto de implantação do software de mediação analisado por esse texto. Cada uma das características da Figura 15 deve ser avaliada para a detecção de requisitos reutilizáveis.

Para os fins do presente trabalho, somente será verificada a característica definida como Tipo de dado/registo. Conforme apresentada na Tabela 3, Tipo de dado/registo é o tipo de dado que determina o padrão de dados utilizados para a leitura e escrita de um *plug-in*. Para cada formato de dado existente é feito um *script* para a sua interpretação. Através da identificação dos tipos de dados que são necessários em cada projeto de implantação podem-se definir quais tipos de dados

serão utilizados em cada projeto e, desta forma, pode-se reutilizar os *scripts* desenvolvidos anteriormente para tratar cada tipo de dado. As Tabelas 4 e 5 apresentam os tipos de dados utilizados pelos sistemas de *upstream* para os projetos de implantação X e Y, respectivamente.

Tabela 4 – Catálogo de características tipos de dados e sistemas de upstream
Projeto X

	Central A	Central B	Central C
ASCII	Compatível	Compatível	Compatível
BELLTP	Compatível	Não compatível	Não compatível
BPLX	Somente v3.0	Compatível	Não compatível
DPM	Não compatível	Compatível	Não compatível
EBAF	Compatível	Compatível	Compatível
EMR	Não compatível	Não compatível	Compatível
RADIUSB	Compatível	Compatível	Compatível
TRNICT	Somente v3.0	Não compatível	Compatível

Tabela 5 – Catálogo de características tipos de dados e sistemas de upstream Projeto
Y

	Central A	Central C	Central D
ASCII	Compatível	Compatível	Compatível
BELLTP	Compatível	Não compatível	Não compatível
EBAF	Compatível	Compatível	Não compatível
EMR	Não compatível	Compatível	Não compatível
RADIUSB	Compatível	Compatível	Compatível
TRAININ	Compatível	Compatível	Compatível
TRNICT	Somente v3.0	Compatível	Não compatível

A partir da análise das Tabelas 4 e 5, pode-se definir inicialmente que o *script* desenvolvido durante a implantação do projeto X, para tratar dados em formato ASCII para a central telefônica do fabricante A, pode ser reutilizado no projeto Y, assim

como todos os outros tipos de dados que forem elencados como necessários no projeto Y.

Caso se avance na modelagem de domínio – algo que não será realizado nesse trabalho – outros dois domínios do método FODA são definidos: a análise de contexto e a modelagem da arquitetura. Pode-se então refinar cada vez mais os requisitos que poderão ser completamente reutilizados em projetos de implantação similares. Requisitos semelhantes e que necessitem de pequenos ajustes para adaptação em novos projetos, trazem também, desta maneira, ganhos no desenvolvimento e na implantação do produto de software.

É importante ressaltar que, para o reuso de requisitos, é necessária a padronização do método de elicitação de requisitos empregado, para que os requisitos sejam comparados e analisados no mesmo nível de detalhe. Cabe à organização definir um padrão sobre métodos que serão utilizados. A padronização de um método de elicitação de requisitos traz outros benefícios além da análise do reuso de requisitos. A produção de documentos padronizados, assim como a padronização das fases de implantação e metodologia de trabalho, facilita a tomada de decisões e a identificação de pontos de melhoria e falhas em projetos.

5. Considerações Finais

A utilização de um método de elicitação de requisitos é um passo fundamental para o estabelecimento de requisitos em áreas onde reuso de produtos de software é desejado. Além disso, os métodos de elicitação de requisitos têm ferramentas de apoio que podem ser utilizadas com vários propósitos além do reuso de requisitos. Como auxílio na identificação de requisitos e divulgação de requisitos entre as partes interessadas do projeto, o método deve utilizar uma linguagem padrão objetiva e inteligível para os diferentes níveis de indivíduos que terão acesso às fases do projeto de implantação do produto de software.

Como apresentado no Capítulo 4, a utilização do método FODA oferece recursos que podem ser aplicados de modo sistemático em projetos de implantação de software de mediação para a elicitação de requisitos potencialmente reutilizáveis em diferentes implantações. Com a utilização do método, produz-se a documentação do histórico do projeto sobre elementos desenvolvidos e ações tomadas no projeto. Esse histórico servirá como contribuição aos futuros projetos de implantação, os quais podem ser aperfeiçoados em cada novo projeto de implantação, contribuindo para a melhoria da qualidade do serviço entregue.

No presente trabalho, mesmo com as limitações impostas pelas hipóteses simplificadoras para o modelo conceitual da implantação do software de mediação descrito, bem como a aplicação parcial do método, foi possível identificar requisitos potencialmente reutilizáveis em implantações do produto.

Finalmente, pode-se dizer que este estudo preliminar permite concluir que a utilização do método FODA pode trazer benefícios ao projeto de implantação, uma vez que define de forma clara e coerente uma linguagem padrão de comunicação entre as partes interessadas no projeto de implantação, e possui técnicas que permitem a validação dos requisitos identificados e, sobretudo, auxilia na identificação de requisitos reutilizáveis em futuras implantações do produto de software.

Referências Bibliográficas

- [1] (DA SILVA, S. M. A.; BONIN, Marcos R.; PALUDO, Marco A.). Levantamento de Requisitos Segundo o Método Volere. 2006.
- [2] (DAVEY, Bill ; COPE, Chris). *Requirements Elicitation – What is missing?*. 2008.
- [3] (HEISKARI, Juho) *Role of User Involvement in Software Product Development*. 2008
- [4] (HOMOD, M. Sami ; RINE, David C.) *Building Requirements Repository using Requirements Transformation Techniques to Support Requirements Reuse*. Estados Unidos. 1998
- [5] (HUBBARD, R.; MEAD, N.; SCHROEDER, C.) *An Assessment of the Relative Efficiency of a Facilitator-Driven Requirements Collection Process with Respect to the Conventional Interview Method*. 2000
- [6] (KANG, Kyo C.; COHEN, Sholom; HESS, James A.; NOVAK, William; PETERSON, A. Spencer) *Feature-Oriented Domain Analysis (FODA) Feasibility Study*. 1990
- [7] (MERWE, Alta van der; CRONJE, Johannes; KOTZÉ, Paula) *The Functionality of a Requirements Elicitation Procedure Developed for Process Modeling within the Higher Education Application Domain*. 2005.
- [8] (MOULDING, M.R. e SMITH, L.C.) *Formalizing a CORE requirements model in the air traffic control domain*. 1992
- [9] (SINDRE, Guttorm e Opdahl, ANDREAS L.). *Capturing Security Requirements through Misuse Cases.*: 2001.
- [10] (VILLEGAS, Oscar L.; LAGUNA, Miguel A.) *Requirements Reuse for Software Development.*. 2001.

Referências consultadas

- [1] (CHRISTEL, Michael G.; KANG, Kyo C.). *Issues in Requirements Elicitation*. 1992
- [2] (EBERLEIN, Chethana Kuloor Armin). *Requirements Engineering for Software Product Lines*. 2002
- [3] (Intec). *Introdução ao Inter-mediatE*. 2006.
- [4] (LOBO, Lester). *Analysis and Evaluation of Methods for Activities in the Expanded Requirements Generation Model*. 2004
- [5] (MEAD, Nancy R.). *Requirements Elicitation Introduction*. Estados Unidos. 2006.
- [6] (PRESSMAN, Roger S.). *Engenharia de Software*. 1995.
- [7] (SOMMERVILLE, Ian.). *Engenharia de Software*. 2005.

APÊNDICE A – Atributos das entidades

- 1) Atributos do portal de entrada
 - a) Coletar
 - b) Configurar
 - c) Entregar
- 2) Módulo lógico
 - a) Agrupar entradas, operações e saídas
- 3) Parâmetros de configuração
 - a) Protocolo de transferência de dados
 - b) Frequência de coleta de dados
 - c) Rota de aquisição
 - d) Padrão de dados de coleta
 - e) Plug-ins
- 4) Coletor de dados
 - a) Inicia transferência de dados
- 5) Processos de execução
 - a) Inicia operações
- 6) Operações
 - a) *Scripts* de tratamento de dados