ADRIANA HERDEN

UPKDD: UM PROCESSO PARA DESENVOLVIMENTO DE SISTEMAS DE DESCOBERTA DE CONHECIMENTO EM BANCO DE DADOS

MARINGÁ

2007

ADRIANA HERDEN

UPKDD: UM PROCESSO PARA DESENVOLVIMENTO DE SISTEMAS DE DESCOBERTA DE CONHECIMENTO EM BANCO DE DADOS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Estadual de Maringá, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientadora: Profa. Dra. Maria Madalena Dias

MARINGÁ

2007

H541 Herden, Adriana

UPKDD: um processo para desenvolvimento de sistemas de descoberta de conhecimento em banco de dados. / Adriana Herden . -- Maringá: Universidade Estadual de Maringá, 2007.

171 f.: il. , 30 cm.

Orientadora: Profa. Dra. Maria Madalena Dias

Dissertação (Mestrado) - Programa de Pós-Graduação em Ciência da Computação. Universidade Estadual de Maringá, 2007.

1. Engenharia de software – experimentos 2. Banco de dados – apoio à decisão 3. Engenharia de software – metodologia. 4.Data warehousing. 5. Banco de dados – mineração de dados I. Dias, Maria Madalena, orient. II. Universidade Estadual de Maringá. III.Título.

CDD 22.ed.

005.74

ADRIANA HERDEN

UPKDD: UM PROCESSO PARA DESENVOLVIMENTO DE SISTEMAS DE DESCOBERTA DE CONHECIMENTO EM BANCO DE DADOS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Estadual de Maringá, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Aprovada em 04/09/2007.

BANCA EXAMINADORA

Profa. Dra. Maria Madalena Dias
Universidade Estadual de Maringá – DIN/UEM

Profa. Dra. Itana Maria de Souza Gimenes
Universidade Estadual de Maringá – DIN/UEM

Profa. Dra. Sueli de Fátima Poppi Borba
Universidade Paranaense – UNIPAR-Cianorte

Dedico este trabalho a DEUS.

AGRADECIMENTOS

Primeiramente agradeço a DEUS, pela sua luz, pela sua benção e pela graça.

À minha família que sempre soube me incentivar e me apoiar em todos os momentos desta caminhada.

À minha orientadora Madalena pela orientação, dedicação e paciência de sempre, que em muitas situações foi muito mais que orientadora.

Aos professores do Curso de Pós-Graduação em Ciência da Computação, pela orientação e ensino presente em todos os momentos desde início do mestrado. Especialmente a Prof.ª Elisa Huzita que sempre teve palavras otimistas e também de delimitação da minha pesquisa.

Aos amigos da minha turma, que puderam me mostrar o valor da amizade, companheirismo, profissionalismo em meio à correria do curso e também da concorrência explícita da área de computação. Especialmente aos meus amigos Jô, Flávio, Adriano, Lafaiete, Everson, Rafael Gatto e José Rafael. Aos amigos dos grupos de desenvolvedores que participaram efetivamente no meu estudo de caso. Aos alunos Paulo Alonso e Marlos.

Aos amigos da UTF, que puderam me auxiliar com palavras de incentivo, solidariedade, orientações, correções compreendendo as etapas difíceis que o caminho do mestrado exige. Especialmente aos professores Gabriel, Marcos Vallim, Luciana Hernandez, Pozza, Vanderley, Thesko, Sérgio, Geromel e, também, à Coinf e à Direção do Campus.

Aos amigos que me apoiaram em cada uma das etapas, que muitas vezes achei impossível de serem finalizadas. Agradeço imensamente a hospedagem do Bruno, Janira e Mariana, da mesma forma a hospedagem da Madalena e Nardênio. As orações que me sustentaram, sem as quais eu não teria terminado este trabalho, especialmente a Dona Leonor, ao Movimento dos Focolares, a Rose Ribas, a Taciana, a Sílvia, ao Zé Antônio e ao Cássio.

"Confiai no Senhor perpetuamente,
porque o Senhor Deus é uma rocha
eterna". Isaías 26:3.

HERDEN, Adriana. **UPKDD**: um processo para desenvolvimento de sistemas de descoberta de conhecimento em banco de dados. Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Maringá.

RESUMO

O processo de descoberta de conhecimento em banco de dados (KDD - *Knowledge Discovery in Databases*), também conhecido apenas como Mineração de Dados, tem sido estudado desde 1989 na indústria e na academia como estratégia de apoio à decisão. Através de uma seqüência de passos iterativos e interativos, padrões de conhecimento ocultos são descobertos para auxiliar o tomador de decisão. Cada vez mais os processos de desenvolvimento de engenharia de software abordam aspectos relacionados à interação com o usuário, mecanismos que aumentem a produtividade e estimem de maneira realística orçamentos e prazos. O processo unificado de desenvolvimento de software (*Unified Process* - UP), além de propor soluções para esses problemas, também é considerado um *framework* de processo para ser otimizado, conforme necessidades da aplicação. Assim sendo, este trabalho apresenta uma adaptação do UP com ênfase nas necessidades específicas do desenvolvimento de sistemas de descoberta de conhecimento em banco de dados. Existem alguns requisitos para esse tipo de aplicação como: o sistema construído sob a visão de banco de dados e não enfatizando o tratamento das funcionalidades implementadas pelas linguagens de programação e os processos KDD existentes não abrangem a elaboração de componentes e sim a utilização de aplicações KDD. Assim, o desafio principal deste trabalho é adaptar os três passos importantes do processo KDD (pré-processamento, mineração de dados e pós-processamento) em uma perspectiva transformacional inerente ao UP.

Palavras-chave: Descoberta de Conhecimento em Banco de Dados, Processo Unificado, Engenharia Experimental.

HERDEN, Adriana. **UPKDD**: a process for development the systems of the Knowledge Discovery in Database. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Maringá.

ABSTRACT

The knowledge discovery process in database (KDD - Knowledge Discovery in Databases), also known barely as Data Mining, has been studied since 1989 in the industry and in the academy like strategy of support to the decision-making. Through a sequence of activities iterative and interactive, standards of hidden knowledge are discovered for help the end users. More and more the software engineering development process approach aspects related to the interaction with the user, mechanisms that increase the productivity and estimate of real way budgets and terms. The unified process of software development (UP), beyond propose solutions for those problems, also is considered a framework of process for to be optimized in agreement needs of the application. Like this being, this work presents an adaptation of the UP with emphasis in the specific needs of development of KDD systems. Some requirements for that kind of application exist like: the system built under the vision of database and not emphasizing the handling of the functionalities implemented by the programming languages and the processes KDD existing do not include the elaboration of components and yes the utilization of application KDD. Like this the main challenge of this work is going to adapt the three important activities of the process KDD (pre-processing, data mining and pos-processing) in a perspective transformer inherent to the UP.

Key-words: Knowledge Discovery in Database Unified Process, Experimental Engineering.

LISTA DE ILUSTRAÇÕES

QUADROS

FIGURAS

DISCIPLINA DE REQUISITOS

DISCIPLINA DE ANÁLISE

DISCIPLINA DE PROJETO

FASES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

(i,i)	iterativo e interativo
(nl)	não linear
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CASE	*Computer-Aided Software Engineering*
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
DW	*Data Warehouse*
EC	Elemento-Chave
ETL	*Extraction/Transformation/Loading*
GQM	*Goal/Question/Metric*
H0 / H1	Hipóteses Estatísticas
ISO	*International Organization for Standardization*
KDD	*Knowledge Discovery in Database*
MOF	*Meta-Object Facility*
MSF	*Microsoft Solutions Framework*
O.O.	Orientação a Objetos
ODL	*Object Definition Language*
OLAP	Processamento Analítico *On-line*
OMG	*Object Management Group*
PML	*Process Modeling Language*
PSEE	*Process Centered Software Engineering Environment*
QIP	*Quality Improvement Paradigm*
RUP	*Rational Unified Process*
SPEM	*Software Process Engineering Metamodel Specification*
SUA	(S)Utilização (U)Utilidade (A)Adequação
UML	*Unified Modeling Language*
UP	*Unified Process*
UPKDD	*Unified Process for Knowledge Discovery in Database*
UPM	*Unified Process Modeling Language*
UTF	Universidade Tecnológica Federal
XML	*Extensible Markup Language*
XP	*Extreme Programming*

SUMÁRIO

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Os processos de engenharia de software possuem artefatos, responsabilidades e atividades bem definidas, possibilitando adequação às necessidades específicas. Esses processos são geralmente voltados ao desenvolvimento de sistemas de apoio a atividades operacionais de uma organização, não atendendo totalmente às necessidades de sistemas de apoio à tomada de decisão.

A descoberta de conhecimento em banco de dados surgiu com o objetivo de extrair conhecimentos desconhecidos e interessantes para apoio à tomada de decisão.

O processo de descoberta de conhecimento em banco de dados, ou simplesmente KDD (*Knowledge Discovery in Databases*), tem o foco na descoberta de padrões ocultos em grandes bancos de dados, porém as atividades para a obtenção da descoberta não são explícitas. Este processo identifica as exigências mínimas para obter conhecimento podendo ser dividido em três passos básicos que são: pré-processamento, mineração de dados e pós-processamento.

Durante o pré-processamento, diferentes fontes de dados são integradas e os dados são transformados de acordo com os objetivos estabelecidos para a descoberta de conhecimento. No passo de mineração de dados, são aplicados algoritmos que buscam extrair conhecimentos entre os dados integrados e transformados. O pós-processamento possibilita a análise dos resultados obtidos na mineração de dados e sugere maior iteração do processo KDD e interação do usuário com o sistema.

Esses passos do processo KDD[1] podem ser vistos como os requisitos domínio[2] de um sistema KDD[3] e não como diretrizes de um processo de software de aplicação KDD[4] .

Desta forma, os modelos de processo KDD propostos por diversos autores como Fayyad, Piatetsky-Shapiro e Smyth (1996), Han e Kamber (2001), Brachman e Anand (1996)

---

[1] Passos que conduzem a descoberta de conhecimento em banco de dados.
[2] Requisitos do Domínio são requisitos que se originam do domínio de aplicação do sistema e refletem características desse domínio (Sommerville, 2003). Neste caso o domínio de KDD.
[3] Sistema KDD representa a implementação computacional para os passos do processo KDD.
[4] Processo de desenvolvimento de sistema KDD representa atividades sistemáticas para a construção de software KDD.

e também Reinartz (1999) definem e contribuem positivamente para identificação das principais funções de um sistema KDD para a obtenção de conhecimento em banco de dados.

Portanto, existe carência de procedimentos no aspecto de sistematização do processo de desenvolvimento de aplicações KDD, por exemplo, identificação de modelos a serem gerados antes da implementação das aplicações.

Neste contexto, percebe-se a inexistência de um processo sistemático, em que devem ser definidos claramente as atividades, os artefatos e os papéis necessários em cada etapa do desenvolvimento de aplicações KDD, assim como os processos de engenharia de software possuem. É interessante que este seja baseado no Processo Unificado de desenvolvimento de software (UP), proposto por Jacobson, Booch e Rumbaugh (1999), pelo fato de ser considerado um processo estabelecido e que se adapta de acordo com o domínio da aplicação.

1.2 OBJETIVOS

Esta dissertação tem como objetivo principal a definição de um processo para desenvolvimento de aplicações KDD por meio da adaptação dos elementos-chave que compõem UP.

1.2.1. Objetivos específicos

- Realizar revisão bibliográfica;
- Especificar os módulos do processo KDD;
- Estabelecer os elementos-chave e *workflows* para esta aplicação;
- Elaborar versão especializada do UPKDD (*Unified Process for Knowledge Discovery in Database*);
- Realizar estudo de caso.

1.3 CONTEXTUALIZAÇÃO

As aplicações KDD passaram de simples relatórios gerenciais feitos por linguagens de consulta em arquivos simples, a grandes e sofisticadas aplicações KDD que demandam esforço de analista de negócio a usuários finais do sistema.

Os modelos de processo KDD, propostos atualmente, referem-se aos passos mais relevantes para obtenção de padrões e relacionamentos de dados estruturados da organização, ou ainda, como as exigências mínimas adequadamente separadas, de acordo com objetivos específicos necessários na busca de conhecimento úteis em grandes bancos de dados. Porém, a definição dos passos básicos do processo de busca, ainda que relevante, pode ser um método insuficiente para o desenvolvimento de requisitos funcionais[5] de aplicações KDD.

Atualmente, a busca por conhecimentos úteis faz parte de decisões estratégicas e gerenciais da maioria das empresas de grande e médio porte. O uso de ferramentas que auxiliam esse processo tem aumentado nos últimos anos, representando mais vendas e mais desenvolvimento de software desse tipo, que apóiam as decisões executivas. Exemplo disso é a SPSS[6] fornecedora de soluções de mineração de dados e análise preditiva, que registrou receita líquida de US$58,3 milhões, resultados do terceiro trimestre de 2005 provenientes de novas licenças. Outro exemplo é a IBM[7] que mostra tendências do trabalho da área de finanças, 69% correspondem ao processamento de transações e atividades de controle e os diretores financeiros esperam diminuir essa porcentagem para 55% em dois anos e dedicar-se mais às atividades estratégicas.

A profissão de desenvolvedor de aplicações KDD é relativamente nova e, conseqüentemente, o desenvolvedor de software precisa definir e organizar suas atividades, para tal necessita seguir um processo de desenvolvimento de software. O UP propõe a definição de um processo que aborda um conjunto de atividades executadas por papéis específicos para transformar um conjunto de requisitos do cliente em um sistema de software. UP torna-se adequado para o desenvolvimento de sistemas KDD, por ser padrão adequado para o desenvolvimento de sistemas orientados a objeto. Outros processos orientados a objeto,

---

[5] Requisitos de sistema de software são classificados como Funcionais ou Não Funcionais. Requisitos Funcionais são declarações de funções que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações. Requisitos Não Funcionais são restrições sobre os serviços ou as funções oferecidas pelo sistema (Sommerville, 2003).

[6] http://www.spss.com.br/press/receitarecorde.htm

[7] http://www.ibm.com/news/br/pt/2006/02/20-02-2006.html

como Catalisys e UML Components identificados por Werner e Braga (2005), não definem claramente o uso de arquitetura de software como direção do processo de desenvolvimento, assim como o UP define. Os sistemas KDD existentes destacam a importância da definição da arquitetura de software para o desenvolvimento como DBMiner e I-MIN descritas por Gupta, Bhatnagar e Wasan (2005).

O desenvolvimento de sistemas KDD tem sido aperfeiçoado a cada dia, até então houve maior interesse no entendimento de algoritmos de aprendizado de máquina, fornecimento de serviços e técnicas de mineração e visualização analítica dos dados, sendo características que não abrangem o processo de desenvolvimento de aplicações KDD.

1.4 METODOLOGIA DE PESQUISA

Esta pesquisa foi realizada em cinco etapas, conforme representadas na Figura 1.1, a seguir: revisão bibliográfica, especificação dos módulos do processo KDD, estabelecimento dos elementos-chave e *workflows*, elaboração da versão especializada do UP para sistemas KDD (**UPKDD**) e realização de estudo de caso.

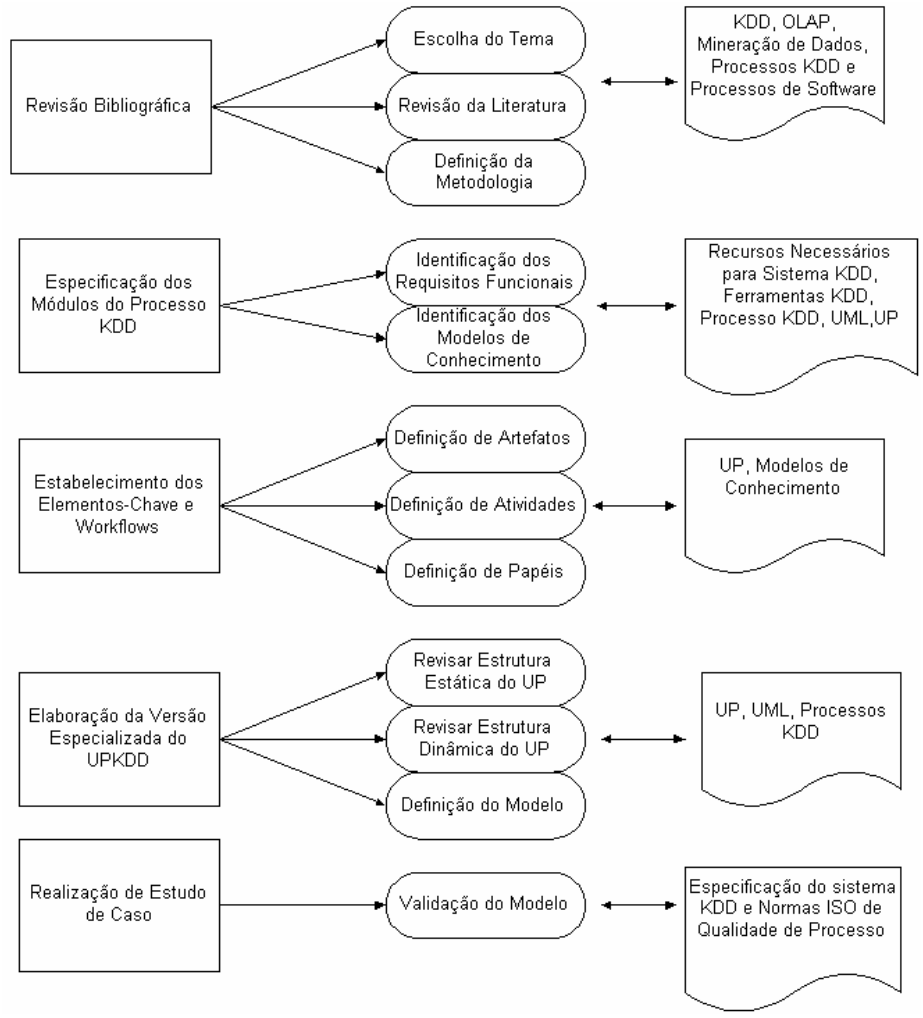


Figura 1.1 - Metodologia de pesquisa

1.4.1 Revisão bibliográfica

Esta fase teve como atividade principal estudar os processos KDD existentes e estudar o UP, assim como a experimentação em engenharia de software e métodos de modelagem de processo.

1.4.2 Especificação dos módulos do processo KDD

Primeiramente foi necessário identificar os requisitos funcionais do processo KDD por meio do entendimento de aplicações KDD. Os módulos compreendidos são: Pré-Processamento, Mineração de Dados e Pós-Processamento. Esta fase representa o entendimento das funcionalidades do processo KDD, utilizando modelos e baseando-se no estudo das capacidades e limitações das aplicações KDD existentes. Este entendimento serviu de mecanismo de abstração dos comportamentos de cada etapa da descoberta de conhecimento, com a finalidade de posteriormente serem utilizados como recursos na definição dos elementos-chave do processo proposto.

1.4.3 Estabelecimento de elementos-chave e *workflows*

Nesta fase foram ser definidos claramente os elementos-chave para o processo proposto: Artefatos, Papéis e Atividades. As representações dos elementos-chave identificados foi feita por meio de modelos que atendem as peculiaridades do processo proposto. Os papéis são baseados nas abordagens indicadas pelos autores Brachman e Anand (1996) e Fayyad, Piatetsky-Shapiro e Smyth (1996). E, por fim, as atividades unem papéis e artefatos. O estabelecimento de fases e *workflows* depende do UP.

1.4.4 Elaboração da versão especializada do UPKDD

Neste estágio do trabalho foi descrito o processo proposto, situando-o entre o comportamento dinâmico do UP e modelo estático dos processos KDD.

1.4.5 Realização de estudo de caso

Neste estágio do trabalho realizou-se um estudo de caso quantitativo para avaliação do processo proposto, usando dois grupos de desenvolvedores como estratégias de medição.

Obteve-se direcionamento da pesquisa após a determinação de passos e recursos necessários para ter um processo personalizado e avaliado para projetos de desenvolvimento de aplicações KDD.

1.5 ORGANIZAÇÃO DO TRABALHO

Além deste capítulo que apresenta a introdução desta dissertação de mestrado, incluindo motivação, objetivos, contextualização e a metodologia de pesquisa utilizada, o trabalho está dividido em mais quatro capítulos.

No Capítulo 2 são apresentados conceitos e características das áreas de pesquisa envolvidas, mais especificamente uma revisão bibliográfica dos processos KDD existentes, com quadro de revisão da literatura, que identifica a característica principal de cada um. Além disso, é apresentada a revisão bibliográfica do UP, justificando sua escolha como parâmetro de paradigma de engenharia de software pela sua estrutura e funcionamento definidos suficientemente. Também são apresentados os fundamentos da modelagem de processos de software, enfatizando o metamodelo SPEM e também os fundamentos da experimentação em engenharia de software. Complementando o capítulo, são apresentados uma classificação de métodos de avaliação experimental e fundamentos da experimentação em engenharia de software. Para finalizar o capítulo são apresentados os principais trabalhos relacionados.

No Capítulo 3 é apresentada a caracterização de uma aplicação KDD, depois os modelos do processo UPKDD, usando a ferramenta Rational Rose com os estereótipos do metamodelo SPEM. A divisão do capítulo foi evolutiva, primeiramente mostra-se a visão geral do processo modelado, depois em cada seção do capítulo é abordada uma Disciplina ou *Workflow* do UPKDD.

No Capítulo 4 são demonstrados o estudo de caso realizado seguindo a abordagem GQM e o teste de hipóteses estatísticas qui-quadrado.

No Capítulo 5 são apresentadas e sumarizadas as principais contribuições, dificuldades encontradas, juntamente com os trabalhos futuros a partir deste trabalho de mestrado.

Para finalizar, são apresentadas as referências mais utilizadas para a elaboração deste documento, assim como os apêndices e anexos necessários.

2 REVISÃO BIBLIOGRÁFICA

São apresentados neste capítulo os conceitos relevantes de busca de conhecimento em banco de dados e também de processos de engenharia de software, que são usados para fundamentação deste trabalho e apoiam a elaboração do processo de desenvolvimento de aplicações KDD.

Além disso, são apresentados os fundamentos da modelagem de processos enfatizando o metamodelo SPEM (OMG, 2005) e os fundamentos da experimentação em engenharia de software. Também são descritos sucintamente estudos relacionados a este trabalho.

Certamente, é importante para a criação da versão especializada do UP, fundamentar as áreas de Descoberta de Conhecimento em Banco de Dados e Processo de Engenharia de Software. Da mesma maneira é relevante modelar o processo seguindo o SPEM e avaliá-lo por um Estudo de Caso Quantitativo.

2.1 PROCESSOS DE DESCOBERTA DE CONHECIMENTO EM BANCO DE DADOS

Atualmente, descoberta de conhecimento em bancos de dados surge como uma estratégia de ação para empresas que possuem grandes bancos de dados. A dimensão de grande é explicada por Fayyad, Piatetsky-Shapiro e Smyth (1996) que considera grande um banco de dados com terabytes (10^{12}bytes) de tamanho. Nesse contexto, decisões empresariais necessitam constantemente de mais recursos computacionais na resolução de problemas relacionados a adequações dos investimentos, identificação de benefícios dos negócios. Assim sendo, justifica-se a presença de aplicações KDD[8], onde dados brutos são transformados em conhecimento útil para análises futuras.

Sistemas de apoio à decisão tiveram sua origem nas décadas de 40 e 50 e partem do princípio básico de analisar o comportamento do negócio com base na busca de dados operacionais a fim de modificar comportamentos da empresa de maneira adequada. Já nas décadas de 60 e 70 promoveu-se a utilização de computadores durante o processo de tomada de decisões. No contexto histórico de Date (2003) é identificada a forma de utilização dos

[8] Aplicações KDD representam sistemas implementados com objetivos de descoberta específicos.

computadores sendo apenas por geradores de relatórios implementados pelas linguagens de consulta da época.

Através da evolução inevitável da tecnologia, surgiram os bancos de dados relacionais nos anos 80, que incentivaram abandonar o uso de arquivos simples e também incentivavam estudos de técnicas na área de apoio à decisão.

Atualmente, a maioria dos bancos de dados possibilita o uso de tecnologia de apoio à tomada de decisão, tais como *data warehouse*[9], processamento analítico *on-line* (OLAP), modelos multidimensionais e mineração de dados. Essas tecnologias, em linhas gerais, objetivam consultar os dados do negócio em ambientes não operacionais para não causar interrupções ou inconsistências, executar algoritmos específicos para extração de informações e também disponibilizar visualizações do conhecimento descoberto.

O conhecimento descoberto, para tomada de decisões, passa por alguns passos antes mesmo de tornar-se conhecimento propriamente dito. A extração de padrões ocultos em banco de dados necessita considerar a hierarquia entre dados, informação e conhecimento e, também, permitir a troca iterativa de estratégia de busca por conhecimento de interesse.

A diferença básica da hierarquia de dados, informação e conhecimento é apresentada por Passos e Goldschmidt (2005) através de representações diferentes. Os dados representam os itens elementares a serem trabalhados, enquanto informação envolve o processamento de dados através de recursos tecnológicos e, por fim, o conhecimento utiliza recursos tecnológicos mais avançados que apóiam a aplicação de técnicas de mineração de dados com o objetivo de descobrir regras e padrões entre os dados.

O termo "descoberta de conhecimento em banco de dados" representa um fundamento para tecnologias de banco de dados, voltado para sistemas de apoio à decisão, pois define os passos fundamentais para descobrir conhecimento oculto relevante em banco de dados. É importante diferenciar que o escopo de KDD não atende as particularidades da descoberta de conhecimento em dados semi-estruturados ou não estruturados, como os bancos de dados multimídia e *webmining* em textos *web*. Diante disso, Fayyad, Piatetsky-Shapiro e Smyth (1996, p. 40) afirma a necessidade emergente de criação de ferramentas computacionais que auxiliem o ser humano na extração rápida de conhecimento e também define KDD como "o processo não trivial de identificação de padrões válidos, novos, potencialmente úteis e compreensíveis em conjuntos de dados".

[9] Um *data warehouse* é uma coleção de dados orientada por assuntos, integrada, variante no tempo, e não volátil, que tem por objetivo apoiar aos processos de tomada de decisão (Inmon, 1997).

Portanto, o processo KDD tem por objetivo principal descobrir padrões ocultos nos dados com base nos recursos disponíveis para cada aplicação e mostrar os resultados obtidos aos tomadores de decisão.

Logo, é um consenso nesta área que a descoberta de conhecimento em banco de dados é um processo propriamente dito, com passos adequadamente divididos e passíveis de repetição, independente de quem os execute.

Vale ressaltar que freqüentemente o termo KDD é conhecido de maneira popular como Mineração de Dados e, na maioria das vezes, sem distinção alguma entre eles. A caracterização dos termos é bastante variada, pois alguns autores consideram a mineração de dados como um dos passos do processo KDD, apesar de que a essência da descoberta de conhecimento está na extração de dados úteis através dos algoritmos de mineração.

A seguir são apresentadas algumas visões a respeito de processos KDD e mineração de dados. A escolha dos autores deve-se ao grau de relevância considerado nas referências bibliográficas estudadas.

2.1.1. Processo KDD segundo Fayyad

Segundo Fayyad, Piatetsky-Shapiro e Smyth (1996), inicialmente, a descoberta de conhecimento contava com análises e interpretações manuais do conhecimento por meio das opiniões dos especialistas, exemplos claramente vistos na área de saúde, marketing, ciências e também em finanças.

Devido ao aumento do volume de dados, torna-se cada vez mais impraticável esse tipo de análise manual, incentivando assim trocar tarefas manuais por aplicações KDD que servem como interface entre usuários e seus dados, auxiliando o processo de descoberta. Estas aplicações estão em diversas áreas como na astronomia com análise de imagens de objetos e nos negócios como marketing, investimentos, detecção de fraudes, telecomunicações, agentes inteligentes e outros.

A natureza multidisciplinar da descoberta de conhecimento em banco de dados é consolidada através dos métodos e ferramentas fornecidas, por exemplo, pelas áreas de inteligência artificial, banco de dados, estatística, visualização gráfica, aprendizado de máquina, reconhecimento de padrões.

A obtenção de conhecimento concentra-se em etapas e possui suas condições. Assim, se o processo de busca por padrões for orientada a dados, o produto final não pode ser diferente do conhecimento desejado e a mineração de dados é apenas um passo dentro do processo inteiro de busca ou simplesmente KDD. A partir da identificação dos passos para KDD, Fayyad, Piatetsky-Shapiro e Smyth (1996, p.39) também define o passo mineração de dados como “aplicação de algoritmos específicos para extração dos padrões de dados”.

As principais caracterizações do processo KDD, por Fayyad, Piatetsky-Shapiro e Smyth (1996) são: (1) a natureza interativa e iterativa, (2) a condição para obter conhecimento possui nove passos básicos. Interativo porque se não houver mecanismos de comunicação com o usuário, principalmente no domínio da aplicação e a avaliação de padrões interessantes, a busca torna-se inválida e inadequada. Iterativo porque possibilita repetições entre quaisquer dois dos nove passos do processo KDD.

O processo KDD é complexo, iterativo e interativo. A natureza iterativa proporciona maior interação do usuário ao processo de descoberta, visto que não é suficiente apenas seguir passos, mas sim interagir ao processo de descoberta várias vezes até que o conhecimento realmente esperado, pelo especialista, seja encontrado. As iterações dos passos do processo KDD indicam mais interação do usuário e a obtenção de conhecimento mais preciso, já que os refinamentos só podem ser realizados pelos usuários e não automaticamente pelas técnicas de mineração de dados. Na Figura 2.1 pode-se perceber a forma de agir para buscar conhecimento em banco de dados, onde o envolvimento do usuário é fundamental para o sucesso do processo KDD e a seqüência de passos oferece gradualmente o conhecimento desejado.

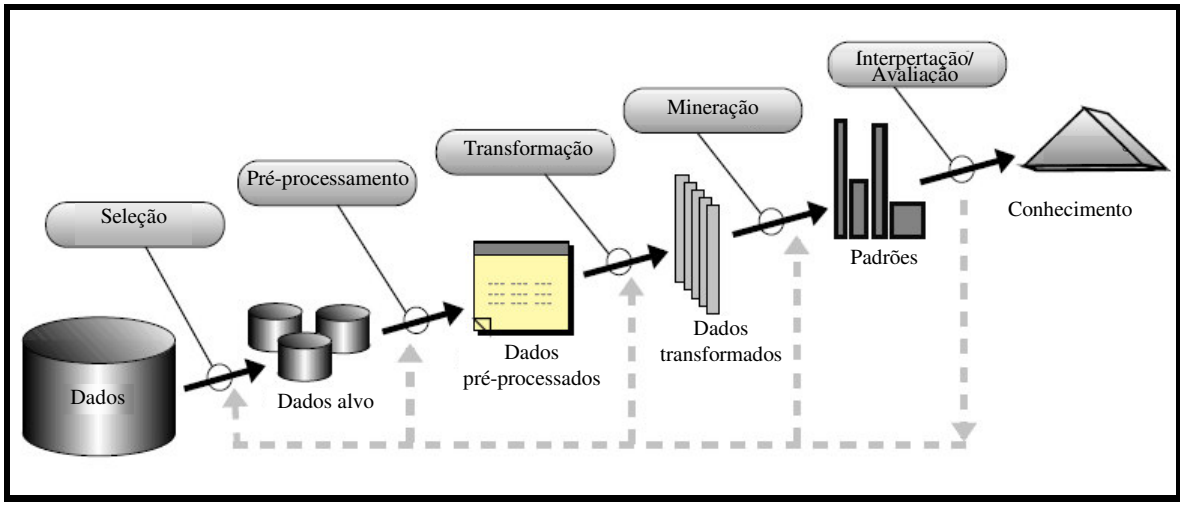


Figura 2.1. Processo KDD segundo Fayyad

A seguir são brevemente explicados os passos do processo KDD segundo Fayyad, Piatetsky-Shapiro e Smyth (1996):

1) Seleção de Dados

O objetivo é desenvolver um entendimento do domínio da aplicação e o conhecimento prévio relevante e, também, identificar as metas do processo KDD de acordo com o ponto de vista do usuário.

2) Criação de Dados Alvo

O objetivo é a criação de um conjunto de dados alvo, selecionando um conjunto de dados, ou focando um subconjunto de variáveis ou dados de exemplo, no qual a descoberta é executada.

3) Pré-Processamento

Os objetivos são limpeza e pré-processamento. Operações básicas que incluem remoção de ruído (inconsistências de dados) quando apropriado, colecionando informações necessárias para modelar ou explicar o ruído, decidindo sobre estratégias para manipulação de campos de dados e explicando informação seqüencial de tempo e de mudanças de conhecimento.

4) Dados Pré-Processados

O objetivo é a redução e projeção de dados. Encontrar características úteis para representar a dependência dos dados no objetivo da tarefa de descoberta. Com redução da dimensionalidade

ou métodos de transformação, o número eficaz de variáveis sob consideração pode ser reduzido, ou representações invariantes para os dados podem ser encontradas.

5) Transformação de Dados

O objetivo é a combinação das metas do processo KDD (passo 1) para um método particular de Mineração de Dados. Por exemplo, sumarização, classificação, regressão, clusterização.

6) Dados Transformados

O objetivo é a análise exploratória e representa modelo e seleção de hipóteses: escolhendo algoritmos de mineração de dados e selecionando métodos para serem usados pelas buscas por padrões de dados. Esse processo inclui decisões cujos modelos e parâmetros deveriam ser adequados e combinados a um método particular de mineração de dados com o critério inteiro do processo KDD.

7) Mineração de Dados

O objetivo é a busca por padrões de interesse em um formato representativo particular ou um conjunto de tais representações, incluindo regras de classificação ou árvores, regressão e agrupamento. O usuário pode facilitar significantemente o método de mineração de dados pela execução correta dos passos precedentes.

8) Padrão Minerado

O objetivo é a interpretação dos padrões minerados, possivelmente retornando para quaisquer dos passos anteriores, realizando assim iterações. Esse passo pode envolver visualização de padrão e modelos extraídos ou visualização dos dados obtidos de modelos extraídos.

9) Interpretação/Avaliação

O objetivo é a representação do conhecimento descoberto: usando o conhecimento diretamente, incorporando o conhecimento dentro de outro sistema para uma ação futura, ou simplesmente documentação dele e reportando para as partes interessadas. Este processo inclui verificação e resolução de conflitos potenciais através de conhecimento previamente conhecido ou extraído.

A visão geral de mineração de dados envolve métodos específicos para cada meta e algoritmos que implementam esses métodos, por exemplo, classificação, regressão,

agrupamento e sumarização. Em quaisquer algoritmos de mineração de dados podem ser identificados três componentes, a saber: modelo de representação, modelo de avaliação e busca.

Portanto, o processo KDD refere-se à descoberta de conhecimento útil em dados e mineração de dados refere-se a um passo particular desse processo.

2.1.2. Processo KDD segundo Han e Kamber

Para Han e Kamber (2001), a mineração de dados é um passo do processo de descoberta de conhecimento. Mineração de dados é definida como "processo de descoberta de conhecimento interessante vindo de grande volume de dados armazenados nos bancos de dados, *data warehouse* ou outro repositório de dados".

O passo de mineração de dados interage com o usuário e com a base de conhecimento e os padrões descobertos poderiam ser armazenados como conhecimento na base de conhecimento para futuras buscas, como mostrados na Figura 2.2.

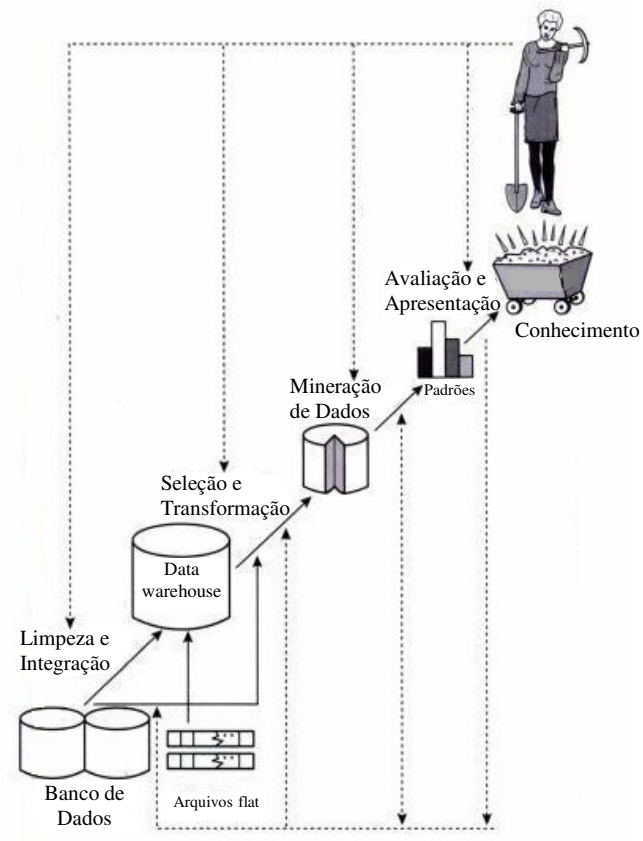


Figura 2.2 - Processo KDD segundo Han e Kamber

Os seguintes passos representam a seqüência iterativa sugerida pelos autores:

1) Limpeza de Dados

O objetivo é remover dados inconsistentes e inconsistência de padrões.

2) Integração de Dados

O objetivo é combinar múltiplas fontes de dados. É comum tratar da limpeza e integração de dados como passo de pré-processamento e é a fase em que os resultados são armazenados em *data warehouse*.

3) Seleção de Dados

O objetivo é selecionar dados relevantes para a tarefa de análise e aplicação dos métodos de mineração de dados.

4) Transformação de Dados

O objetivo é transformar ou consolidar dados em formatos apropriados para execução de mineração de dados, por meio de operações como sumarização e agregação.

5) Mineração de Dados

O objetivo é aplicar métodos inteligentes na análise e extração de padrões de dados.

6) Avaliação dos Padrões

O objetivo é identificar o verdadeiro padrão interessante do conhecimento representado baseado em algumas medidas de interesse.

7) Apresentação do Conhecimento

O objetivo é utilizar técnicas de visualização e representação do conhecimento para mostrar o atual conhecimento minerado para o usuário.

Portanto, a qualidade dos dados influencia no processo KDD inteiro e os dados minerados dependem das técnicas aplicadas nos quatro primeiros passos do processo, suportado algumas vezes pela utilização de *data warehouse*.

2.1.3 Processo KDD segundo Groth e Lans

De acordo com Dias (2001) o processo de descoberta de conhecimento segue cinco passos básicos sob a visão de Groth (1998), juntamente com mais uma fase sugerida ao processo por Lans (1997) enfatizando a definição de objetivos. Como mostra a Figura 2.3.

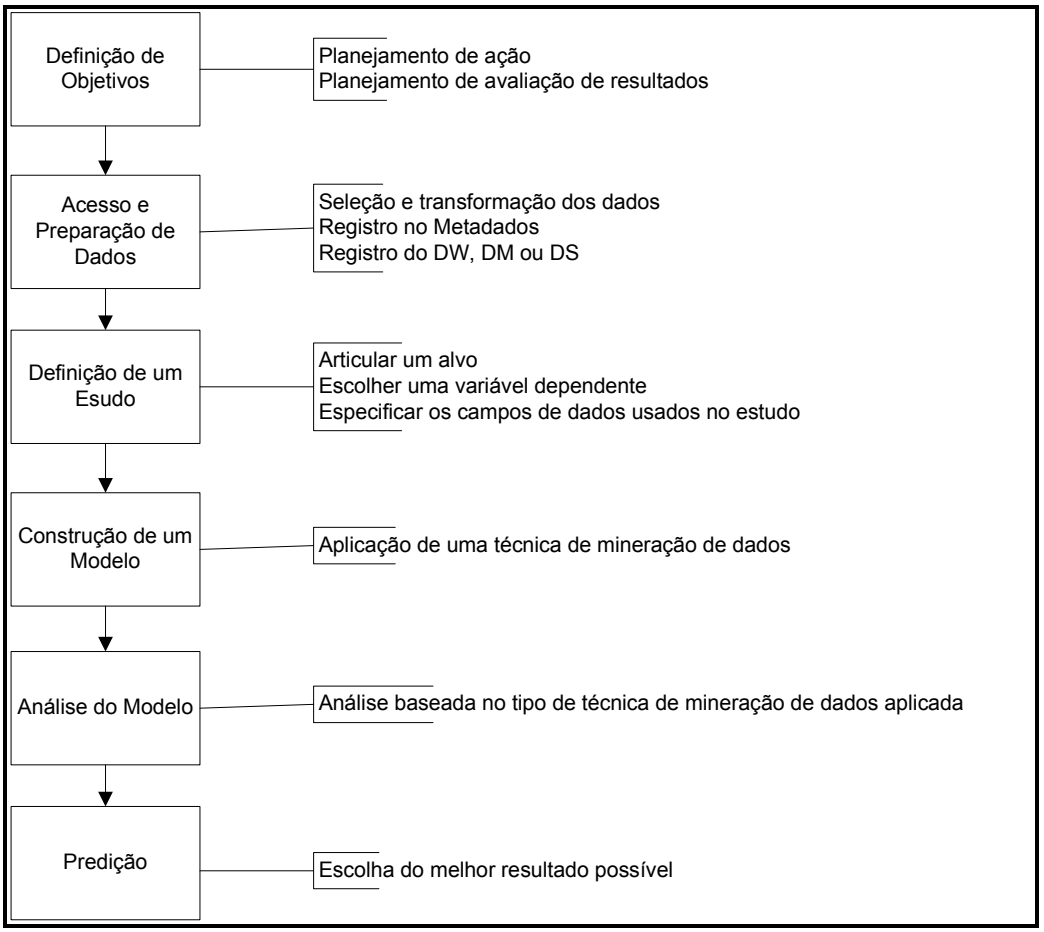


Figura 2.3 - Processo KDD segundo Groth e Lans (adaptado por Dias, 2001, p.18)

Os sistemas de descoberta de conhecimento em banco de dados possuem características de indeterminismo, então Dias (2001) propõe um modelo de formalismo desse tipo de sistema. Nesse modelo, são utilizados diagramas UML para representação das características de sistemas KDD. Esses diagramas são mapeados para a linguagem de especificação formal (E-LOTOS) para tornar possível a verificação e a validação do sistema especificado. Também é definida uma estrutura de agentes para o controle e a realização dos serviços necessários em um sistema KDD.

Após estudo inicial de vários autores da área de KDD, identificou-se a característica principal de cada definição do processo KDD, e respectivamente a divisão de passos adotada por todos os autores. Estes são sumarizados a seguir no Quadro 2.1., que está ordenado de forma crescente conforme número de passos do processo KDD adotado pelos autores. A maioria dos processos KDD estudados são iterativos e interativos, representados pelas letras (i,i), alguns consideram o processo não linear também, representado pelas letras (nl).

Autores	Característica Principal	Passos Principais
Feldens, M. A. et al. (1998) [i,i,nl]	Orientado a **Aplicação**	1.Pré-Processamento 2.Mineração de Dados 3.Pós-Processamento
Baranauskas, J. A. (2001) [i,i]	Processo **Empírico** e pode incluir Aprendizado de Máquina	1.Pré-Processamento 2.Mineração de Dados 3.Pós-Processamento
Rezende, S. O. et al. (2005) [i,i]	Inclui 2 fases: Conhecimento do **Domínio**, Utilização do **Conhecimento** obtido	1.Identificação do Problema 2.Pré-Processamento 3.Extração de Padrões 4.Pós-Processamento
Weiss, S. M.; Indurkhya, N. (1998) [i,i]	Mineração de dados **Previsivo** dirigido por meta	1.Preparação de Dados 2.Redução de Dados 3.Modelagem e Previsão de Dados 4. Análise de Caso e Solução
Brachman, R. J.; Anand,T. (1996) [i,i]	Enfatiza processo centrado no **Humano**. Como suportar análise humana	1.Limpeza dos Dados 2.Modelo de Desenvolvimento 3.Análise dos Dados 4.Geração da Saída
Mannila, H. (1997) [i,i]	KDD não é um sistema de análise **Automática**	1.Entendimento do Domínio 2.Preparação do Conjunto de Dados 3.Descoberta de Padrões 4.Pós-Processamento 5.Apresentação do Resultado p/ Usuário
Klemettinen, M.; Mannila, H.; Toivonen, H. (1997) [i,i]	Enfatiza a **Descoberta** de padrões e flexibilidade na **Visualização**	1.Pré-Processamento 2.Transformação 3.Descoberta 4.Representação 5.Utilização
Groth, R. (1998) [i,i]	Duas **formas de aplicar** DM: Verificação ou Descoberta	1.Preparação de Dados 2.Definição de um Estudo 3.Construção de um Modelo 4.Entendimento do modelo 5. Predição
Saitta, L.; Neri, F. (1998)	Envolver o usuário **em todo** o processo (AM)	1.Compreensão do Domínio 2.Aquisição de Conhecimento 3.Pré-Processamento dos Dados 4.Interpretação e Avaliação 5.Apresentação do Conhecimento

Autores	Característica Principal	Passos Principais
Mitra, S.; Pal, S.K.; Mitra P. (2002) [i,i,nl]	Dados não são obtidos de forma Direta (abordagem **híbrida**)	1.Seleção de Dados 2.Pré-Processamento 3.Transformação 4.Mineração de Dados 5.Interpretação e Avaliação
Chapman, P. et al. (CRISP-DM 1.0) (2000) [i,i]	Divisão do processo em **Fases**, **Tarefas** (genéricas e especializadas) e Instanciação.	1.Entendimento do Negócio 2.Entendimento dos Dados 3.Preparação dos Dados 4.Modelagem 5.Avaliação 6.Implantação
Reinartz, T. (1999) [i,i]	Defini 4 classes de **Pessoas** e define **Pré-Requisitos** para entendimento do processo KDD	1.Compreensão do Domínio 2.Compreensão dos Dados 3.Preparação dos Dados 4.Exploração dos Dados 5.Mineração dos Dados 6.Avaliação 7.Apresentação
Han, J.; Kamber, M. (2001) [i,i]	Padrões interessantes são armazenados em **base de conhecimento**	1.Limpeza de Dados 2.Integração de Dados 3.Seleção de Dados 4.Transformação de Dados 5.Mineração de Dados 6.Avaliação do Padrão 7.Apresentação do Conhecimento
Fayyad, U. ; Piatetsky-Shapiro,G.; Smyth, P. [i,i,nl] (1996)	Define **Atores**: Analista de dados, Especialista do Domínio e Usuário	1.Seleção 2.Dados Selecionados 3.Pré-Processamento 4.Dados Processados 5.Transformação 6.Dados Transformados 7.Mineração de Dados 8.Padrão Minerado 9.Integração/Avaliação

Autores	Característica Principal	Passos Principais
Goebel, M.; Gruenwald, Le, (1999) [i,i,nl]	Propõe **Esquema** de classificação de características em softwares	1.Entendimento do Domínio 2.Aquisição do data set 3.Integração dos data set 4. Limpeza, Pré-Processamento e Transformação 5.Desenvolvimento e Construção 6.Escolha de Algoritmos de Mineração 7. Visualização 8.Teste e Verificação 9. Manutenção de Descoberta de Conhecimento

Quadro 2.1 – Características e passos principais de processos KDD

As publicações mais antigas são de 1996, aliás, um ano relevante para área de KDD, e as mais novas são de 2005 em congresso brasileiro de computação e livros de mineração de dados.

2.2. PROCESSOS DE ENGENHARIA DE SOFTWARE

Atualmente, o desenvolvimento de sistemas recorre às técnicas e aos métodos explicados pela engenharia de software. Entretanto, antigamente a criação e entrega de soluções automatizadas aos usuários limitava-se a linguagens de programação, em um paradigma que abrangia a atividade de informatizar procedimentos manuais das empresas.

Juntamente à evolução de hardware e software, a engenharia de software[10] surgiu para se dedicar a problemas práticos da produção de sistemas. Seu principal intuito é oferecer um controle para os aspectos de produção e manutenção de software, objetivando qualidade de produto e aproximação com os usuários proprietários, mas também o aspecto de gerenciamento da equipe de trabalho.

Neste contexto, a evolução dos processos de desenvolvimento de software, desde a década de 70 até os dias de hoje, ocorreu paralelamente à evolução dos computadores, linguagens de programação e banco de dados. Existiram diversos modelos que representavam

[10] Engenharia de Software: é uma disciplina da engenharia que se ocupa de todos os aspectos da produção de software (Sommerville, 2003).

seqüências de atividades, tendo como resultado final um produto de software. Um dos modelos mais conhecidos é o Modelo Cascata que sugere uma abordagem seqüencial para o desenvolvimento de software. Este oferece a divisão de fases conforme o conjunto de atividades a serem realizadas, por exemplo, atividades de análise são diferentes daquelas de codificação. Outro modelo importante é o Modelo Espiral que agregou ao processo repetições do ciclo de vida, chamadas de iterações e também considerou a abordagem de prototipação, no intuito de tornar o projeto mais realístico que os projetos seguidos anteriormente pelo modelo cascata.

Sendo assim, os modelos evolucionários são adequados ao desenvolvimento orientado a objetos, sendo o UP um dos modelos mais referenciados atualmente, o qual se preocupa com o produto de software e com o processo de software de maneira igualitária.

2.2.1. Processo Unificado de desenvolvimento de sistemas

Historicamente, o UP marcou época através de sua proposta de trabalho que abrange determinados aspectos na certeza do alcance de metas, ou seja, software com qualidade e entregue dentro de prazos e orçamentos estimados. Para alcançar a meta de produzir um software com qualidade, o processo unificado gerencia fatores que influenciam no desenvolvimento, por exemplo, questionamento sobre quem, o que, quando e como o produto de software pode ser obtido.

Desde 1987, os pesquisadores tiveram a intenção de propor através desse modelo de desenvolvimento de software, várias representações do sistema em um roteiro de criação que considerasse as deficiências não atendidas por modelos propostos anteriormente ao UP.

O uso extensivo de notação UML, juntamente com abordagem iterativa para cada ciclo do processo, torna o UP um processo atual que faz uso das melhores práticas do mercado, como desenvolvimento baseado em componentes, reuso, modelagem dirigida à arquitetura e incrementos de software como resultado das iterações previamente planejadas.

Segundo Scott (2003) e Jacobson, Booch e Rumbaugh (1999), o UP possui três aspectos-chave, quatro fases, cinco *workflows* e três elementos-chave que são:

- ***Aspectos-Chave:***

  - Dirigido por Caso de Uso significa que os casos de uso dirigem todo o trabalho de desenvolvimento, permitindo a rastreabilidade de requisitos funcionais;
  - Centrado na Arquitetura significa que a arquitetura, sendo uma organização fundamental do sistema, serve de visão geral comum dos participantes e para facilitar futuros refinamentos;
  - Iterativo e Incremental significa a presença da natureza iterativa (pequenos projetos). Incrementos são melhoras nas versões de cada miniprojeto entregue.

- ***Fases:***

Fase é simplesmente o tempo decorrido entre dois marcos principais. Cada fase possui uma ou mais iterações. Cada ciclo termina a liberação com de uma versão para o cliente. Na Figura 2.4 são apresentados os *milestones*[11] do processo unificado. As fases do UP são:

- Concepção tem por objetivo elaborar a Arquitetura candidata;
- Elaboração tem por objetivo elaborar a Base Arquitetônica;
- Construção tem por objetivo elaborar a Capacidade Operacional Inicial;
- Transição tem por objetivo disponibilizar a Liberação do Produto.

<table>
<tr><th>Concepção</th><th>Elaboração</th><th>Construção</th><th>Transição</th></tr>
<tr><td rowspan="3">Iteração 1</td><td>Iteração 2</td><td>Iteração x+1</td><td>Iteração y+1</td></tr>
<tr><td>...</td><td>...</td><td>...</td></tr>
<tr><td>Iteração x</td><td>Iteração y</td><td>Iteração z</td></tr>
<tr><td>Objetivos do Ciclo de Vida</td><td>Arquitetura Base</td><td>Capacidade Operacional Inicial</td><td>Liberação do Produto</td></tr>
</table>

Figura 2.4 – *Milestones* do processo unificado

[11] *Milestones* no contexto do UP e deste trabalho significa *Marcos de Progresso* no processo de engenharia de software.

- ***Workflows:***

Workflow é um conjunto de atividades que os membros do projeto executam. E os cinco *workflows* relacionados a seguir, atravessam o conjunto das quatro fases:

  - Requisitos: visa construir o Modelo de Casos de Uso;
  - Análise: visa construir o Modelo de Análise para refinar e estruturar os requisitos funcionais;
  - Projeto: visa construir o Modelo de Projeto, que são as realizações físicas do caso de uso e, também, o Modelo de Instalação;
  - Implementação: visa construir o Modelo de Implementação, realizando o empacotamento em componentes de software;
  - Teste: visa construir o Modelo de Testes, oferecendo mais integração de sistemas e mais teste de unidade.

- ***Elementos-Chave:***

  - Artefatos são qualquer porção significativa de informação interna ou a ser fornecida a interessados externos que desempenhem um papel no desenvolvimento do sistema;
  - Papéis são o que um indivíduo pode desempenhar no projeto;
  - Atividades são tarefas que cada papel executa a fim de produzir artefatos.

2.3 MODELAGEM DE PROCESSO DE SOFTWARE

A característica de qualidade dos softwares, associada aos requisitos funcionais e implícitos, estão em conformidade com a satisfação dos usuários que os encomendam e, também, aos critérios sistemáticos escolhidos para o desenvolvimento do software. Segundo Fuggetta (2000), a qualidade do software desenvolvido possui correlação direta com a

qualidade do processo de software que o serviu. Portanto, o autor sugere que a pesquisa investigue as melhorias dos processos, quantificadas por meio da integração de novas técnicas às atividades tradicionais do desenvolvimento.

De acordo com Acuña e Ferré (2001), o processo de software é um conjunto ordenado de atividades para o gerenciamento, desenvolvimento e manutenção de software, ou ainda representa o conjunto de atividades de produção de um sistema solicitado por um grupo de pessoas, conforme condições organizacionais. Já o modelo de processo de software é uma representação abstrata de um processo de software.

Em geral, os modelos de processos descrevem a arquitetura, o projeto e a definição do processo, podendo ser analisados, validados e simulados, caso forem descritos com formalismos executáveis (Acuña e Ferré, 2001).

A modelagem de processo de software descreve a criação de modelos do processo de desenvolvimento de software, referindo-se à definição de processos como modelos (Acuña e Ferré, 2001).

Neste contexto, a modelagem de processos de software surge então como um formalismo para tratar a complexidade natural do desenvolvimento de software, oferecendo essencialmente a representação das características dos mesmos, de maneira precisa e compreensiva para os engenheiros de software.

Existe um consenso quanto às metas e aos benefícios da modelagem de processos. De acordo com Kellner e Hansen (1988) e Curtis, Kellner e Over (1992), o desenvolvimento de modelos de processos de software tem quatro objetivos principais, a saber: facilitar a comunicação e o entendimento do processo permitindo treinamentos, controlar e apoiar o gerenciamento de projetos de processos específicos, fornecer orientações automatizadas para o desempenho e execução do processo e suportar melhorias do processo.

Nas décadas de 1960 e 1970, o foco das atividades dos engenheiros de software estava relacionado principalmente com a definição de princípios, métodos de projeto e linguagens de programação estruturadas. Nesta época, a preocupação quanto ao processo de software limitava-se à definição dos ciclos de vida de software como o Modelo Cascata, Desenvolvimento Incremental e Desenvolvimento baseado em Prototipação (Fuggetta, 2000).

A partir da reflexão sobre as mudanças ocorridas durante quinze anos de inovações nos modelos de ciclos de vida de software, percebe-se a crescente necessidade de aproximação dos usuários às tarefas da criação e entrega dos softwares. Para que a elaboração do software fosse mais proveitosa sob o ponto de vista dos usuários, aceitou-se o acompanhamento de pequenas partes funcionais dos mesmos, chamadas de incrementos, que

traz consigo a vantagem de discorrer sobre a complexidade dos sistemas mais vezes e com melhor entendimento. Para que a entrega dos softwares fosse também mais eficiente, a prototipação e a análise prévia e iterativa de riscos emergem como abordagens alternativas à condução das tarefas de maior relevância, tanto para o usuário quanto para o desenvolvedor. Contrariamente aos estágios sistemáticos que o Modelo Cascata administra.

Então, os esforços de pesquisa centrados em processos de software acontecem depois de 1980, quando o suporte tecnológico ao desenvolvimento juntamente à definição de fatores influenciáveis, como o controle do projeto ou o comportamento organizacional, representam um impacto significativo nos prazos e orçamentos dos sistemas. A investigação do desenvolvimento de software como um processo auxilia na visualização de diferentes dimensões das tarefas de especificação e na entrega de resoluções de problemas, a fim de estabilizar a área de engenharia de software.

Acuña e Ferré (2001) reforça a idéia de que um modelo de processo requer uma linguagem de modelagem para sua descrição. Existem vários tipos de modelagem de processo, alguns orientados a atividades, outros a pessoas. Porém, todos modelam elementos essenciais de qualquer processo.

2.3.1 Elementos Básicos da Modelagem de Processo de Software

Apesar de existirem diversas abordagens de modelagem de processo, os elementos comuns dentre estas compõem a estrutura de um modelo de processo de software. Os diferentes elementos de um processo são: Agente ou Ator, Papel, Atividade, Artefato ou Produto e Ferramenta.

Agente ou Ator significa as entidades que executam um processo, ou assumem um papel durante a execução de uma tarefa. Podem ser classificados em entidades humanas ou entidades automatizadas como hardware e software. Um ator, uma pessoa ou um sistema pode passar por vários papéis, que são compostos de conjuntos consistentes de atividades.

Papel significa um grupo de responsabilidades ou descreve um conjunto de agentes, que executam uma atividade específica do processo de software.

Atividade representa o elemento principal do processo, pela sua capacidade de diferenciação entre modelos de processo. Pode ser um estágio de um processo que produz mudanças visíveis externamente no estado do produto de software. Uma atividade pode ter

uma entrada, uma saída ou algumas vezes resultados intermediários, genericamente chamados de produtos. Estas estão associadas aos papéis, a outras atividades e a artefatos.

Artefato ou Produto significa uma porção representativa de informação do produto. A natureza dos artefatos permite que estes sejam criados, acessados ou modificados pelas atividades do processo. Diferentes versões de artefatos podem ser criadas conforme a necessidade.

Ferramenta significa apoio automatizado à execução de tarefas (ex. ferramentas CASE, compiladores).

Os elementos básicos se comunicam entre si, conforme ilustrado na Figura 2.5, compondo o relacionamento exigido quando se opta por um processo sistemático para o desenvolvimento de sistemas. Este relacionamento serve de mecanismo para visualização, com mais clareza, da grande quantidade de informações pertencentes aos processos, sendo fundamental para a tarefa de modelagem de processos.

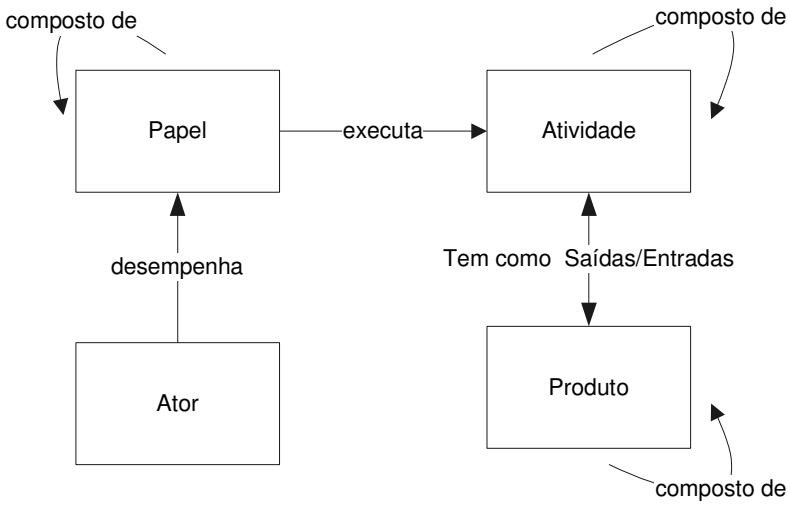


Figura 2.5 – Elementos básicos da modelagem de processos de software

2.3.2 Abordagens de Modelagem de Processo de Software

Há vários propósitos que induzem o uso da modelagem de processos, desde a melhoria do entendimento de um processo, passando por treinamentos e aprendizados, até o mais comum deles que é o projeto de um novo processo, descrito pela sua estrutura e organização. Sabe-se também que o recurso de simulação e otimização dos processos é útil para as organizações.

Devido à necessidade de representação variada de informações dos processos de software, foram construídos diferentes tipos de modelagem. Estes por sua vez, consideram como necessidade a integração de múltiplos paradigmas de representação, a fim de oferecer nível adequado para especificação dos processos aos engenheiros de software. A seguir são descritos alguns destes paradigmas de representação.

O formalismo, freqüentemente abordado pelas Linguagens de Modelagem de Processo (*Process Modeling Languagens* - PMLs), exige aprendizado no uso de suas técnicas e métodos, permitindo a representação completa das atividades, papéis, artefatos e ferramentas usados no processo em questão. Em geral, as PMLs são baseadas em paradigmas lingüísticos como redes de Petri[12] e apoiadas por ambientes de criação e exploração dos modelos de processos de software chamados de Ambiente de Engenharia de Software Centrado a Processo (*Process-Centered Software Engineering Environment* - PSEE). Por outro lado, as PMLs não têm grande aceitação na comunidade de engenharia de software, pois são complexas, extremamente sofisticadas e fortemente orientadas à modelagem detalhista do processo de software. Como elas são limitadas pelos PSEEs, que as suportam, geralmente quem modela processos percorre pelos estágios de maneira inflexível e rígida, porque não é tolerado pelos ambientes especificações incompletas, informais ou parciais, o que impossibilita a construção incremental da modelagem do processo de software.

Segundo Cereja Junior, Sant'Anna e Borrego Filho (2002), PSEEs são soluções tecnológicas que apóiam a melhoria dos processos de software, especialmente aqueles que precisam ser modelados e formalizados usando as PMLs ou modelados usando uma linguagem informal como a UML (*Unified Modeling Language*).

Jäger, Schleicher e Westfechtel (1999) afirmam que o uso da UML para modelagem do processo de software é benéfico do ponto de vista do engenheiro de software, pois suporta facilmente as fases da modelagem de processo, que são análise e projeto, e constrói representações que dependem tanto dos diagramas estruturais e comportamentais como dos mecanismos de extensão da UML.

Hauck e Wangenheim (2004) identificam como necessidade de melhoria da qualidade do software a avaliação do processo de software nas pequenas empresas. Para que esta tarefa possa acontecer, faz-se necessária a aplicação de métodos de modelagem de processo, como intenção de viabilizar a constante revisão e adaptação do processo à realidade limitada e

[12] (1) Uma máquina de estado finito estendida que permite a habilitação concorrente de transições e comunicação assíncrona. (2) Uma forma de máquina de estado finito utilizada para descrever e analisar a estrutura e o fluxo de informações em sistemas (Peters e Pedrycz, 2001).

vulnerável das micro e pequenas empresas no Brasil. Como na filosofia empresarial muitas vezes não é valorizado um processo sistemático para as tarefas do desenvolvimento de software, a abordagem escolhida pelos autores para um estudo de caso foi menos rígida quanto aos investimentos relacionados à sua aplicação. A metodologia é baseada em cenários e primeiramente gera-se o modelo descritivo dos processos executados atualmente na empresa, documentando-o e disseminando-o, com a intenção de avaliação e revisão do processo, possibilitando a melhoria contínua na organização.

A notação para modelar processos de desenvolvimento de software, proposta por Lai no início da década de 1990 (Pfleeger, 2004), pretende representar qualquer processo em qualquer nível de detalhe. Os modelos propostos são: o estático, que trata dos aspectos da transformação das entradas em saídas e o dinâmico, que trata da visualização de como os produtos intermediários são transformados. Na visão do autor, os elementos de qualquer processo são: as atividades e suas próprias seqüências, o modelo de processo, os recursos necessários, os controles de aprovação manuais ou não, a política e as restrições organizacionais e, finalmente, a hierarquia dos agentes do processo; todos igualmente importantes em um projeto de modelagem de processo. Todas estas estratégias servem para ordenar a grande quantidade de informação inerente à modelagem de um processo.

O SPEM (*Software Process Engineering Metamodel Specification*) apresenta uma proposta de unificação das diferentes metodologias existentes de modelagem de processo de software e denota a definição completa dos modelos de processo OMG (2005). Na Seção 2.4 estão resumidos os princípios deste metamodelo que foram escolhidos para este trabalho de mestrado.

Vale ressaltar que da mesma forma ocorrida na modelagem de sistemas, em que atividades abrangem aspectos estruturais e comportamentais por meio de diagramas, como o de casos de uso e de classes, a modelagem de processos visa representar, também por meio de diagramas, aspectos funcionais, estruturais e comportamentais do processo (Kellner e Hansen, 1988).

Segundo Jacobson, Booch e Rumbaugh (1999), um processo define "*quem*", está fazendo "*o que*", "*quando*" e "*como*", para atingir determinada meta que é a construção do software. Então os aspectos funcionais dos processos de software são geralmente representados por diagramas de atividade mostrando "*o que é feito*". Os aspectos estruturais e organizacionais podem ser representados por diagramas de pacotes mostrando "*quem faz e onde é feito*". Por fim, os aspectos comportamentais, que freqüentemente são representados por diagramas de estado, mostram "*quando e como é feito*" o desenvolvimento.

Para Martins e Silva (2004), a implementação de processos de software não é uma tarefa fácil, devido à complexidade das tarefas e à instabilidade dos ambientes. O objetivo dos autores é investigar a expressividade e a adequação do metamodelo SPEM à especificação de processos. Por este motivo, esses autores comparam o metamodelo aos processos de desenvolvimento como o RUP (*Rational Unified Process*), o XP (*Extreme Programming*) e o MSF (*Microsoft Solutions Framework*). Os modelos analisados apresentam correspondência no metamodelo, o que confere a este a completude na especificação de diversos tipos de processo.

A escolha de uma das abordagens de modelagem de processo depende do estilo de trabalho e algumas vezes dos objetivos de representação, variando entre textuais ou gráficas, sendo relevantes para alguns casos ambos os aspectos.

2.4 SOFTWARE PROCESS ENGINEERING METAMODEL SPECIFICATION - SPEM

Para Martins e Silva (2004), o SPEM é um metamodelo que serve de base para a modelagem de qualquer processo. Para Genvigir (2004) o SPEM é também uma proposta de unificação entre as diferentes metodologias para a modelagem de processo.

Com a finalidade de representar a tradução do metamodelo, optou-se por permanecer a diferenciação entre letras maiúsculas e minúsculas no texto. Para representar os elementos do modelo do processo usa-se as iniciais do nome do elemento em maiúsculas.

2.4.1 Visão Geral

O grupo OMG (2005) define o SPEM como o metamodelo usado para descrever concretamente um processo de desenvolvimento de software. Um processo de software realizado em um dado projeto está fora do escopo do metamodelo. A Figura 2.6 identifica a arquitetura da modelagem do SPEM, onde M3 representa a especificação dos metamodelos, M2 representa os *templates* para os processos de nível M1 como o metamodelo SPEM, M1

representa a definição do processo associado ao processo em execução como o UPKDD e M0 representa o processo de produção no mundo real ou processo em execução.

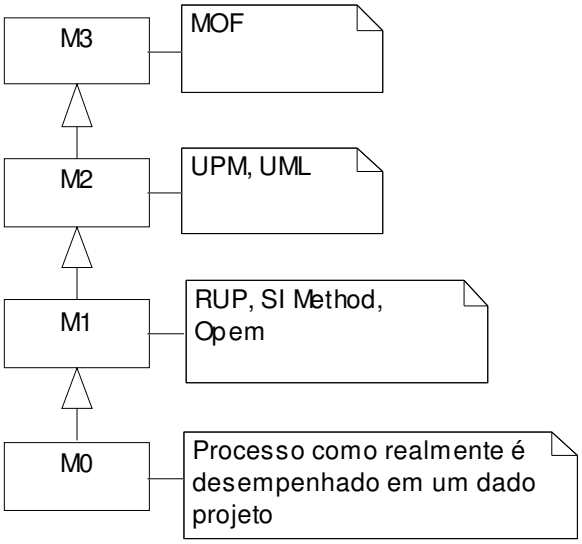


Figura 2.6 – Arquitetura em camadas do metamodelo SPEM
Fonte: OMG (2005)

O princípio do SPEM considera que um processo de desenvolvimento de software é uma colaboração entre entidades ativas e abstratas chamadas de Papéis do Processo que executam operações chamadas Atividades em entidades tangíveis e concretas chamadas de Produtos de Trabalho. A Figura 2.7 mostra a interação entre estes elementos do metamodelo.

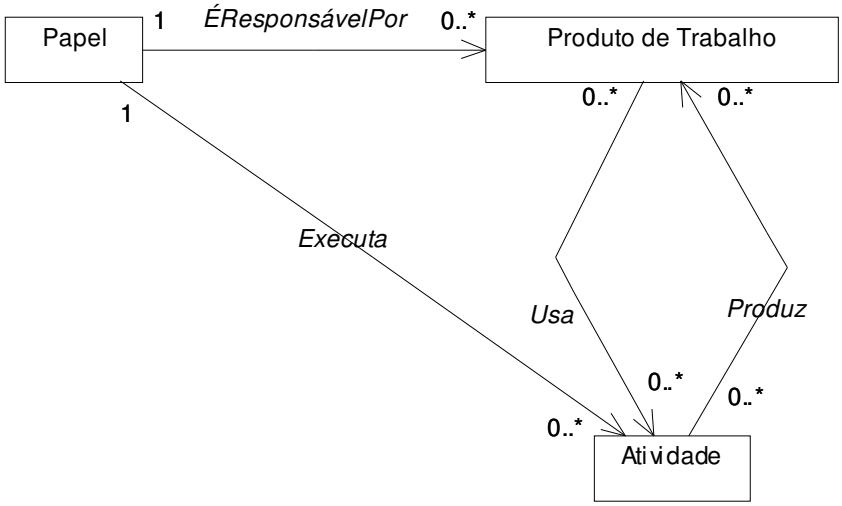


Figura 2.7 - Modelo conceitual do SPEM
Fonte: OMG (2005)

Na Figura 2.8 é apresentada a visão geral do SPEM, identificando os principais elementos do metamodelo que fundamentam a modelagem de qualquer processo de software. Nesta figura foram marcados com sombreamento cinza os elementos do modelo SPEM utilizados para a modelagem do processo UPKDD, para este trabalho de mestrado. Nota-se por este diagrama de classes que um Processo a ser modelado pode ser divido em Fases, estas por sua vez em Definições de Trabalho que têm associadas a elas Papéis no Processo que executam alguma Atividade a fim de criar ou modificar um Produto de Trabalho, sendo que este último pode participar de algum tipo de Produto de Trabalho como documentos textos, códigos-fonte, entre outros.

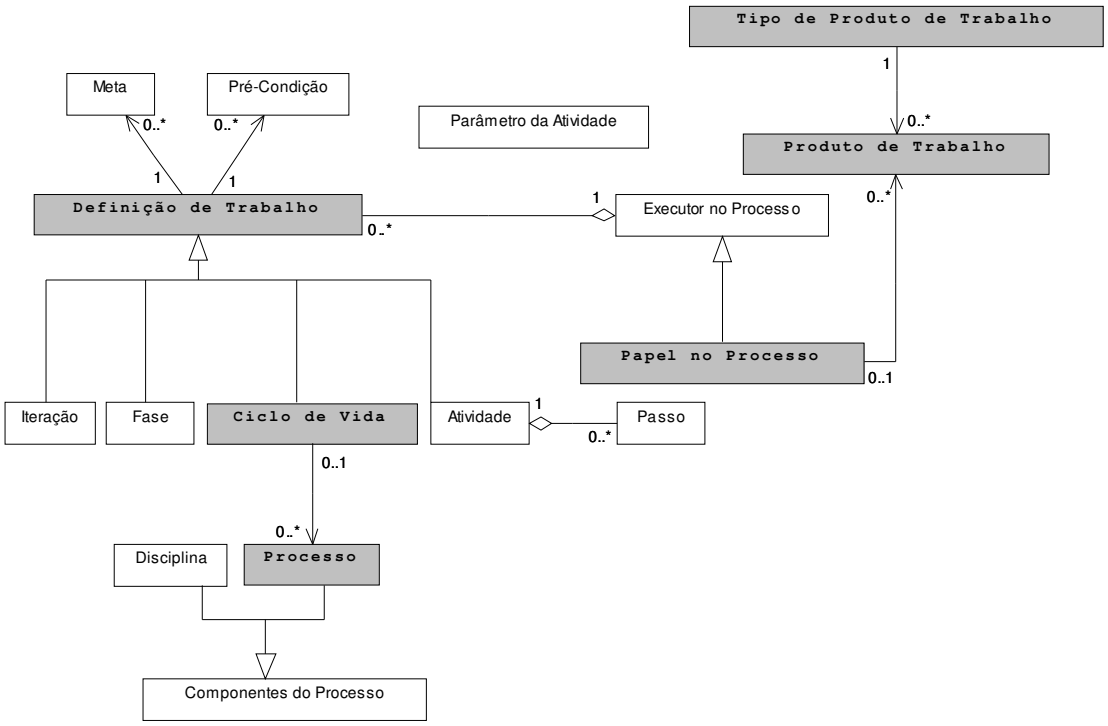


Figura 2.8 – Visão dos pacotes do SPEM
Fonte: OMG (2005)

A notação do SPEM é baseada na UML e por este motivo oferece para a modelagem de processo os mesmos diagramas que são usados para modelar sistemas. A sintaxe e a semântica dos diagramas são também as mesmas definidas pela UML.

Segundo o OMG (2005) os diagramas que podem ser usados para representar diferentes perspectivas dos modelos de processo de software são:

(1) diagrama de classes para representar aspectos de relacionamento entre Executores do Processo ou Papéis no Processo e Produtos de Trabalho,

herança, dependência, associações simples, comentários para apontar as Orientações, estrutura, decomposição e dependência dos Produtos de Trabalho;

(2) diagrama de pacotes para representar Processos, Componentes do Processo, Pacotes do Processo e Disciplinas;

(3) diagrama de atividade para representar a seqüência de atividades com suas entradas e saídas de artefatos;

(4) diagrama de casos de uso para representar o relacionamento entre papéis do processo e as principais Definições de Trabalho;

(5) diagrama de seqüência para representar interações padrões entre o metamodelo SPEM e instâncias de elementos;

(6) diagrama de estado para representar o comportamento de um elemento do modelo SPEM.

Segundo o OMG (2005) os diagramas de implementação e componentes não estão inclusos no metamodelo SPEM, por causa de algumas características semânticas da UML.

A explicação dos estereótipos do SPEM juntamente com a explicação do ambiente para a modelagem do UPKDD estão detalhadas Capítulo 3 e Apêndice B.

2.4.2 Estrutura do Processo

Um dos pacotes que explicam o metamodelo é mostrado na Figura 2.9, que define os principais elementos estruturais cuja descrição do processo é construída. Um Produto de Trabalho ou Artefato é qualquer coisa produzida, consumida ou modificada por um processo. Um Tipo de Produto de Trabalho descreve uma categoria de um artefato. Definição de Trabalho é um tipo de operação que descreve o trabalho executado em um processo. Atividade é a principal subclasse da Definição de Trabalho e descreve um pedaço de trabalho executado por um Papel no Processo. Uma Atividade pode consistir de elementos atômicos chamados Passos. Executor no Processo define um executor para um conjunto de Definições de Trabalho em um processo e tem uma subclasse chamada Papel no Processo. Papel no Processo define responsabilidades sobre Produtos de Trabalho específicos e define os papéis que executam <<perform>> ou auxiliam <<assist>>/<<assistant>> em atividades específicas.

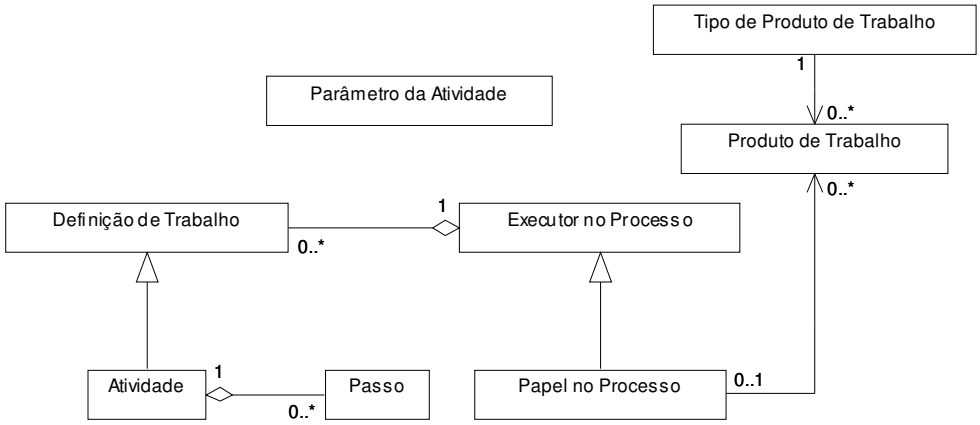


Figura 2.9 – Pacote da estrutura do processo, segundo SPEM
Fonte: OMG (2005)

2.4.3 Componentes do Processo

Outro pacote explicativo do SPEM é mostrado na Figura 2.10, sendo que as classes deste pacote são focadas em partes internas descritivas. Componentes do Processo é uma parte da descrição do processo que é consistente internamente e pode ser reusada por outros Componentes do Processo para montar um processo completo. Processo é um Componente do Processo que tem por intenção ser único, processo a processo. Disciplina é uma especialização particular do pacote que divide as Atividades dentro de um processo de acordo com temas comuns.

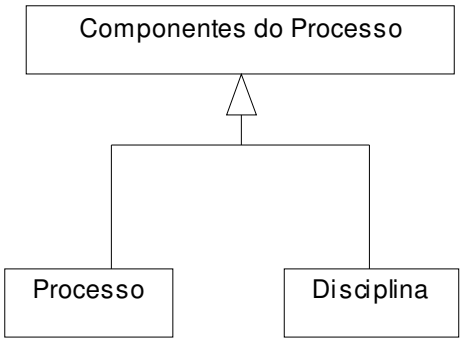


Figura 2.10 – Pacote dos componentes do processo, segundo SPEM
Fonte: OMG (2005)

2.4.4 Ciclo de Vida do Processo

O último pacote que explica o SPEM é mostrado na Figura 2.11 que identifica os elementos da definição do processo auxiliando a definir como o processo será executado. Um processo pode ser visto como uma colaboração entre papéis para alcançar determinada meta ou um objetivo. Para guiar esta execução pode considerar as restrições para ordem que as atividades devem ser executadas, chamadas de pré-condições. Fase é uma especialização da Definição de Trabalho tal que sua pré-condição define os critérios de entrada da fase, os critérios de saída da fase são suas metas (freqüentemente chamadas e *milestones*). Ciclo de Vida é definido como uma seqüência de Fases que buscam uma meta específica. Iteração é uma Definição de Trabalho com menores *milestones*.

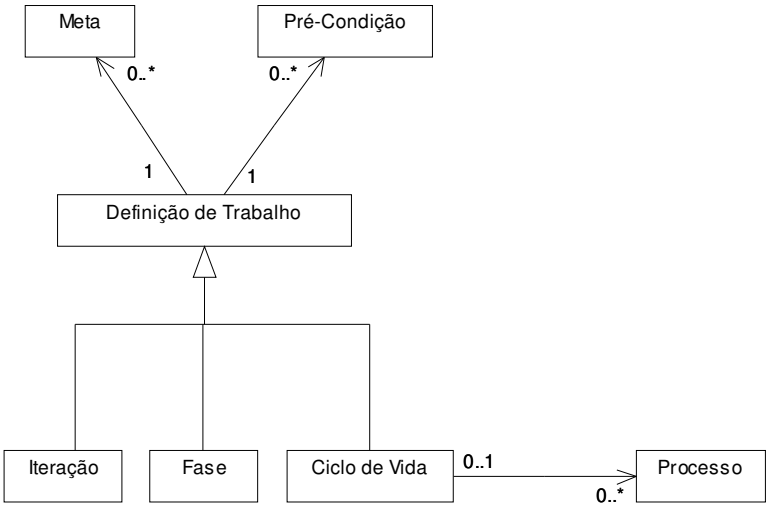


Figura 2.11 – Pacote do ciclo de vida do processo, segundo SPEM
Fonte: OMG (2005)

2.5 EXPERIMENTAÇÃO EM ENGENHARIA DE SOFTWARE

A engenharia de software é uma área que enfatiza aspectos do desenvolvimento e manutenção de sistemas, por meio de suas técnicas e métodos os quais controlam a produtividade da equipe de trabalho e a qualidade do produto final. Por ser uma área que permite a aplicabilidade de teorias, estas devem ser avaliadas de forma científica, beneficiando-se do rigor oferecido pelos modelos matemáticos, para que as mesmas possam

participar de eventos de replicação tanto na avaliação por meio de métodos diferentes, quanto na avaliação em novos projetos que abordam domínios variados.

Para que a avaliação de determinada técnica alternativa exista, é necessário valorar termos comuns da área de experimentação em engenharia de software, como por exemplo, quais tratamentos são avaliados, quais variáveis são dependentes e quais são independentes, ou ainda, qual hipótese nula ou alternativa sob a população pode ser elaborada. Esta terminologia associada aos experimentos é explorada com mais detalhes em Kitchenham, Pickard e Pfleeger (1995, p.54) e Pfleeger (1994). Na Seção 4.1.3. são sumarizados e instanciados os principais termos do estudo de caso deste trabalho de mestrado.

Selecionar o método de avaliação adequado também é fundamental, por isso Kitchenham, (1996b, 1996c) propõe critérios, como a natureza ou custo da avaliação e também restrições que influenciam na escolha de um método específico para uma circunstância particular. Neste contexto, esta dissertação segue diretrizes do método *estudo de caso*.

Um experimento depende de um método de avaliação para sua execução e, também, de um projeto de pesquisa elaborado de maneira adequada, juntamente com a opinião e observação das pessoas envolvidas. Dentre as diretrizes para realização de um estudo de caso, as conclusões válidas possuem papel essencial para que o experimento torne-se efetivo. Por meio de alguns critérios de validação é possível assegurar a qualidade desse projeto de pesquisa, como explica Kitchenham, Pickard e Pfleeger (1995, p.55) e Kitchenham e Pickard (1998, p.25).

Para Basili, Selby e Hutchens (1985), experimento é um processo de investigação que fornece entendimento necessário, de maneira científica, de produtos e processos de desenvolvimento de software. A experimentação em engenharia de software caracteriza-se por verificar teorias candidatas, as quais possuem ausência de controle ou falta de predição de suas particularidades. Esta verificação é sugerida, pelo mesmo autor, como um *framework* do processo experimental, contendo procedimentos como elaboração e teste de hipóteses sobre os modelos do produto ou processo de software. Este processo de experimentação também leva em consideração diferentes abordagens para conduzir a investigação, de acordo com cada classe de experimento e cada tipo de projeto.

A complexidade inerente à tarefa de desenvolver software não se relaciona apenas ao entendimento do domínio da aplicação, mas também, ao conhecimento sobre como desenvolver o software e, se possível, buscar este conhecimento de maneira científica. No entanto, a busca científica é dificultada devido ao fato do software ser elaborado e não

simplesmente produzido e, ainda, seguir um processo de desenvolvimento e não um processo de manufatura (Basili, Rombach e Selby, 1993).

Kitcheham, Pickard e Pfleeger (1997) afirmam que a experimentação em engenharia de software auxilia na melhoria do processo de software, por meio da investigação de novos métodos e ferramentas propostos, aumentando a produtividade e a qualidade dos softwares construídos. Neste contexto, os estudos de caso necessitam ser adequadamente elaborados, fornecendo análises importantes dos resultados, assim como assegurando que o entendimento sobre o modelo de processo ou produto estudado servirá para controle das melhorias e, também, do impacto de novas tecnologias na atual estrutura da organização.

Algumas questões da engenharia de software, como avaliação de novas técnicas, métodos ou ferramentas, poderiam ser respondidas e aprendidas por meio de simples observação da experiência de outros. Porém, esta maneira não pode ser considerada objetiva nem científica. Portanto, a sugestão de Pfleeger (1994) é atentar-se para a escolha adequada da técnica de experimentação, além das variáveis representativas, considerando fatores que influenciam como o custo da replicação do experimento, entre outros.

Kitchenham, Linkman e Law (1997) explicam a metodologia DESMET, que tem por meta a avaliação de métodos e ferramentas de engenharia de software. As autoras enfatizam a taxonomia dos métodos de avaliação, indicando sete critérios para seleção do método adequado conforme circunstância e, também, não deixam de levar em consideração alguns problemas que dificultam a execução de um experimento.

A comparação, por observação, entre metodologias ou ferramentas que se acreditam ser convenientes, representa um estudo experimental ou empírico. Este estudo depende de um método científico e auxilia principalmente no entendimento do trabalho atual da organização, permitindo a materialização de melhorias em processos existentes, como afirmam Perry, Porter e Votta (2000). Em suma, a sugestão de eficácia dos autores está na alteração da maneira como se conduz os experimentos. A alteração é dada pela agregação do estudo orientado a requisitos, que por sua vez é representado pela efetividade da melhoria do processo de desenvolvimento de software. A visão orientada à implementação é limitada por fatores de confusão.

Dada à natureza criativa e inovadora da disciplina de engenharia de software, torna-se essencial a análise do produto como também do processo de desenvolvimento de software. A necessidade de realizar experimentos nesta disciplina é primordial para o sucesso de qualquer novo método ou de qualquer nova ferramenta.

Um novo processo está baseado na seqüência temporal das atividades dos desenvolvedores, como também no grau de aceitação dos *stakeholders*. Portanto, é papel dos experimentos provarem as estratégias para os novos métodos, ferramentas ou técnicas, disponibilizando-os para efetivo uso.

2.5.1. Classificação dos Métodos de Avaliação Experimental

Os métodos de avaliação experimental podem ser classificados em três aspectos. O primeiro abrange os tipos de experimentos conforme o (1) paradigma analítico de investigação, sendo o método empírico o mais adequado à engenharia de software. O segundo aspecto relaciona-se às diferentes (2) formas de organização do experimento, em que os experimentos formais e os estudos de caso representam grande influência para o amadurecimento da área de engenharia de software. A terceira e última dimensão é quanto ao (3) objetivo da investigação, podendo estabelecer medidas ou formas de aceitação sobre um determinado método ou ferramenta. Assim, os três tipos de classificação dos métodos de avaliação experimental são descritas a seguir.

1) Paradigma Analítico de Investigação

Conforme Travassos, Gurov e Amaral (2002), existem quatro métodos para analisar e conduzir experimentos, a saber: científico, de engenharia, experimental, e analítico. Em resumo, o método científico tenta extrair do mundo algum modelo que possa explicar um fenômeno qualquer, focando no seu entendimento, por meio de indução dos fatos. Já o método de engenharia concentra-se na melhoria evolutiva, propondo novas soluções mais adequadas, com base naquelas já existentes. Não menos importante, o método empírico apresenta a melhoria revolucionária, por meio da experiência, observando os efeitos no produto ou processo a partir das mudanças sugeridas. Finalmente, o método matemático, ou analítico, recorre à dedução principalmente quando compara experimentos com teorias formais. Nota-se que esta classificação discute e abrange diversas situações na área de engenharia de software, na tentativa de torná-la menos imatura, estabelecendo uma base científica para entendimento dos componentes e seus respectivos relacionamentos.

Anteriormente aos autores citados, Basili, Rombach e Selby (1993) já considerava que o paradigma experimental requer: um projeto experimental, observações, coleção de dados e validação do processo de software ou produto estudado. Além disso, o autor também diferencia a maneira de conduzir os métodos de avaliação. Em sua análise, os dois métodos experimentais são: o científico e o matemático. Este definido como um paradigma indutivo, usado para entender processo, produto, pessoas ou ambiente; sempre na intenção de extrair modelos explicativos. Podendo ter variações definidas como método de engenharia, que observa soluções existentes; e como método empírico, que aplica estudo de caso. Aquele definido como um paradigma dedutivo, a partir das teorias formais propostas em comparação com resultados experimentais.

Na realização deste trabalho de mestrado, o método escolhido para conduzir a experimentação foi o *empírico*, organizado em forma de *estudo de caso*, devido a fatores como custo do experimento e perfil dos participantes.

Diversos autores propõem modelos para execução de métodos de avaliação. Na abordagem de experimentação GQM (*Goal/Question/Metric*) proposta por Basili, Caldiera e Rombach (1994), o experimento é direcionado por metas (*Goal*) específicas, que representam o cerne da investigação. Estas por sua vez, são melhor especificadas em questões (*Questions*), que caracterizam o caminho da avaliação. E, finalmente, as medidas (*Metrics*) servem para responder as questões de maneira quantitativa. O autor enfatiza que as medidas são mecanismos que auxiliam na resposta a diversas questões sobre o processo de software e, por isto, a abordagem GQM não precisa necessariamente ser usada de maneira isolada, mas poderia estar incorporada às equipes de projetos que gerenciam os negócios da empresa.

2) Formas de Organização do Experimento

Nas técnicas de avaliação apresentadas por Pfleeger (2004; 1994), o direcionamento depende do método escolhido e, também, da forma de organização do experimento para medição da qualidade proposta. A autora ainda apresenta alguns fatores relevantes para a seleção dos métodos de avaliação, justificando a condução de experimentos que variam conforme a natureza dos mesmos ou conforme critérios e restrições ora técnicos ora práticos.

Com o intuito de minimizar as dificuldades que naturalmente acontecem no processo experimental em engenharia de software, Kitchenham (1996a, p.12) e Kitchenham, Linkman e Law (1997, p.122) mostram algumas maneiras diferentes de organizar os experimentos, que são identificadas como: Experimento Formal, Estudo de Caso e *Survey*.

Resumidamente, experimento formal é usado quando a experimentação envolve muitos sujeitos utilizando diferentes métodos ou ferramentas para executar suas tarefas. A obtenção dos resultados, neste caso, é bastante imparcial. Visto que os sujeitos indicados para utilizar cada método ou ferramenta não são tendenciosos em suas tarefas executadas, nem na escolha de métodos ou ferramenta, o que permite perfeitamente o uso de técnicas estatísticas para análise e interpretação dos resultados.

Quando se organiza o experimento na forma de estudo de caso, os métodos e ferramentas são aplicados em projetos reais (projeto piloto), recorrendo aos procedimentos padrão da organização para o desenvolvimento de software.

E, por fim, quando a organização do experimento está na forma de *survey,* o objetivo é direcionar o estudo em retrospectiva, pois as informações são fornecidas por usuários que já utilizaram os métodos ou ferramentas e identificaram características descritivas que permite o aprofundamento necessário à investigação.

Segundo Pfleeger (2004), não existem apenas três categorias em que as técnicas de avaliação podem estar classificadas, mas identifica quatro classes, a saber: Experimento Formal, Estudo de Caso, Pesquisa de Opinião e Análise de Características. Em experimentos formais enfatiza-se a aleatoriedade na escolha de objetos experimentais, para que os mesmos não sofram influência de nenhuma técnica de seleção. E nos estudos de caso salienta a comparação dos resultados na utilização de um método, com os resultados no uso de outro.

Em visão complementar, experimento formal para Pfleeger (1994, 1995) é um tipo de técnica que exige que a investigação seja mais rigorosa do que a análise de características e menos simples do que a pesquisa de opinião, que é executada em retrospectiva. Experimento formal requer que todos os casos sejam investigados pelo experimento a fim de formar um modelo completo, para ser reutilizado em contextos semelhantes. Também exige preparo da técnica pelos participantes do experimento, contando com o rigor matemático e também com alto custo para o experimento.

Já a categoria de método de avaliação chamada estudo de caso, na qual este trabalho baseia-se, abrange o rigor dos modelos matemáticos, que usam testes estatísticos de hipóteses e conta com a possibilidade de definir adequadamente os objetivos de medição e análise.

Para entender a técnica de pesquisa de opinião, é preciso visualizar o estudo em retrospectiva, assim como no tipo *survey*, na tentativa de descrever as relações e os resultados de determinada situação. Esta técnica é muitas vezes usada para mostrar tendências e diferenças significativas em torno do processo de desenvolvimento de software adotado.

Por fim, Pfleeger (1994) identifica o método de avaliação chamado análise de características. Geralmente utiliza-se desta análise quando se faz necessário descobrir a relevância de uma nova técnica, uma nova ferramenta ou ainda um novo método inserido no contexto do trabalho atual de uma organização, que possui controle sobre o próprio processo de desenvolvimento de software. Esta categoria de avaliação fornece uma lista de respostas discretas, como a dicotomia Sim/Não, para as características comuns aos métodos e/ou ferramentas em demanda. A intenção desta é demonstrar, por meio da análise e representação dos resultados, as melhores características que deveriam estar presentes no novo método proposto, já que determinada característica, possuindo boa pontuação, determina qual método ou ferramenta é o melhor. Vale ressaltar que, análise de características é um método de avaliação subjetivo, pois é altamente dependente da opinião pessoal do avaliador (Pfleeger, 1994, 2004; Kitchenham, Linkman e Law, 1997).

3) Objetivo da Investigação

Não menos importante que a forma de organizar, ou a forma de analisar um experimento, a divisão em dois tipos de avaliação fornece a identificação de medidas quantitativas ou qualitativas para o experimento, a fim de tornar a análise dos dados mais precisa e útil.

Os diferentes tipos de avaliação, contendo objetivos diferentes, são: avaliações quantitativas e avaliações qualitativas. As avaliações podem estar direcionadas no estabelecimento de efeitos mensuráveis para a utilização de um método ou ferramenta ou ainda podem estar objetivando estabelecer o grau de aceitabilidade (*appropriateness*) de um método ou ferramenta em relação à população de usuário.

Quando a avaliação requer objetividade, ela é baseada na identificação de benefícios esperados a partir da utilização do novo método ou ferramenta, representado em termos de efeitos mensuráveis, por exemplo, tempo de retrabalho, custos com manutenção, entre outros. Este tipo de avaliação é também chamada de quantitativa.

Quando a avaliação caracteriza-se pelo aspecto subjetivo, ela é chamada qualitativa e é usada para analisar quão bem um método ou ferramenta completa as expectativas e necessidades da organização.

Segundo Kitchenham (1996a) e Kitchenham, Linkman e Law (1997), além desses dois tipos de avaliação, existe a avaliação híbrida, que envolve ambos os aspectos objetivos e

subjetivos. Devido a características deste trabalho de mestrado, optou-se pela avaliação *quantitativa*.

Na taxonomia dos métodos, segundo Kitchenham, Linkman e Law (1997), existem nove métodos diferentes de avaliação classificados como quantitativos, qualitativos e/ou híbridos.

Os Métodos Quantitativos estabelecem efeitos mensuráveis em relação à expectativa criada a partir da utilização do novo método ou ferramenta. No Quadro 2.1 está representada a taxonomia desses métodos.

Método de Avaliação	Caracterização e Condições Favoráveis
Experimentos	Investigação de um método ou ferramenta com alto grau de controle, de uma ou mais variáveis que representam o comportamento dos eventos. Os resultados dos experimentos são normalmente mais generalizados que os estudos de caso. Condições Favoráveis: benefícios claramente quantificáveis, pessoal disponível para executar o experimento, método ou ferramenta relacionado a uma única tarefa, tempo relativamente pequeno para aprendizado
Estudos de Caso	Investigação de um método ou ferramenta que não requer replicação, contudo obtém informações limitadas. Envolve a avaliação de métodos ou ferramentas após terem sido usadas em projetos reais de software. Condições Favoráveis: benefícios quantificáveis em um único projeto, procedimentos estáveis de desenvolvimento, pessoal com experiência em mensuração.
Surveys	Investigação de um método ou ferramenta, antes do seu efetivo uso, demonstrando apenas associações e não causalidade. Condições Favoráveis: benefícios não são quantificáveis em um único projeto, existência de dados sobre o método ou ferramenta em análise, experiência em projetos que usam as ferramentas.

Quadro 2.2 – Taxonomia dos métodos de avaliação quantitativos

Os Métodos Qualitativos estabelecem o grau de adequação de determinado método ou ferramenta em relação a sua população de usuários. A taxonomia desses métodos é apresentada na Quadro 2.3.

Método de Avaliação	Caracterização e Condições Favoráveis
Análises de Características – Modo *Screening*	Investigação de um método ou ferramenta feita por uma única pessoa (ou grupo coeso), que não apenas determina as características a serem avaliadas, mas também as escalas da avaliação. Condições Favoráveis: grande número de métodos e ferramentas a serem avaliados.
Análises de Características – Experimento	Investigação de um método ou ferramenta feita por um grupo de potenciais usuários que são indicados para usar as ferramentas e métodos e tarefas típicas antes de realizar as suas avaliações. Usa seleção aleatória para o grupo de potenciais usuários. Condições Favoráveis: benefícios dificilmente quantificáveis, população de usuários bastante variada.
Análises de Características – Estudo de Caso	Investigação de um método ou ferramenta feita por pessoas que tem usado o método ou ferramenta em projetos reais, ou seja, na prática. Condições Favoráveis: benefícios dificilmente quantificáveis, benefícios observados em um único projeto, população de usuários limitada, procedimentos estáveis de desenvolvimento.
Análises de Características – *Survey*	Investigação de um método ou ferramenta feita por pessoas que têm experiência no uso da ferramenta ou método, ou ainda tem estudado as ferramentas ou métodos. Condições Favoráveis: benefícios dificilmente quantificáveis, população de usuário bastante variada, benefícios não observados em um único projeto.

Quadro 2.3 – Taxonomia dos métodos de avaliação qualitativos

Os Métodos Híbridos envolvem ambos os aspectos subjetivos e objetivos de avaliação e estão relacionados na Quadro 2.4.

Método de Avaliação	Caracterização e Condições Favoráveis
Análises de Efeitos Qualitativos	É uma avaliação subjetiva dos efeitos quantitativos dos métodos ou ferramentas, baseado em opinião de *experts* como engenheiros e gerentes *seniores*. Condições Favoráveis: disponibilidade das avaliações de ferramentas e métodos dos *experts*, falta de procedimentos estáveis de desenvolvimento, interesse em avaliação de métodos e ferramentas genéricas.
Benchmarking	É um processo que executa vários testes padrão que usam ferramentas alternativas para avaliar o desempenho relativo dessas ferramentas contra aqueles testes. Condições Favoráveis: saídas dos métodos e ferramentas disponíveis para serem listadas em termos de critérios de adequação (*goodness*).

Quadro 2.4 – Taxonomia dos métodos de avaliação híbridos

A seleção de um método de avaliação adequado nem sempre é uma tarefa fácil e segura. Portanto, Kitchenham (1996b, 1996c) destaca alguns critérios técnicos e práticos que podem influenciar nessa escolha. Critérios para a seleção como o contexto da avaliação e a natureza do impacto esperado no uso do novo método ou ferramenta são categorias relevantes, já que esses eventos acontecem no início do processo de experimentação. Na classificação de critérios do tipo técnico ainda são inclusos os que abordam a natureza do objeto a ser avaliado, a maturidade do método ou ferramenta, a curva de tempo de aprendizado associada ao método ou ferramenta e, finalmente, a capacidade de mensuração que a organização possui em avaliações.

Quando um exercício de experimentação inicia-se, podem existir limitações ou restrições práticas como: a influência de fatores humanos e sociológicos, como motivação e expectativa dos participantes; prazos; riscos e custos associados ao método de avaliação escolhido.

Nota-se que na escolha por um método como os experimentos formais, os custos são mais elevados em comparação a outras categorias de método de avaliação. Percebe-se claramente, também, que os riscos nos resultados diminuem em experimentos formais, em detrimento aos custos associados na aplicação de vários projetos com objetivos de mensuração diferentes.

Para Pfleeger (1995), o experimento segue um caminho científico caracterizado por um planejamento cuidadoso como também execução e análise minuciosa dos procedimentos e dados. Por isso, afirma ser o experimento tarefa para os desenvolvedores, no assunto de novas

tecnologias inseridas ao processo que já se segue, ou ainda na grande área de engenharia de software.

2.5.2. Processo Experimental

Para Pfleeger (2004), experimento é um mecanismo de avaliação em que uma nova técnica ou ferramenta é submetida, com o propósito de mostrar se existe aperfeiçoamento, ou ainda, adequação para determinado uso.

As diretrizes para realização de estudos de caso, segundo Kitchenham, Pickard, Pfleeger (1995, p.55) e Kitchenham e Pickard (1998, p.25) são: Identificar o Contexto do Estudo de Caso, Definir Hipóteses, Selecionar os Projetos Piloto, Identificar o Método de Comparação, Minimizar os Efeitos dos Fatores de Confusão, Planejar o Estudo de Caso, Monitorar o Estudo de Caso contra o Planejado e Analisar e Reportar os Resultados.

a) Identificar o Contexto do Estudo de Caso

Esta tarefa estabelece as metas e restrições nas quais o estudo de caso deve operar. As metas estão relacionadas com o que o patrocinador deseja conhecer e também relatam o custo do investimento associado à tecnologia investigada. As restrições auxiliam no planejamento do estudo de caso, identificando o patrocinador, os recursos disponíveis para organização e execução, prazos e a importância do estudo de caso.

b) Definir Hipóteses

Esta é uma tarefa relevante, pois a condução de todo estudo de caso depende da especificação das hipóteses nula e alternativa, também é a tarefa responsável pela definição do efeito esperado do novo método. Ela deve ser detalhada o suficiente para permitir a identificação de medidas que são necessárias para validar o experimento. É importante também definir o que não é esperado que aconteça.

c) Selecionar os Projetos Piloto

Nesta tarefa são identificados os projetos representativos da organização, podendo descrevê-los em termos de variáveis estáveis como: domínio da aplicação, linguagem de programação associada, método de projeto e grau de reuso.

d) Identificar o Método de Comparação

Esta tarefa é responsável por permitir o contraste dos resultados da utilização de um método com outro. Existem três formas para organizar o estudo que facilitam esta comparação, são elas: comparar os resultados com projetos semelhantes, comparar resultados do uso do novo método com uma *baseline* da organização e comparar se o método usado em componentes individuais aplica-se também aleatoriamente em outros componentes.

e) Minimizar os Efeitos dos Fatores de Confusão

Esta tarefa é realizada quando um fator não pode ser adequadamente diferenciado do efeito de outro fator, então os fatores são confundidos. Fatores de confusão podem afetar a validade interna do estudo de caso, como usar o estudo com mecanismo de aprendizado, usar participantes que estejam muito céticos ou entusiasmados com o novo método ou ainda realizar comparação com diferentes tipos de domínios de aplicação. Segundo Pfleeger (2004, p.421), se o experimento apresenta evidências insuficientes para auxiliar na tomada de decisão, por exemplo, investir na nova tecnologia investigada, é possível recorrer a outros estudos que tratam as ciladas como confusão, prazo curto demais, situação errada entre outros.

f) Planejar o Estudo de Caso

Esta tarefa contribui para a confiabilidade do experimento. Basili, Rombach e Selby (1993) sugere a construção de um plano de avaliação composto por definição das informações, planejamento dos critérios e formas de medição do estudo, operação juntamente com a análise dos dados e interpretação dos resultados.

g) Monitorar o Estudo de Caso em relação ao Planejado

Nesta tarefa é assegurado que os métodos ou ferramentas sejam usados corretamente, executando atividades de auditoria e reportando as mudanças ocorridas, incluindo recomendações.

h) Analisar e Reportar os Resultados

Esta tarefa depende do número de itens de dados que deverão ser analisados, estes estão associados a valores das variáveis de respostas disponíveis do experimento. Podem ser usados métodos estatísticos tais como análise de variância, tabela de contingência, testes não paramétricos entre outros.

O processo experimental proposto por Kitchenhan, Pickard (1998) e Kitchenham, Pickard e Pfleeger (1995) é composto por fases com ênfases diferentes e todas com o intuito de assegurar a condução do experimento.

Assim como para os outros autores, para Basili, Selby e Hutchens (1985) a experimentação também é executada em etapas, auxiliando assim na obtenção de melhores avaliações, predições, entendimentos, controle e melhoria dos processos de desenvolvimento de software e produto. O *Framework* Experimental proposto pelo autor consiste em quatro fases genéricas do processo experimental, a saber: Definição, Planejamento, Operação, Interpretação.

1) Definição do Experimento

Esta fase incluí a tarefa de determinar as metas do experimento, assim como a motivação e os objetos experimentais, sob pontos de vista relevantes para determinado projeto.

2) Planejamento

Nesta fase são abordados temas como a definição do projeto e métodos analíticos que serão utilizados no experimento, como também os critérios e restrições de projeto e medidas adequadas à experimentação.

3) Operação

Esta fase inclui o estudo do projeto piloto, confirmando o cenário para o experimento. A análise dos dados inclui a combinação de métodos qualitativos e quantitativos. O processo de análise dos dados requer uma investigação preliminar, como histogramas, antes da aplicação de testes e métodos estatísticos.

4) Interpretação dos resultados

Esta é a última fase e pode ser realizada em diversos contextos, dependendo do paradigma estatístico escolhido para análise dos dados.

Em suma, o *framework* experimental oferece a estrutura necessária para representar os estudos de caso e também valorizar a experimentação na engenharia de software.

No processo experimental utilizado para avaliar uma nova técnica ou ferramenta, pode-se utilizar a abordagem GQM (*Goal/Question/Metric*) que, segundo Travassos, Gurov e Amaral (2002), é um paradigma que faz parte dos princípios da melhoria do processo de software, claramente discutido no QIP (*Quality Improvement Paradigm* - Paradigma da Melhoria da Qualidade).

A abordagem GQM, proposta por Basili, Caldiera e Rombach (1994), objetiva especificar metas particulares ao contexto da organização com o intuito de observar características do produto e do processo de desenvolvimento de software. O resultado da aplicação desta abordagem reflete em especificações de software mensuráveis, juntamente com um conjunto de regras para interpretação dos dados de mensuração.

O modelo de mensuração é dado em três níveis: nível conceitual (*Goal*), nível operacional (*Question*) e nível quantitativo (*Metric*). O modelo começa com a meta, que especifica o propósito da medida, objeto a ser medido, assuntos a serem mensuradas e o ponto de vista em que a medida será avaliada. Então a meta é refinada em várias questões, isto significa dividir o assunto em importantes componentes. Cada questão é refinada em métricas, algumas vezes como objetivas e outras vezes como subjetivas.

Portanto, a abordagem GQM é um mecanismo para definir, interpretar operacionalmente e mensurar o software. Recomenda-se que seja utilizada em conjunto com a abordagem mais abrangente chamada QIP.

2.6 TRABALHOS RELACIONADOS

Esta seção apresenta, brevemente, alguns trabalhos relacionados a esta dissertação de mestrado. Alguns destes trabalhos são modelos de adaptação do UP em áreas diferentes a área de KDD. Eles mostram a possibilidade de adaptação de UP, com sucesso em relação à criação

de modelos, artefatos, papéis e atividades específicas conforme domínio de aplicação escolhido. Outros trabalhos apresentam abordagens no contexto de metodologias e arquiteturas de DW e KDD. Também estão relacionados alguns trabalhos que utilizaram o metamodelo SPEM e também experimentação em engenharia de software.

2.6.1. Abordagens de Metodologia e Arquiteturas de DW e KDD

Luján-Mora (2005) propôs a definição de um método cujo conjunto de modelos poderá ser usado pelos projetistas de DW para mostrar seus projetos para os usuários finais. O método permite que o projetista seja conduzido por diferentes fases e passos para projetar o DW. Estas fases são baseadas no UP e o método envolve a utilização de mecanismos de extensão da UML, XML (*Extensible Markup Language*), banco de dados orientado a objetos e banco de dados objeto-relacional. Primeiramente, o autor divide a Modelagem de Dados em níveis como conceitual, lógico e físico, depois caracteriza o processo de engenharia de DW.

Neste trabalho existem contribuições na incorporação de dois *workflows* (Manutenção e Revisão Pós-Desenvolvimento) ao UP, e também na elaboração de *Profiles UML* (para Modelagem Multidimensional, para funções ETL, para implantação da Base de Dados). Como também há contribuição na criação de um novo tipo de diagrama denominado Mapeamento de Dados com vários níveis de detalhes do ambiente de DW. Para apoiar o *Profile UML* de Modelagem Multidimensional, o autor desenvolveu um *add-in* para o Rational Rose.

Borba (2006) propôs uma metodologia para implementação da modelagem multidimensional, cujo conjunto de técnicas e processos possibilita a persistência de dados, especialmente em banco de dados orientados a objeto. Para representar de maneira conceitual os modelos multidimensionais, que é uma das cinco tarefas da metodologia proposta, a autora optou por Diagramas de Classes e de Estrutura Composta definidos pela UML 2.0. Por sua vez estes são formalizados em ODL (*Object Definition Language*), possibilitando a implementação do modelo em banco de dados orientado a objetos. Assim, por meio do estudo de caso realizado, esta metodologia apresenta, dentre uma das contribuições, a integração do processo completo da modelagem à implementação do modelo multidimensional no

paradigma OO, pois os outros métodos de modelagem multidimensional não abordam o processo completo e nem todas usam como referência para a representação de dados a UML.

Dias (2001) propôs um modelo de formalização do processo de desenvolvimento de sistemas KDD. Para representar os objetos do sistema foram usados diagramas da UML, mapeando-os posteriormente por meio do formalismo denominado E-LOTOS. Este trabalho engloba também a tecnologia de agentes inteligentes na especificação de ambiente de implementação de sistemas KDD. A principal contribuição deste trabalho é dada pela incorporação de uma linguagem de especificação formal como modelo de metodologia para o desenvolvimento de sistemas KDD.

Menolli (2004) propôs a definição de uma arquitetura de *data warehousing*, juntamente à construção de um data warehouse com a possibilidade de integração das bases institucionais do CNPq e CAPES com dados regionais. Os resultados obtidos pelos estudos de casos realizados mostram que o DW construído em conformidade à arquitetura proposta é eficiente para sistemas KDD.

Valentim (2006) propôs a definição de uma arquitetura para sistemas KDD, especialmente os que integram as funcionalidades do sistema em uma única ferramenta. Neste sentido, o trabalho oferece a contribuição do modelo de referência para sistemas KDD, como também uma arquitetura de referência para solucionar problemas arquiteturais.

2.6.2. Adaptação de Processos

Sousa (2004) propôs uma abordagem para separação de preocupações transversais, desde o início do processo de desenvolvimento, considerando o paradigma orientado a aspectos e as fases e *workflows* do UP. As contribuições deste trabalho apresentam melhoria no reuso, manutenção e compreensão dos artefatos gerados no processo de desenvolvimento.

Existe uma abordagem de adaptação do RUP para o domínio de Jogos Móveis proposto por Almeida (2006). Este trabalho propõe adequações que partem do UP discutindo papéis, artefatos e fases envolvidos na produção de jogos móveis. Como contribuição deste trabalho é a solução de processo para o domínio de jogos móveis.

Em Álvares (2001) foi definido um processo para aplicações web, especificamente para o sistema e-Merci. O trabalho é caracterizado por personalizar o processo Práxis para a realidade de aplicações Web, denominado WebPraxis. Houve inserção de fluxos principalmente relacionados à usabilidade, e mostrou-se correto quanto a ordem temporal e lógica das atividades.

A adaptação mais conhecida pelo meio acadêmico e industrial é o RUP. Nesta adaptação, conforme descrito por Kruchten (2003), são inseridos três *workflows* gerenciais, que atendem adequadamente a realidade empresarial como gerência de projeto e controle do ambiente envolvendo atividades de treinamento.

2.6.3. Instanciação de Processos com Modelagem seguindo o SPEM

Genvigir (2004) propõe em seu trabalho a organização da engenharia de requisitos de software usando as técnicas de modelagem de processos definidas no metamodelo SPEM. O autor realiza um experimento comprovando a qualidade do processo proposto.

No processo proposto por Bulcão Neto (2006) para aplicações sensíveis ao contexto, o SPEM é usado para modelar os principais aspectos do processo, representando fielmente a visão geral do mesmo, assim como as particularidades de cada atividade proposta.

2.6.4 Experimentação em Engenharia de Software

Oliveira Junior (2005) propôs um processo que possibilita a identificação, representação, delimitação, escolha de mecanismos de implementação e monitoração das variabilidades em todo o ciclo de vida de uma Linha de Produção. Como uma das contribuições deste trabalho foi a realização de um estudo de caso para avaliar o processo proposto, sendo que este usou as técnicas da experimentação em engenharia de software.

Farias (2002) propôs uma definição de uma abordagem para o planejamento de riscos em projetos de software, baseada na reutilização do conhecimento organizacional de riscos.

Uma das contribuições deste trabalho foi o modelo do plano do estudo experimental servindo de base para estudos similares.

2.7. CONSIDERAÇÕES FINAIS

A fundamentação teórica destacada neste capítulo serviu de base para construir o processo de software. A partir da consulta aos trabalhos relacionados foi possível visualizar que a composição de um processo de software requer muito mais que motivação e oportunidade.

Por meio das investigações realizadas na estrutura dinâmica e estática do UP pode-se perceber a adequação do mesmo à realidade de aplicações KDD. Devido a aspectos complexos deste tipo de aplicação, exige-se ordenar atividades para conduzir tanto a descoberta de conhecimento quanto a construção da estrutura de dados que suporte tecnologias analíticas como DWs.

A pesquisa por metodologias de modelagem de processo oferece uma variedade de opções que mudam conforme objetivos de medição. Como é o caso do metamodelo SPEM, que representa completamente um processo modelado.

Os conceitos fundamentais da experimentação em engenharia de software permitem visualizar a importância de avaliar novos métodos, ferramentas e processos. A validação de determinada ferramenta depende da opinião de quem precisa de uma solução automatizada para controle de procedimentos. A maioria dos métodos de avaliação oferecem resultados quantitativos ou qualitativos para o engenheiro de software, auxiliando-os nos processos de sua responsabilidade.

No próximo capítulo é apresentada a especificação do processo UPKDD, mostrando o que compõe uma aplicação KDD e também são detalhados os modelos de processo desde a visão geral até a visão detalhista de cada disciplina importante para este processo modelado.

3 UM PROCESSO PARA APLICAÇÕES KDD

Neste capítulo são apresentados os modelos de processo utilizados para especificar o UPKDD (*Unified Process for Knowledge Discovery in Database)*, seguindo o metamodelo SPEM, que oferece uma estrutura de modelagem de processos baseada na UML. Primeiramente, uma visão geral do processo é descrita, permitindo entender o tipo de aplicação que o modelo de processo de software se propõe a apoiar. Na seqüência, as principais motivações são evidenciadas e são explicados os detalhes de cada área fundamental (Disciplina) para o desenvolvimento de sistemas de descoberta de conhecimento em banco de dados.

3.1 UMA VISÃO GERAL DO PROCESSO UPKDD

A área de engenharia de software representa a combinação de três elementos principais, a saber: métodos para construção de sistemas, ferramentas automatizadas e procedimentos adequados para a sucessão de métodos que deverão ser aplicados. Todos estes elementos interagem com a finalidade de tornar-se fundamento para o desenvolvedor produzir softwares com qualidade e confiabilidade.

Neste cenário, o processo de software oferece uma seqüência ordenada de atividades relacionadas com a especificação, projeto e implementação, assim como validação e evolução dos softwares, transformando expectativas dos usuários em soluções automatizadas (Pressman, 2006; Sommerville, 2003).

A escolha por um dos paradigmas da engenharia de software é dada pela natureza da aplicação como também pela caracterização do projeto de software.

Na área de KDD, Fayyad, Piatetsky-Shapiro e Smyth (1996) afirmam que o processo KDD é uma atividade multidisciplinar responsável por mapear os dados de baixo nível para outros formatos mais compactados, mais abstratos ou mais úteis para a descoberta de conhecimento em banco de dados. Os autores reforçam que atualmente a análise manual de dados torna-se impraticável, devido ao volume crescente dos mesmos, caracterizando assim

aplicações analíticas que valorizam os métodos e técnicas como mineração de dados, OLAP (processamento analítico *on-line)* e *data warehousing*.

Além de o processo KDD depender de técnicas analíticas, Brachman e Anand (1996) apontam que este deve ser centrado na interação constante entre usuários e banco de dados, denominado como aplicações de descoberta dirigidas a usuários finais. Para os autores, a presença do tomador de decisão torna-se decisiva para este tipo de aplicação, já que os dados analisados e extraídos buscam metas definidas pelos usuários finais. Estes discordam da maioria dos autores, que incentiva o uso de um ambiente KDD genérico, implementando várias técnicas de mineração de dados. Por outro lado, eles defendem os benefícios de aplicações específicas de KDD em conformidade aos objetivos dos tomadores de decisão.

Então o processo KDD caracteriza-se por conduzir os passos para descoberta de conhecimento, aplicando algoritmos específicos para extração de padrões de dados, podendo utilizar a estrutura de dados dimensional para esta tarefa. E, caso haja participação intensa do tomador de decisão interagindo com estes passos, as chances de obter sucesso para a organização são maiores.

Em se tratando de uma aplicação KDD, as atividades do desenvolvedor não se restringem à descoberta de conhecimento propriamente dita, ou ainda ao processo que conduz à descoberta, mas sim à integração destes ao processo de desenvolvimento de software.

Neste sentido, o processo UPKDD apresenta um processo de software para apoiar o desenvolvimento de aplicações de descoberta de conhecimento, especialmente aquelas cujas informações estão em banco de dados.

O UPKDD tem como referência algumas diretrizes, a saber: (1) condução do processo KDD baseado principalmente em Fayyad, Piatetsky-Shapiro e Smyth (1996) e Brachman e Anand (1996), (2) definição e implementação da arquitetura de DW segundo Kimball e Ross (2002), (3) boas práticas do *Unified Process* (UP) segundo Jacobson, Booch e Rumbaugh (1999) e (4) o Conjunto Comum de ECs (Elementos-Chave), (Apêndice A). Vale ressaltar que, os princípios, as fases, os *workflows* e também os elementos-chave do processo proposto, são similares aos estabelecidos no processo de software UP, assim como as estruturas estática e dinâmica.

A nomenclatura que define o UPKDD baseia-se na notação do SPEM conforme é mostrada no Quadro 3.1.

Estereótipos	Descrição	Notação
Produto de Trabalho	Um Produto de Trabalho é a descrição de um pedaço de informação ou uma entidade física produzida ou usada por atividades de um processo de engenharia de software. Exemplos de produtos de trabalho incluem: modelos, planos, códigos executáveis, documentos, banco de dados entre outros.	
Definição de Trabalho	É um elemento do modelo[13] do processo que descreve a execução, as operações desempenhadas e as transformações realizadas em um Produto de Trabalho por papéis. Atividade, Interação, Fase e Ciclo de Vida são tipos de Definição de Trabalho.	
Orientação (*Guidance*)	É um elemento do modelo associado aos principais elementos de definição do processo, contendo informação adicional tais como: técnicas, *guidelines* e *profiles* UML, procedimentos, padrões, *templates* e exemplos de produtos de trabalho e definições.	
Atividade	É uma Definição de Trabalho que descreve *"o que"* um Papel no Processo executa. Atividade é o principal elemento de trabalho.	
Executor do Processo	Um Executor do Processo é um elemento do modelo que descreve os proprietários das Definições de Trabalho. São usados para Definições de Trabalho que não podem associar-se com os Papéis no Processo, tais como um Ciclo de Vida ou uma Fase.	
Papel no Processo	É um elemento do modelo que descreve os papéis, responsabilidades e competências de um indivíduo que realiza Atividades dentro de um Processo e é responsável por determinado Produto de Trabalho.	

[13] Um Elemento do Modelo descreve um aspecto do processo de engenharia de software.

Estereótipos	Descrição	Notação
Pacote de Processo ou Disciplina	Uma Disciplina é um Pacote de Processo organizado na perspectiva de uma das disciplinas de engenharia de software, por exemplo, gerenciamento de configuração, análise e projeto entre outras.	
Fase	É uma Definição de Trabalho de alto nível, limitada por um *milestone*.	
Processo	Um Processo é uma descrição completa do processo de engenharia de software, em termos de Executores do Processo, Papéis no Processo, Definições de Trabalho, Produtos de Trabalho e Orientações associadas.	
Documento	É um tipo de Produto de Trabalho.	
Modelo UML	É um tipo de Produto de Trabalho.	

Quadro 3.1 – Notação e descrição dos estereótipos do metamodelo SPEM

Uma aplicação KDD, assim como qualquer aplicação da engenharia de software, requer a elaboração de uma arquitetura de software. Neste sentido, acredita-se que o modelo de arquitetura de software para sistemas KDD proposto por Valentin (2006), seja adequadamente suficiente para apoiar este tipo de aplicação.

Em suma, o UPKDD é fundamentado nos princípios da engenharia de software e divide-se em fases, que são Concepção e Elaboração. Da mesma forma, divide-se em disciplinas, que são Requisitos, Análise e Projeto. Este trabalho de mestrado não considera as disciplinas de Implementação e Teste, nem as fases de Construção e Transição, devido à incerteza e dependência do ambiente de implantação da solução KDD.

Para a modelagem do processo são usados os diagramas de pacotes, atividades, caso de uso e classes indicados pelo metamodelo SPEM, tanto para representar a visão geral, como para representar cada uma das disciplinas do UPKDD. Para usar os estereótipos do SPEM, foi necessário configurar uma ferramenta que funcionasse como um ambiente de apoio ao

processo modelado (PSEE - *Process Centered Software Engineering Environment*). A ferramenta escolhida foi o Rational Rose[14] da IBM e no Apêndice B é mostrado o código alterado juntamente à tela com os estereótipos instanciados no ambiente.

A visão geral do UPKDD é representada pelas Figuras 3.1 e 3.2, mostradas a seguir.

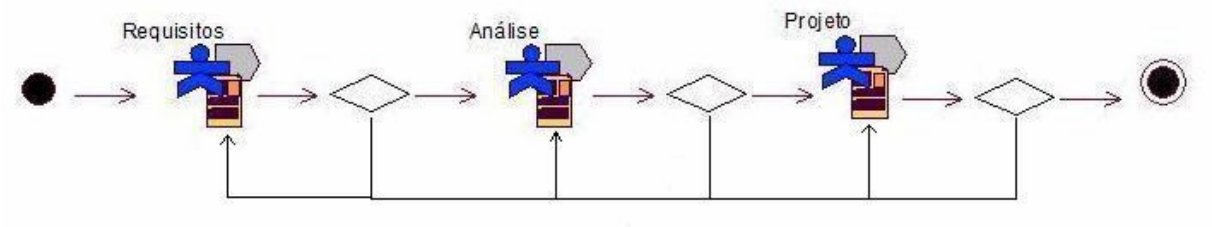


Figura 3.1 – Diagrama de atividades do processo UPKDD

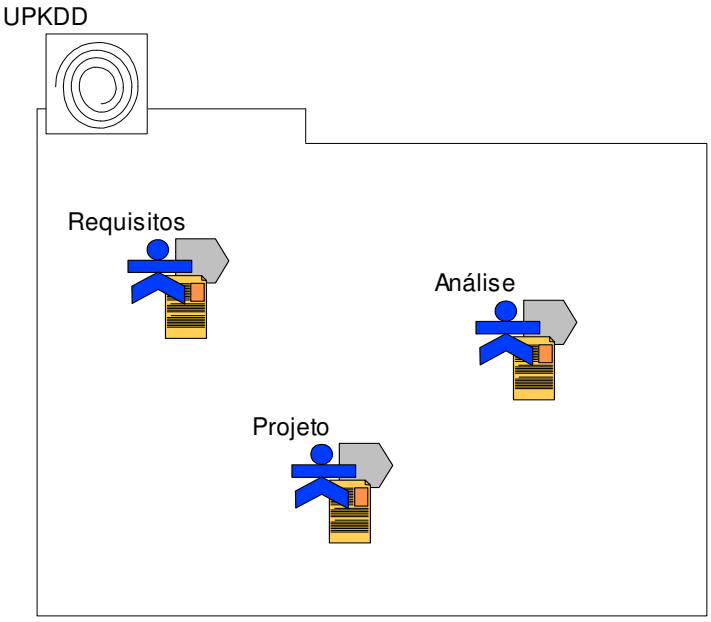


Figura 3.2 – Diagrama de pacote do processo UPKDD

No diagrama de atividades da Figura 3.1 é representada a ordem em que as Disciplinas são executadas. As flechas representam iteração no processo. Estas indicam que é possível modificar tanto as especificações de Requisitos de uma aplicação KDD, como também os modelos gerados em Análise e Projeto, análogo ao princípio iterativo e incremental do UP.

Outra maneira de representar a visão geral do processo é mostrar o conjunto de elementos que o compõem. Os diagramas de pacotes permitem representar os componentes do

[14] http://www.ibm.com/br/

processo que são as Disciplinas, e estas por sua vez representam disciplinas da engenharia de software que necessitam ser especificadas em várias perspectivas diferentes.

No diagrama de pacotes, mostrado na Figura 3.2, o nível de abstração é grande, mas o objetivo é mostrar que todas as Disciplinas possuem igualdade de condições no processo de desenvolvimento de software. Ainda que a atenção maior esteja em Requisitos, devido ao risco de projeto como má elicitação dos mesmos, a preocupação para o desenvolvimento de aplicações KDD estende-se pelas outras disciplinas, refinando e validando as expectativas dos usuários quanto ao conhecimento que será visualizado posteriormente.

Em resumo, componentes de um processo, como o UPKDD, formam agrupamentos lógicos, representados como Disciplinas, que são organizados de maneira a destacar pontos de vista importantes. Para entender esses grupos também chamados de Pacotes de Processo, é preciso investigar seus respectivos componentes. O primeiro elemento do modelo investigado é o Papel no Processo, descrito a seguir.

3.1.1 Papel no Processo

Para identificar os papéis de um processo, é preciso entender a dependência entre eles e suas responsabilidades no desenvolvimento do software. No UPKDD, os papéis foram definidos a partir das diretrizes já conhecidas do UP e, também, das diretrizes para a condução do processo KDD, que são mostrados no Quadro 3.2.

Papéis no Processo	Descrição
Engenheiro de Conhecimento	Responsabilidade ou Competência que realiza aquisição de conhecimento a partir das informações de negócio (conhecimento do domínio) da empresa.
Especialista de KDD	Responsabilidade ou Competência que realiza aplicações analíticas (mineração de dados ou OLAP), em resposta ao contexto da tomada de decisão.
Usuário Final (Tomador de Decisão)[15]	Responsabilidade ou Competência que realiza visualizações de informações extraídas, assim como determina quais motivações/necessidades de busca.

Quadro 3.2 – Papéis no Processo do UPKDD

[15] Neste trabalho os termos *Usuário Final* e *Tomador de Decisão* são usados sem distinção.

Os Papéis no Processo identificados podem participar de duas maneiras diferentes, a saber: como <<perform>> ou como <<assist>>/<<assistant>>. No caso do papel aparecer como <<perform>> representa que este executa uma atividade efetivamente. Porém, algumas vezes ocorrem situações em que um papel auxilia outro papel no desempenho de sua atividade, neste caso aquele quem apóia é definido como <<assist>>/<<assistant>>. Estes estereótipos são melhor visualizados nos diagramas que especificam as Disciplinas como os de caso de uso.

Cada uma das Disciplinas do UPKDD é representada por quatro diagramas diferentes, sendo que cada um deles enfatiza um aspecto relevante para o desenvolvedor de aplicações KDD. A representação segue a ordem de desenvolvimento do software. Esta se inicia em Requisitos, Análise e é finalizada em Projeto. À medida que uma Disciplina é consultada, a sua dinâmica é descrita nos diagramas de caso de uso e atividade e os aspectos de relacionamento e organização dos elementos do modelo nos diagramas de pacote e classe.

3.2. DISCIPLINA REQUISITOS

Durante o desenvolvimento de uma aplicação KDD, entende-se por requisitos a especificação da estrutura para tomada de decisão, agregada à especificação do desenvolvimento de aplicações tradicionais. Os detalhes desta Disciplina, mostrados na Figura 3.3, são vistos ainda em nível alto de abstração, em que esta é dividida em preocupações quanto às expectativas dos usuários, aos eventos dos dados existentes e às ferramentas que apoiarão a seleção, limpeza, extração e visualização das informações.

A Disciplina Requisitos e as demais Disciplinas do UPKDD são compostas por Definições de Trabalho que, além de serem mecanismos de divisão semântica para as Atividades do Processo, também mostram como um elemento do modelo pode ser especificado em vários níveis de exigência conforme a necessidade. As Definições de Trabalho explicam os relacionamentos entre os elementos do modelo.

A seqüência de Definições de Trabalho na Disciplina de Requisitos é Compreender o Contexto da Decisão, Compreender o Contexto de Dados e Compreender o Contexto de Ferramentas.

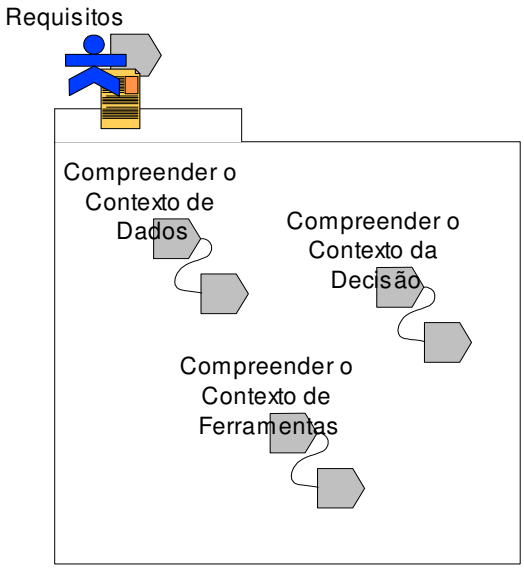


Figura 3.3 – Diagrama de pacote da Disciplina Requisitos

3.2.1 Definição de Trabalho – Compreender o Contexto de Decisão

Compreender o Contexto da Decisão, mostrado na Figura 3.4, é importante porque as aplicações KDD dependem fortemente das expectativas dos usuários finais. Sendo assim, o entendimento dos dados existentes e a delimitação e direcionamento do processo de descoberta de conhecimento fazem parte do contexto para futuras tomadas de decisão por parte dos usuários.

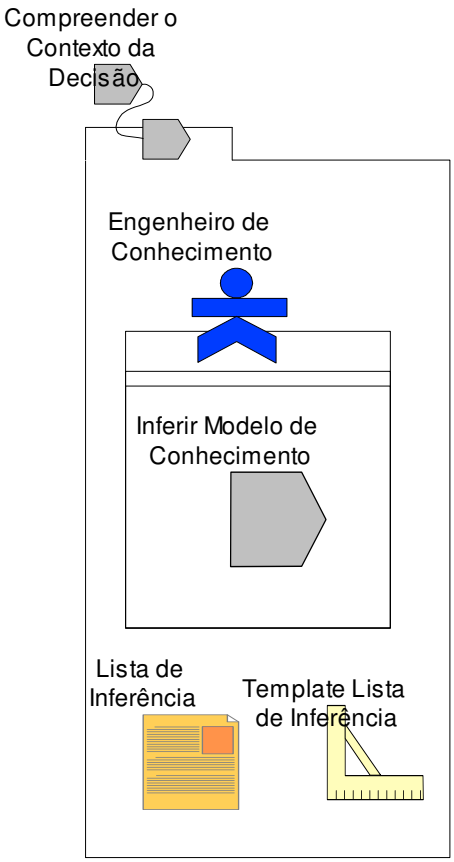


Figura 3.4 – Diagrama de pacote da Definição de Trabalho - Compreender o Contexto da Decisão

Para tornar a atividade o mais realista possível, sugere-se construir um artefato denominado Lista de Inferência[16], que significa reunião de proposições ou questões elaboradas a partir da base de dados transacional da empresa. Estas são comprovadas pelas investigações na base de dados após transformação em modelo dimensional. O modelo da Lista de Inferência está representado no Quadro 3.3.

Template Lista de Inferência
Relação de Inferências:
{representar proposições a serem provadas pelas investigações na base de dados, sugestão iniciar a frase com "por que"}

Quadro 3.3 – *Template* Lista de Inferência

[16] **Inferência** significa Raciocínio, Dedução, Indução (Ferreira, A. B. de H. Novo dicionário da língua portuguesa. 6 ed. Curitiba: Positivo, 2006). Neste trabalho **Inferir** não recorre aos benefícios dos mecanismos de inferência oriundos da área da Inteligência Artificial (IA), para isto dependeria da forma de armazenamento anterior das informações. Porém **Inferir** neste trabalho converge para o modelo conceitual de processamento do conhecimento da IA.

Fayyad, Piatetsky-Shapiro e Smyth (1996) indica que o direcionamento de uma aplicação sob qualquer método de mineração de dados pode ser uma atividade perigosa, no sentido de descobertas de padrões insignificantes e inválidos. Portanto, a atividade Inferir Modelo de Conhecimento tem o intuito de gerar a Lista de Inferência, diminuindo a distância entre o conhecimento esperado pelo tomador de decisão e o conhecimento existente na base de dados. Descobrindo-se assim, conhecimentos válidos para a população de usuários finais que participou do direcionamento do processo KDD. Neste sentido, as Figuras 3.5, 3.6 e 3.7 mostram como gerar o artefato Lista de Inferência e quem participa desta atividade.

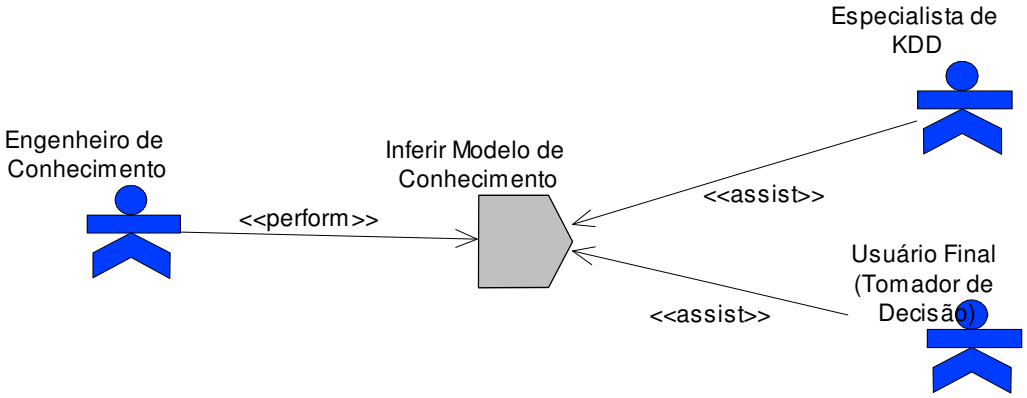


Figura 3.5 – Diagrama de caso de uso da Definição de Trabalho - Compreender o Contexto da Decisão

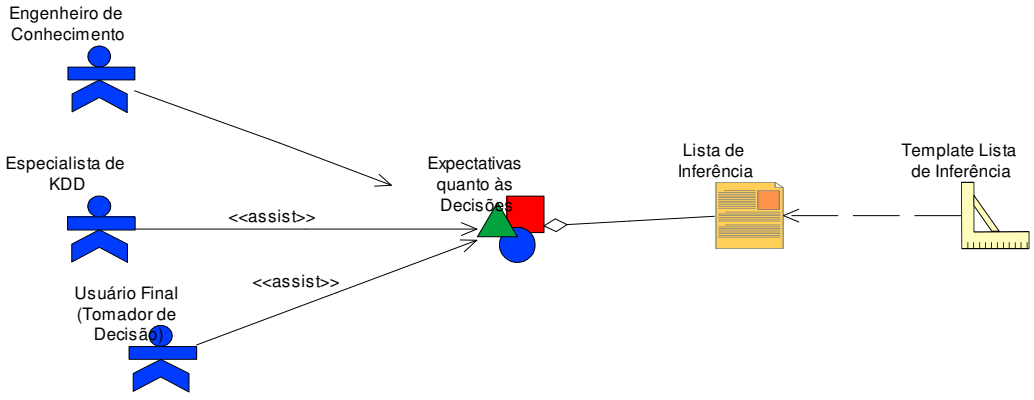


Figura 3.6 – Diagrama de classes da Definição de Trabalho - Compreender o Contexto da Decisão

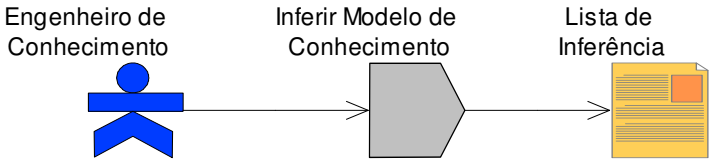


Figura 3.7 – Diagrama de atividade da Definição de Trabalho - Compreender o Contexto da Decisão

3.2.2 Definição de Trabalho – Compreender o Contexto de Dados

Compreender o Contexto de Dados, mostrado na Figura 3.8, é fundamental para aplicações KDD porque representa a preocupação em validar os modelos de conhecimento que usuários finais freqüentemente idealizam. Modelos de conhecimentos são dados organizados sob a perspectiva, por exemplo, de um produto vendido ou um diagnóstico médico conhecido, que dependem do domínio de aplicação, mas todos mostram os fatos históricos que apóiam investimentos futuros nas empresas.

Figura 3.8 – Diagrama de pacote da Definição de Trabalho - Compreender o Contexto de Dados

Algumas vezes, as expectativas dos tomadores de decisão não são cabíveis para a implementação, mesmo em aplicações que têm como recursos as tecnologias analíticas como DW, mineração de dados ou OLAP. Portanto, compreender os dados existentes antes de construir uma aplicação KDD traz coesão ao desenvolvimento.

Por isso, a atividade Compreender o Contexto de Dados realiza a ação de construir o artefato Descrição da Base de Dados Existente, que é o resumo da estrutura de dados

existente, tendo por propósito entender os dados do negócio, por exemplo, destacando as principais tabelas e campos da base de dados. Este artefato tem como modelo o Quadro 3.4. As Figuras 3.9, 3.10 e 3.11 mostram os aspectos dinâmicos e estáticos desta atividade.

Template Descrição da Base de Dados Existente **Descrição:** **{resumir por texto descritivo, a relação e a estrutura de dados existentes. Destacando as principais tabelas e campos}**

Quadro 3.4 – *Template* Descrição da Base de Dados Existente

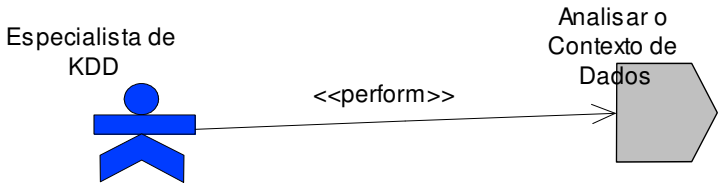


Figura 3.9 – Diagrama de caso de uso da Definição de Trabalho - Compreender o Contexto de Dados

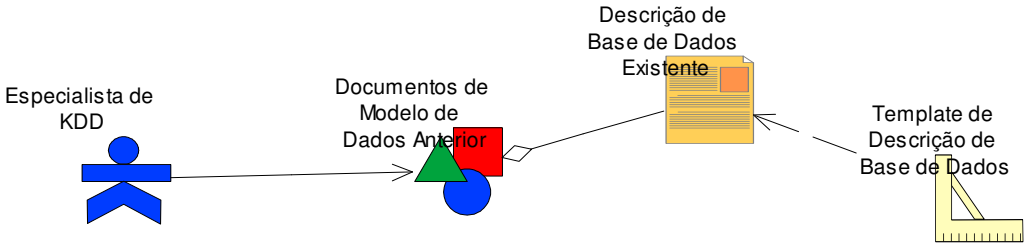


Figura 3.10 – Diagrama de classes da Definição de Trabalho - Compreender o Contexto de Dados

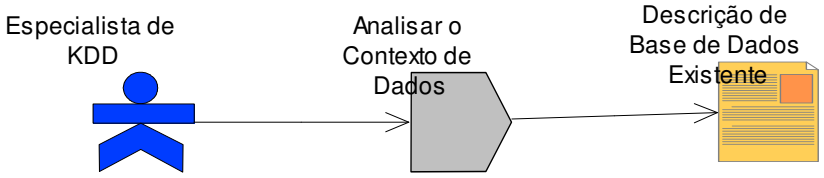


Figura 3.11 – Diagrama de atividade da Definição de Trabalho - Compreender o Contexto de Dados

3.2.3 Definição de Trabalho – Compreender o Contexto de Ferramentas

Segundo Kimball e Ross (2002), na construção de projetos de DW existem considerações rigorosas tanto à elaboração do modelo dimensional e especificação da aplicação analítica quanto ao projeto técnico da arquitetura. A chamada trilha da tecnologia do ciclo de vida de projetos de DW indica uma atividade de seleção e instalação de produtos

adequados ao contexto da aplicação KDD. Muitas vezes, faz-se necessário elaborar uma matriz para avaliação das ferramentas existentes, a partir de critérios como infra-estrutura, viabilidade de aquisição e fornecimento. Conforme observação do autor, o custo relacionado com um projeto e implantação de um DW são significativos e em alguns casos já existem ferramentas configuráveis o suficiente para situações mais comuns, desestimulando o desenvolvimento de uma aplicação KDD completa. A Figura 3.12 apresenta o diagrama de pacotes que explica o que é necessário para compreender o contexto de ferramentas existentes.

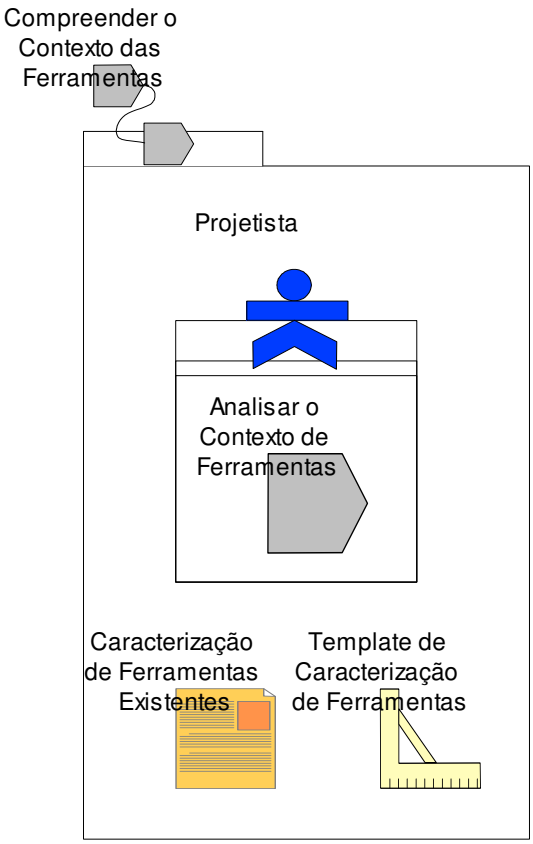


Figura 3.12 – Diagrama de pacote da Definição de Trabalho - Compreender o Contexto de Ferramentas

O artefato Caracterização das Ferramentas Existentes é um resumo das características dos softwares KDD, enfatizando as capacidades e limitações operacionais, como também os requisitos mínimos para instalação, operação e manutenção. O modelo para este artefato está definido no Quadro 3.5.

Template Caracterização de Ferramentas Existentes
Nome da Ferramenta:
{nome da ferramenta}
Fabricante da Ferramenta:
{nome do fabricante e formas de distribuição da ferramenta}
Capacidades Operacionais da Ferramenta:
{funcionalidades oferecidas}
Limitações Operacionais da Ferramenta:
{ funcionalidades não oferecidas }
Recursos para instalação da Ferramenta:
{recursos incluem hardware, software, custo, treinamento, manual}

Quadro 3.5 – *Template* Caracterização de Ferramentas Existentes

Portanto, a atividade de Compreender o Contexto de Ferramentas consiste na realização da ação de caracterizar limitações e capacidades dos softwares, com o intuito de orientar e restringir as atividades dos desenvolvedores somente para as funcionalidades não atendidas pelos softwares prontos. Isto envolve, principalmente, a comunidade de projetista da aplicação KDD. As Figuras 3.13, 3.14 e 3.15 mostram interação entre os elementos do modelo para esta atividade.

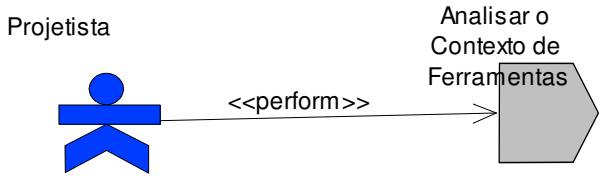


Figura 3.13 – Diagrama de caso de uso da Definição de Trabalho - Compreender o Contexto de Ferramentas

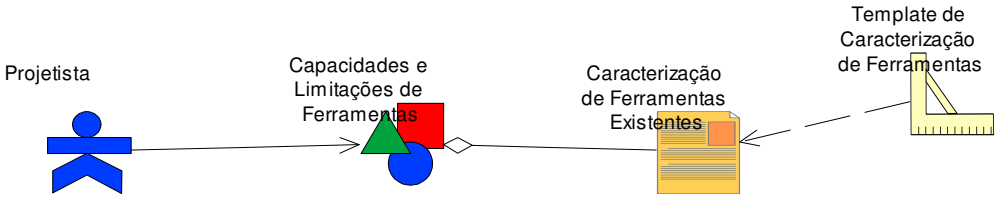


Figura 3.14 – Diagrama de classes da Definição de Trabalho - Compreender o Contexto de Ferramentas

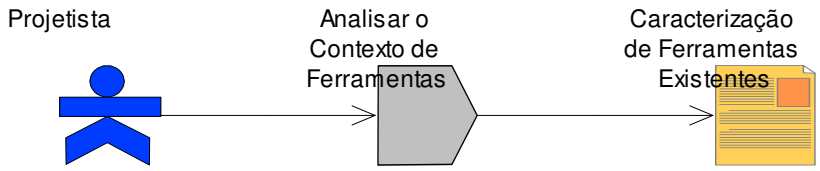


Figura 3.15 – Diagrama de atividade da Definição de Trabalho - Compreender o Contexto de Ferramentas

Por fim, a Disciplina de Requisitos fornece um conjunto de informações que funcionam como pilares de sustentação para as aplicações KDD. A partir dos artefatos gerados em Requisitos, torna-se possível mapear as expectativas dos usuários para a base de dados existente e entrever os dados existentes em uma perspectiva multidimensional.

3.3. DISCIPLINA ANÁLISE

Durante o desenvolvimento de uma aplicação KDD, entende-se por análise as atividades que auxiliarão o desenvolvedor desvendar a natureza subjetiva e complexa deste tido de aplicação. Esta é rica em fatores indeterminísticos como situação em que se encontram os dados, instabilidade ou ilusão quanto às expectativas dos usuários finais e não formalismo das mudanças em geral. Esta Disciplina divide-se semanticamente em duas Definições de Trabalho, conforme mostrado na Figura 3.16. A seqüência de Definições de Trabalhos da Disciplina de Análise é Definir a Estrutura da Decisão e Compreender Arquitetura Dimensional.

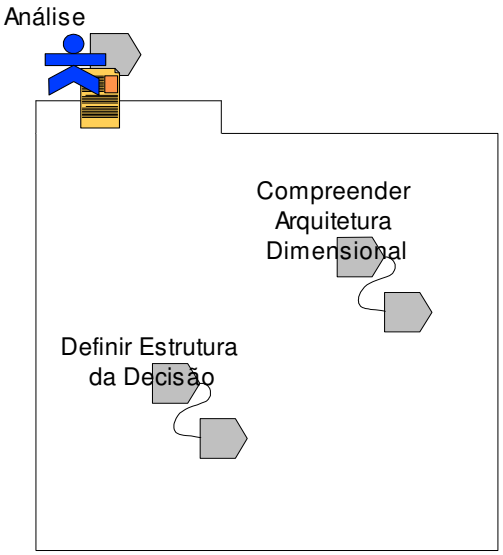


Figura 3.16 – Diagrama de pacote da Disciplina Análise

3.3.1 Definição de Trabalho – Definir Estrutura da Decisão

Conforme Kimball et al. (1998), um projeto de DW requer a investigação nos dados da organização, identificando quais informações do negócio são necessárias para apoiar a tomada de decisões. Neste sentido, a tarefa do Engenheiro de Conhecimento, mostrada na Figura 3.17 é validar as questões hipotéticas de busca no banco de dados, reunidas na Lista de Inferência, com a finalidade de verificar a possibilidade de as mesmas existirem e identificar tabelas e campos imprescindíveis para cada questão. Assim, uma suposição de conhecimento que se espera encontrar na base de dados existente, é transformada na Lista de Informações de Negócio, que representa o possível mapeamento necessário para alcançar o conhecimento esperado.

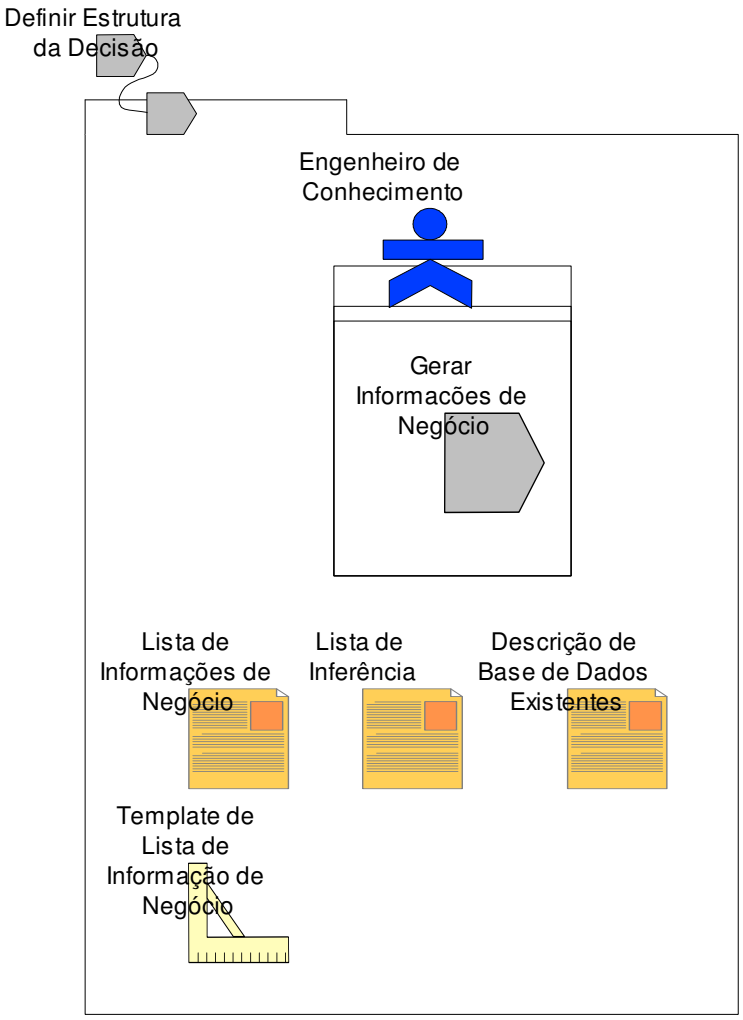


Figura 3.17 – Diagrama de pacote da Definição de Trabalho - Definir Estrutura da Decisão

Para representar e orientar o desenvolvedor, indica-se o modelo de Lista de Informações de Negócio, mostrado no Quadro 3.6, a seguir. Nele é possível identificar desde a motivação do usuário final quanto às suas expectativas de conhecimento, em nível alto de abstração, até o nível mais detalhado das oportunidades de busca na base de dados, como as tabelas e campos.

Template Lista de Informações de Negócio

Necessidade ou Motivação do Tomador de Decisão:

{representar em uma frase o motivo de investigação no banco de dados}

Inferências Propostas:

{a partir da Lista de Inferência já elaborada, escolher a(s) pergunta(s) que melhor representam a intenção de busca no banco de dados}

Informações de Negócio:

{a partir da Descrição da Base de Dados Existente, listar a informações necessárias que estejam relacionadas à resposta da inferência investigada}

Quadro 3.6 – *Template* Lista de Informações de Negócio

A atividade de Gerar Informações de Negócio conta com a ajuda do Especialista de KDD, contribuindo com a visão dos dados em possíveis fatos e dimensões a serem construídos posteriormente. Conta também com o entendimento da base de dados existente, identificando as novas medidas (variáveis) candidatas para o modelo dimensional. As Figuras 3.18, 3.19 e 3.20 resumem esta atividade.

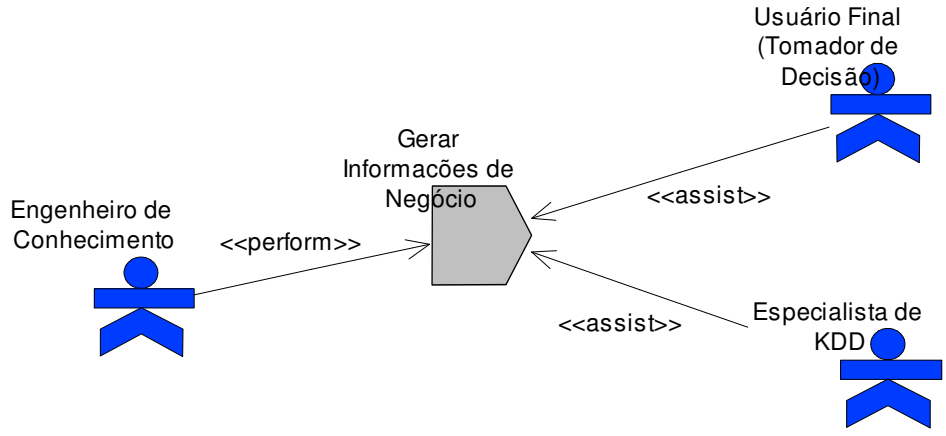


Figura 3.18 – Diagrama de caso de uso da Definição de Trabalho - Definir Estrutura da Decisão

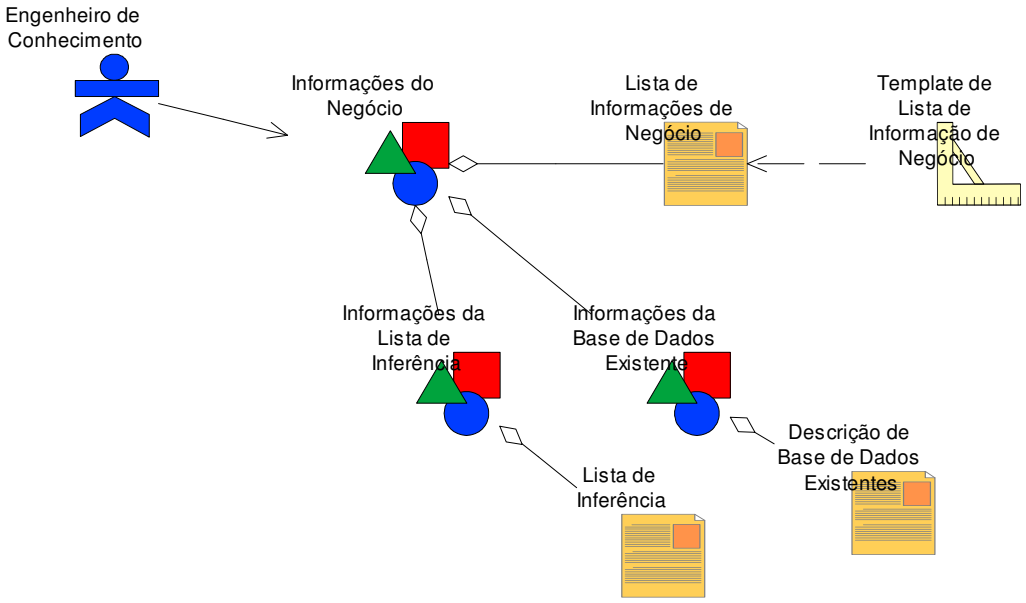


Figura 3.19 – Diagrama de classes da Definição de Trabalho - Definir Estrutura da Decisão

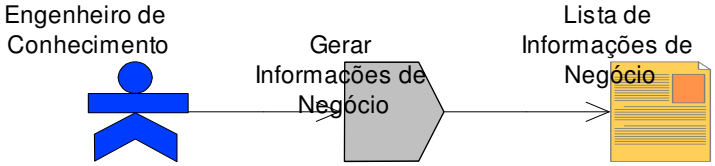


Figura 3.20 – Diagrama de atividade da Definição de Trabalho - Definir Estrutura da Decisão

3.3.2 Definição de Trabalho – Compreender Arquitetura Dimensional

A motivação por Compreender a Arquitetura Dimensional é dada pelas diretrizes de Kimball et al. (1998). O autor propõe uma completa estrutura arquitetural para apoiar os projetos de DW. Nesta o nível de detalhamento se inicia em requisitos de negócio, passando por modelos da arquitetura e finalizando na implementação do DW. Sendo que cada grau de detalhe percorre por três perspectivas importantes que são a de dados, técnica e infra-estrutura.

Em vista disso, o artefato Matriz de Barramento faz parte da arquitetura para projetos de DW, possibilitando a visualização dos relacionamentos existentes dos campos de interesse para a tomada de decisão. A Figura 3.21 resume os componentes desta Definição de Trabalho.

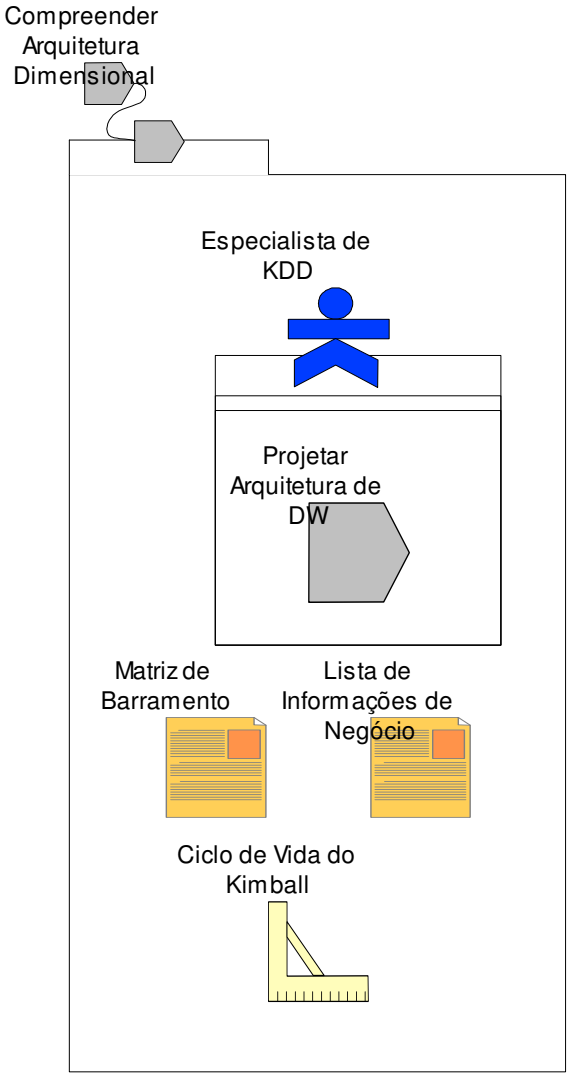


Figura 3.21 – Diagrama de pacote da Definição de Trabalho - Compreender Arquitetura Dimensional

O modelo do artefato Matriz de Barramento segue o modelo proposto por Kimball e Ross (2002), mostrando nas linhas os possíveis *data marts*[17] e nas colunas as possíveis dimensões comuns usadas na empresa. Fato é uma medida de desempenho de negócio. Dimensão é uma entidade independente que serve como ponto de entrada das medidas localizadas nas tabelas de fatos.

Em suma, o artefato Matriz de Barramento é independente de tecnologia e plataforma e é também um mecanismo para documentar, criar e comunicar arquitetura de projeto de DW. O Quadro 3.7 apresenta um modelo deste artefato no contexto do processo de estoque e a cadeia de valores relacionados.

[17] *Data marts* implementam um determinado assunto, enquanto um DW implementa e integra vários assuntos da organização.

Template Matriz de Barramento

PROCESSOS DE NEGÓCIO	DIMENSÕES COMUNS							
	Data	Produto	Loja	Promoção	Warehouse	Fornecedor	Contrato	Transportadora
Vendas no varejo	X	X	X	X				
Estoque no varejo	X	X	X					
Entregas no varejo	X	X	X					
Estoque do warehouse	X	X			X	X		
Entregas no warehouse	X	X			X	X		
Ordens de compra	X	X			X	X	X	X

Fonte: Kimball e Ross (2002, p.92)
Quadro 3.7 – *Template* Matriz de Barramento

A atividade Projetar Arquitetura de DW é explicada nas Figuras 3.22, 3.23 e 3.24.

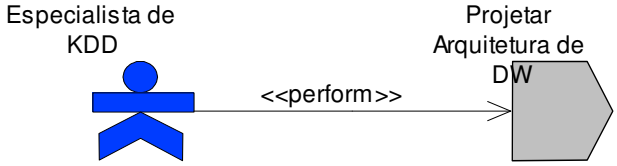


Figura 3.22 – Diagrama de caso de uso da Definição de Trabalho - Compreender Arquitetura Dimensional

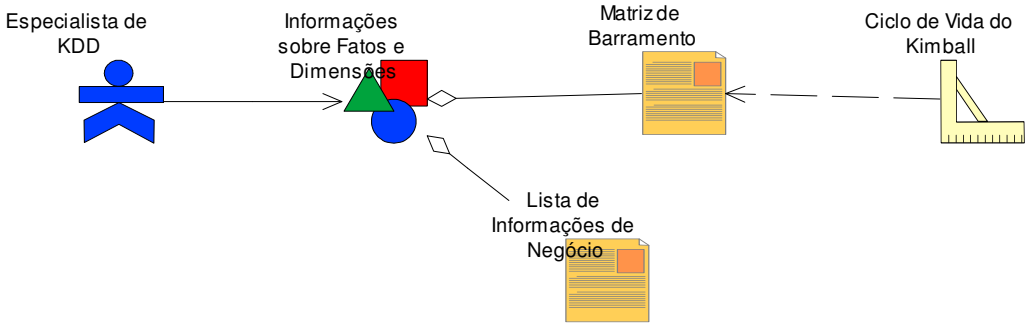


Figura 3.23 – Diagrama de classes da Definição de Trabalho - Compreender Arquitetura Dimensional

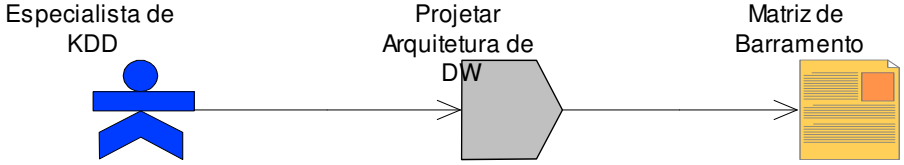


Figura 3.24 – Diagrama de atividade da Definição de Trabalho - Compreender Arquitetura Dimensional

Vale ressaltar que o objetivo deste trabalho de mestrado é propor um p*rocesso de software para aplicações KDD*, então neste contexto o fundamento e as metodologias da área de DW, Modelagem Dimensional, assim como Matriz de Barramento não são tratados em detalhes. O trabalho propõe mostrar a integração destas áreas de KDD à área de engenharia de software.

Em resumo, a Disciplina de Análise especifica as informações necessárias para o projeto de aplicações KDD, diminuindo a complexidade das informações cada vez mais. Por meio dos artefatos gerados em Análise, é possível listar os pontos mais instáveis do projeto de DW e apontar estratégias para a transformação dos dados transacionais em modelos dimensionais.

3.4. DISCIPLINA PROJETO

Durante o desenvolvimento de uma aplicação KDD, entende-se por projeto a concepção do modelo de dados dimensional, envolvendo ferramentas para a construção do projeto arquitetural de DW. A Figura 3.25 mostra a Definição de Trabalho que determina a elaboração do modelo dimensional.

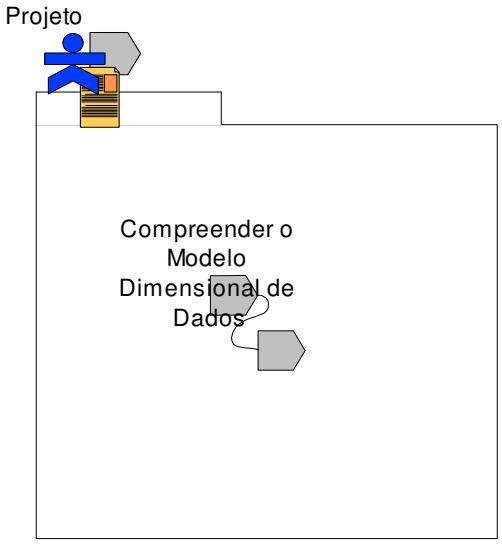


Figura 3.25 – Diagrama de pacote da Disciplina Projeto

3.4.1 Definição de Trabalho – Compreender o Modelo Dimensional de Dados

Conforme as diretrizes do ciclo de vida de um projeto de DW, do autor Kimball e Ross (2002), a modelagem de dados deve ser realizada por alguém que possua experiência em modelagem de dados transacionais com grande ênfase em normalização. A Figura 3.26 mostra o relacionamento deste papel no processo, aqui denominado como Especialista de KDD, com os outros elementos do modelo.

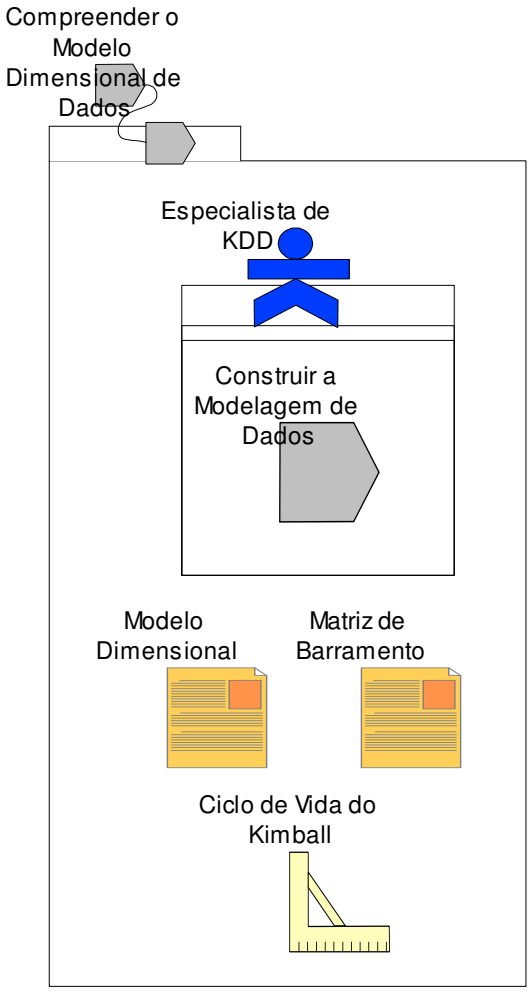
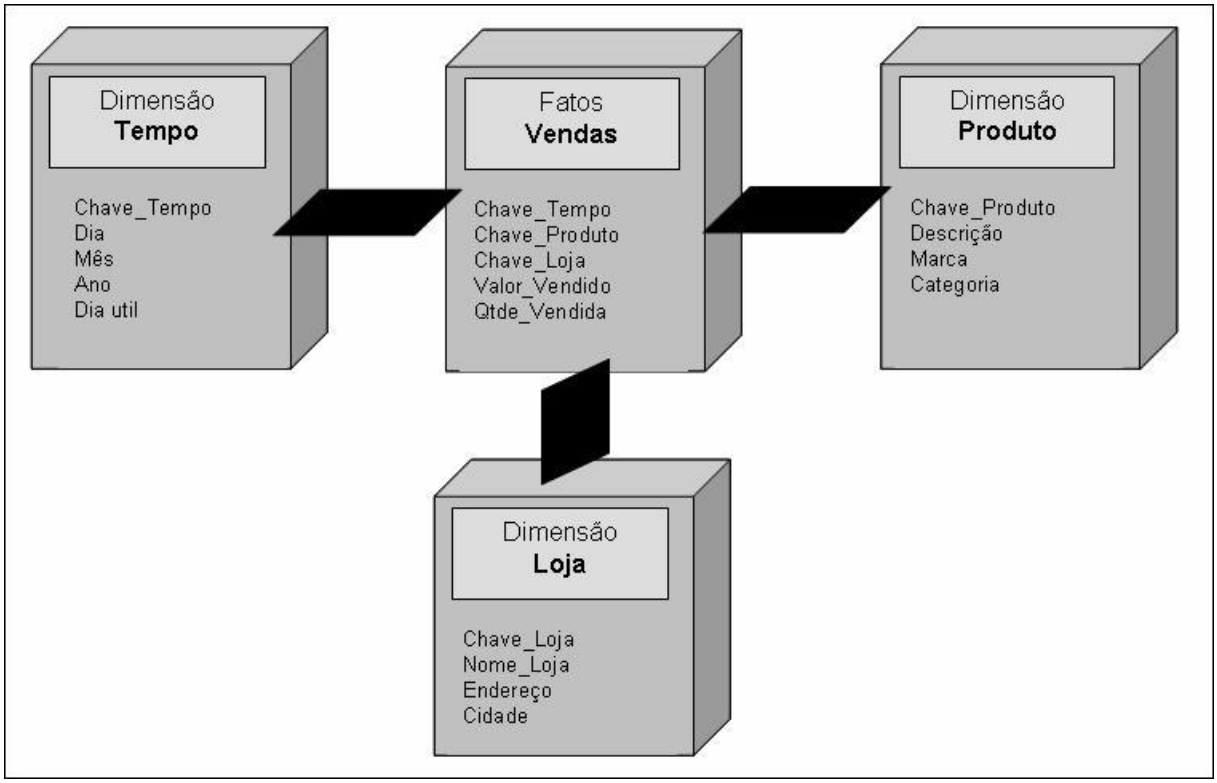


Figura 3.26 – Diagrama de pacote da Definição de Trabalho - Compreender o Modelo Dimensional de Dados

O modelo de dados dimensional é um conjunto de medidas que descreve aspectos comuns de negócio, servindo para sumarizar e reestruturar dados, mostrando-os em visões que auxiliem a tarefa de análise de valores. Este é composto por fatos, dimensões e medidas

(variáveis). Sendo fato um evento do negócio composto das medidas de determinado contexto, enquanto que dimensões não possuem atributos numéricos, pois são apenas descritivas e classificatórias dos elementos que participam de um fato. Por fim, medidas são representações dos indicadores relativos às dimensões que participam de um fato (Machado, 2004). No Quadro 3.8 é mostrado um exemplo do artefato Modelo Dimensional, no contexto de Vendas.

Template Modelo Dimensional



Fonte: adaptação do esquema preliminar do modelo dimensional, Kimball e Ross (2002, p.43)
Quadro 3.8 – *Template* Modelo Dimensional

As Figuras 3.27, 3.28 e 3.29 mostram os aspectos dinâmicos e estáticos da atividade denominada Construir a Modelagem de Dados. Observa-se o artefato Matriz de Barramento como recurso a esta atividade.

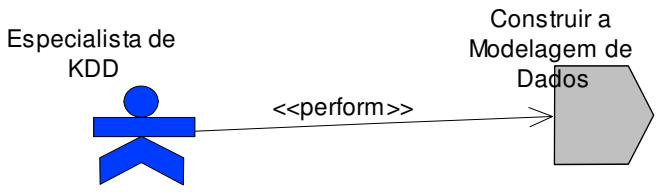


Figura 3.27 – Diagrama de caso de uso da Definição de Trabalho – Construir a Modelagem de Dados

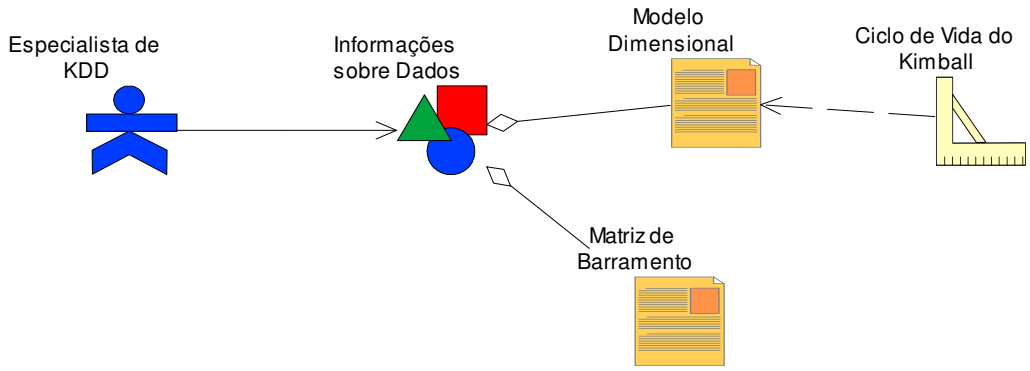


Figura 3.28 – Diagrama de classes da Definição de Trabalho - Construir a Modelagem de Dados

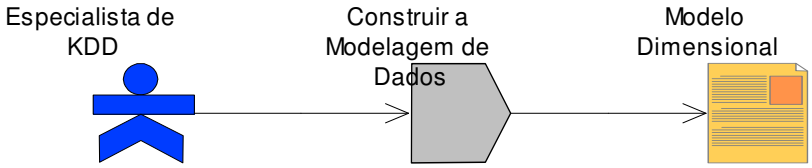


Figura 3.29 – Diagrama de atividade da Definição de Trabalho - Construir a Modelagem de Dados

Em suma, a Disciplina de Projeto preocupa-se em realizar o projeto lógico e físico dos dados para aplicações KDD, estimando esforços da implantação e mudança no DW. Desse modo, o artefato gerado exige que todos os artefatos anteriormente listados nas Seções 3.2 e 3.3, respectivamente, tenham sido construídos, pois estes servem de apoio a decisões do projeto da arquitetura e implementação do DW.

3.5. FASE CONCEPÇÃO

Como o objetivo da fase de concepção no UP é estabelecer a viabilidade do sistema, construindo arquiteturas candidatas para implementação, o UPKDD incorpora a esta fase artefatos importantes para aplicações KDD, que poderiam impactar o desenvolvimento.

A Figura 3.30 reúne os *workflows* de requisitos, análise e projeto, destacando os papéis relevantes que o desenvolvedor de aplicações KDD possui. Neste diagrama de classes é mostrada a interação do Papel do Processo “Especialista de KDD” com o papel de “Analista de Sistemas”, por meio do esteriótipo <<*assistant*>>, que demonstra a importância do entendimento do contexto do sistema na concepção de aplicações KDD, sumarizadas na Lista de Inferências.

Este diagrama tem por propósito mostrar a visão abrangente da fase inicial dos projetos e não a visão detalhista de qualquer *workflow* tratado pelo processo proposto.

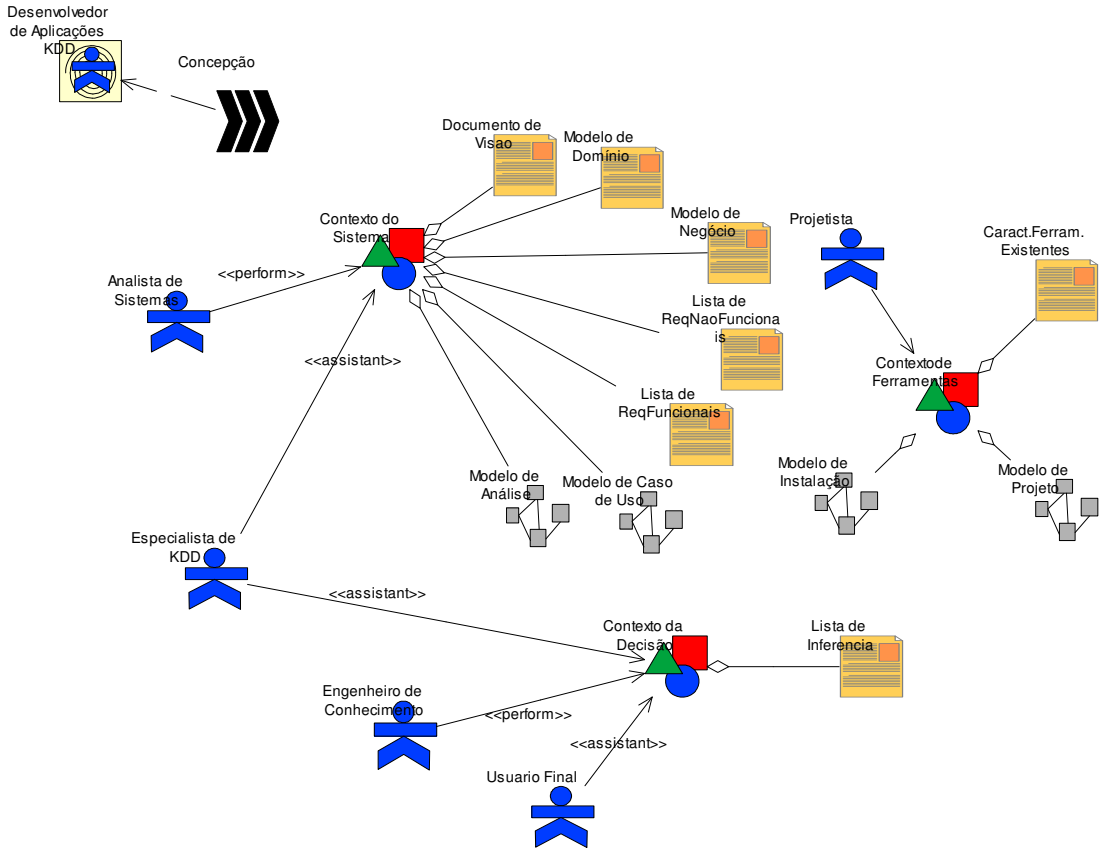


Figura 3.30 – Diagrama de classes da Fase Concepção

3.6. FASE ELABORAÇÃO

Como o objetivo da fase de elaboração no UP é estabelecer uma base arquitetônica sólida, por meio de requisitos funcionais descritos anteriormente, o UPKDD incorpora os modelos e documentos detalhados da arquitetura de DW, com o intuito de especificar tecnologias envolvidas assim como perspectivas sobre os componentes técnicos da aplicação KDD.

A Figura 3.31 reúne os *workflows* de requisitos, análise, projeto e implementação, destacando os artefatos e a dependência destes aos papéis identificados. Apesar do *workflow* de teste não ser tratado de maneira gráfica neste diagrama, a indicação para aplicações KDD é incorporar ao planejamento e casos de testes, requisitos funcionais relativos à descoberta de

conhecimento. Acredita-se que o teste de caixa-preta seja o mais adequado para este tipo de aplicação.

Este diagrama tem por propósito mostrar a integração dos papéis e artefatos do UPKDD, sendo construídos por diretrizes especificadas pelas Orientações como UML 2.0 e Ciclo de Vida de Projeto de DW. A criação do diagrama referente à fase de elaboração serve para mostrar o caminho evolutivo tratado pelo processo proposto.

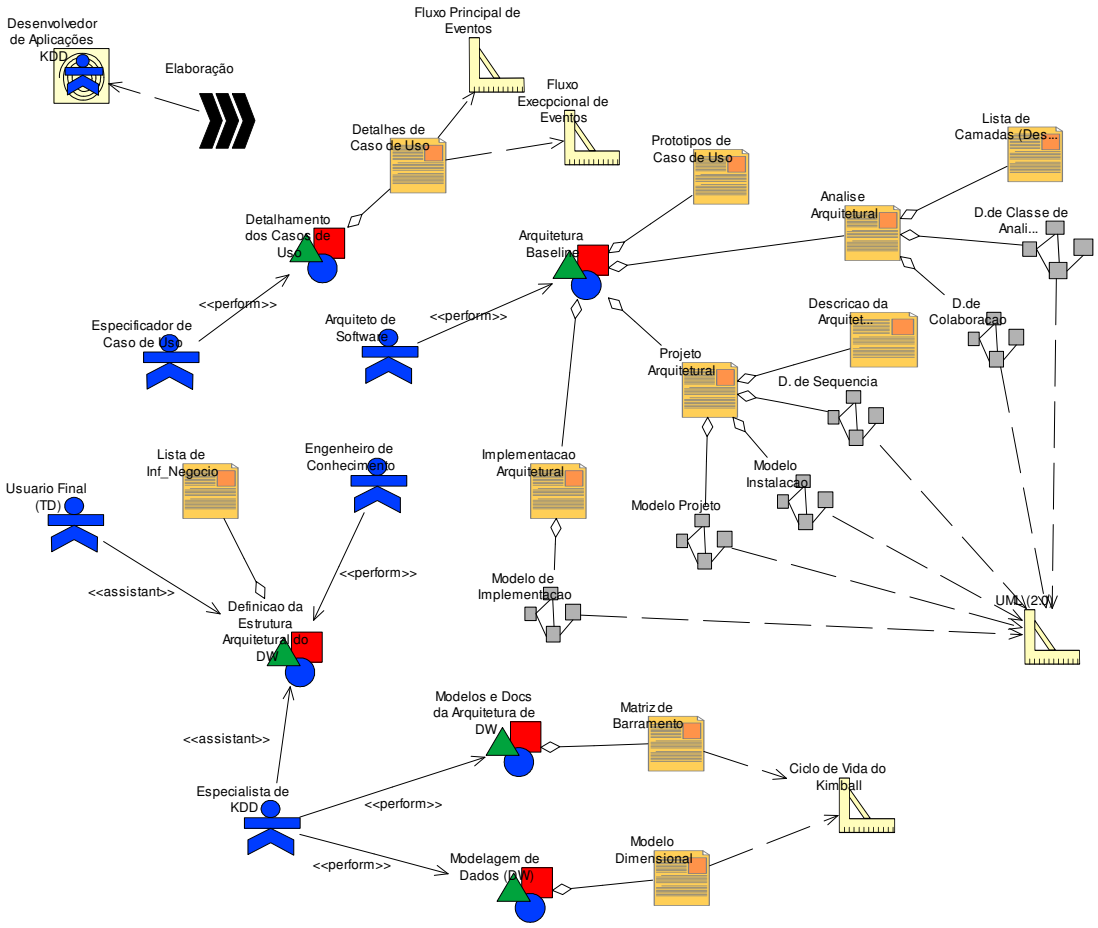


Figura 3.31 – Diagrama de classes da Fase Elaboração

3.7. CONSIDERAÇÕES FINAIS

Para a modelagem do processo optou-se pelo metamodelo SPEM, ainda que no início não houvesse ferramentas que apoiassem as atividades de modelagem usando os estereótipos

do metamodelo. Esta foi a motivação para a configuração realizada na ferramenta Rational Rose, para que a mesma instanciasse a notação SPEM para a modelagem do processo.

Os artefatos idealizados para este processo têm como base o Conjunto Comum (Apêndice A), pois nele constam as principais influências na área de KDD.

A condução das disciplina, assim como a divisão semântica de todo o processo, foi baseada nas diretrizes do UP. A partir da estrutura de processo proposta é possível gerar ou modificar elementos-chave para aplicações KDD. As mudanças aconteceram em requisitos que incorporam artefatos para a condução do processo KDD e também ocorreram em análise e projeto que concentram a visão da estrutura de dados multidimensionais.

Nota-se que tanto papéis quanto artefatos e atividades representados pelos diagramas descritos neste capítulo delimitam o trabalho a ser realizado, permitindo visualizar mecanismo de controle de projeto e produtividade para a equipe de desenvolvedores de aplicações KDD.

Este fato confere ao processo UPKDD a adequação ao desenvolvimento de aplicações KDD, especialmente aquelas cujos modelos de conhecimentos são elaborados a partir da presença do usuário.

No próximo capítulo, o processo UPKDD passa por avaliação usando as técnicas como o GQM e estudo de caso quantitativo.

4 AVALIAÇÃO DO PROCESSO PROPOSTO

Este trabalho de mestrado foi utilizado o método de avaliação experimental, em forma de *estudo de caso*, com o objetivo de representar se as mudanças propostas na utilização do processo UPKDD auxiliam no desenvolvimento de soluções KDD, observando os grupos de trabalho envolvidos e usando análise descritiva e dedutiva para interpretação dos resultados. Assim, neste capítulo são descritos o estudo de caso realizado juntamente com os resultados atingidos, e apresentadas as considerações finais quanto aos resultados obtidos na utilização do UPKDD.

4.1. CARACTERIZAÇÃO DO ESTUDO DE CASO

Para efeitos de instanciação de um processo modelado, optou-se por realizar um estudo de caso. Nele participaram dois grupos, com seis desenvolvedores cada, sendo que o grupo 1 recebeu treinamento do processo modelado e seguiu todos os elementos-chave propostos, enquanto o grupo 2 não recebeu este tipo de treinamento, porém naturalmente construiu os elementos-chave similares ao do processo modelado, mostrando a necessidade inerente do desenvolvedor de aplicações KDD por artefatos estabelecidos pelo UPKDD.

No estudo de caso, conduzido neste trabalho de mestrado, foram usadas duas estratégias para o desenvolvimento de uma mesma solução KDD. Em uma delas, um dos grupos de participantes deveria apresentar a solução KDD apenas recorrendo a ferramentas prontas no mercado como OWB (*Oracle Warehouse Builder*)[18], *Discoverer*[19] e *Weka*[20], resumidas no Apêndice III . Na outra estratégia, exigiu-se do outro grupo de participantes a especificação, análise e projeto de um software encomendado, que abordassem as mesmas funcionalidades das ferramentas prontas. Em suma, observou-se que os artefatos idealizados pelo UPKDD serviram de apoio aos dois grupos de participantes e, principalmente, o artefato Caracterização de Ferramentas Existentes fundamentou o software encomendado e qualificou as ferramentas escolhidas para o desenvolvimento da solução KDD.

[18] http://www.oracle.com/solutions/business_intelligence/warehouse-builder.html
[19] http://www.oracle.com/solutions/business_intelligence/olap.html
[20] http://www.cs.waikato.ac.nz/ml/weka/

Todos os artefatos gerados pelos grupos de participantes do estudo de caso estão relatados nos anexos a seguir:

a) Lista de Inferência (Anexo I);

b) Descrição da Base de Dados Existente (Anexo II);

c) Caracterização de Ferramentas Existentes (Anexo III);

d) Lista de Informações de Negócio (Anexo IV);

e) Matriz de Barramento (Anexo V);

f) Modelo Dimensional (Anexo VI).

4.2. TERMINOLOGIA EXPERIMENTAL

A terminologia associada ao experimento torna-o mais claro e legível, tanto para os participantes da experimentação quanto para compor o modelo de estudo de caso. No Quadro 4.1 os principais termos usados em experimentos estão nas duas primeiras colunas como termos e respectiva definição. Já na terceira coluna consta, para cada termo, a instanciação para este trabalho.

Termos	Definição	Instanciação
Hipótese Experimental	É o que o experimento pretendente testar. É a especificação quantificável dos requisitos do estudo de caso. Ela deve ser testada.	O conjunto de Elementos Chave (ECs) usados é diferente dos ECs do Conjunto Comum (Apêndice A).
Tratamento	É normalmente um método ou ferramenta que se deseja avaliar. Uma hipótese experimental normalmente afirma que diferentes tratamentos têm diferentes efeitos nos sujeitos e objetos experimentais.	Processo UPKDD
Controle	É o método ou ferramenta que atualmente se usa, que deverá ser comparado com o método ou ferramenta alternativa. O uso de um novo método ou ferramenta é então comparado com o controle.	Conjunto Comum
Variável de Resposta	É a medida que captura os efeitos da mudança do método ou ferramenta alternativa, por ex., produtividade do desenvolvedor e qualidade do produto. São fatores que se espera serem diferentes dos atuais com a mudança. Como um resultado da aplicação do tratamento.	ECs usados
Sujeito experimental	São as pessoas, grupos ou indivíduos envolvidos em um experimento, aos quais são aplicados os tratamentos.	Grupo 1 Grupo 2
Objeto experimental	São coisas envolvidas em um experimento. São "No Que" o tratamento será aplicado (programas, especificações, módulos).	ECs do UPKDD
Variável de estado	São medidas usadas para descrever os sujeitos, objetos e condições. Elas capturam fatos que afetam as variáveis de resposta. São fatores que caracterizam o estudo de caso e podem influenciar na avaliação do resultado (área de aplicação, tipo de sistema).	Utilização Utilidade Adequação do nível de descrição
Baseline	São informações necessárias e suficientes disponíveis para a comparação que será feita pelo experimento, fornecida pelo controle.	Não se aplica neste estudo
Projeto piloto	É o projeto que tem por objetivo a entrega de um produto de software. São projetos usados pelos estudos de caso.	Solução KDD para o Laboratório de Exames Clínicos.

Quadro 4.1 – Definição e instanciação de termos experimentais

Para a realização deste trabalho de mestrado, foi escolhido o método de avaliação *empírico*, em forma de *estudo de caso quantitativo*, por melhor se adequar às condições dos pesquisadores envolvidos e também pelas condições dos dados da pesquisa. Nas próximas seções é apresentada a seqüência dos passos realizados, tomando como base o modelo de estudo de caso conduzido por Travassos, Gurov e Amaral (2002).

4.3. ESTUDO EXPERIMENTAL – ESTUDO DE CASO QUANTITATIVO

Conforme natureza deste trabalho de mestrado, que visa propor mudanças na forma de desenvolvimento de soluções KDD, existe a necessidade de mensurar o processo proposto por meio de um método de avaliação. O método de avaliação escolhido foi o estudo de caso, já que se trata de um estudo comparativo que não exige repetições de projetos, como acontece com os experimentos formais, e consome baixo custo de operação. Este estudo de caso é quantitativo, pois no decorrer da experimentação os efeitos das mudanças propostas podem e são mensuráveis.

Este estudo empírico foi realizado em aproximadamente cinco meses e passou por quatro fases, conforme *framework* experimental de Basili, Selby e Hutchens (1985) e a abordagem GQM, proposta pelo mesmo autor, para o estabelecimento de objetivos mensuráveis, a saber: Definição, Planejamento, Operação e Interpretação.

4.3.1. Fase Definição

Na fase de definição são preparados os conceitos da medição como: os objetivos, as questões e as métricas para o experimento.

1) Objetivo global

Avaliar se o UPKDD oferece Elementos-Chave (ECs) necessários para o desenvolvimento de aplicações de descoberta de conhecimento em banco de dados (KDD), do ponto de vista de seus desenvolvedores.

2) Objetivo da medição

Tendo como base o Conjunto Comum de ECs que reúne os principais componentes representativos do processo KDD (Apêndice A), caracterizar:

Objetivo 1 - Quais são os ECs do UPKDD, que são usados:

Então Mensurar

Quais são os ECs do UPKDD que são considerados úteis para o desenvolvimento de aplicações KDD;

Quais são os ECs do UPKDD que são considerados inúteis para o desenvolvimento de aplicações KDD;

Objetivo 2 - Quais são os ECs do UPKDD que são usados e possuem descrição inadequada (quanto ao conteúdo descritivo);

Então Mensurar

Quais são os ECs do UPKDD que necessitam melhor descrição, podendo o texto descritivo ser aumentado ou diminuído;

Quais são os ECs do UPKDD que apresentam descrição suficiente, não necessitando de modificações quanto ao conteúdo descritivo;

Objetivo 3 - Quais são os ECs que os desenvolvedores gostariam de usar, mas não usaram.

3) Abordagem GQM - Meta (Objetivo)/Questões/Métricas

Na abordagem GQM, a definição de Metas (Objetivos) pode ser apresentada da seguinte forma:

Analisar <*Objeto do estudo*>

Com o propósito de <*Objetivo*>

Com respeito à <*Foco da qualidade*>

Do ponto de vista <*Perspectiva*>

No contexto de <*Contexto*>

Analisar o *Conjunto de ECs que são usados pelos desenvolvedores do UPKDD*

Com o propósito de *caracterizar*

Com respeito à interseção *com os ECs do Conjunto Comum*

Do ponto de vista *do desenvolvedor de aplicações KDD*

No contexto de *solução de KDD*

(Q1) Questão1: Existem ECs do Conjunto Comum que não fazem parte do Conjunto de ECs usados do UPKDD?

(M1) Métrica: A lista de ECs do Conjunto Comum que não fazem parte do Conjunto de ECs usados do UPKDD.

(Q2) Questão2: Existem ECs do Conjunto Comum e ECs usados do UPKDD que são considerados inúteis pelos desenvolvedores?

(M2) Métrica: A lista ECs do Conjunto Comum que fazem parte dos ECs usados do UPKDD e são considerados inúteis pelos desenvolvedores.

(Q3) Questão3: Existem ECs do Conjunto Comum e ECs usados do UPKDD considerados úteis pelos desenvolvedores, cuja descrição deve ser modificada?

(M3) Métrica: A lista dos ECs do Conjunto Comum que fazem parte dos ECs usados do UPKDD e considerados úteis pelos desenvolvedores, cuja descrição deve ser modificada.

(Q4) Questão4: Existem ECs do Conjunto Comum que não fazem parte do conjunto de ECs usados do UPKDD, mas que os desenvolvedores gostariam de usar porque consideram úteis para o desenvolvimento de soluções KDD?

(M4) Métrica: A lista dos ECs do Conjunto Comum que não fazem parte do conjunto de ECs usados do UPKDD.

4.3.2. Fase Planejamento

Esta fase é responsável pela elaboração do experimento, nela as hipóteses são formuladas, existe a seleção de variáveis, seleção dos participantes, preparação conceitual da instrumentação e considerações para a validade do experimento.

1) Definição das hipóteses

Hipótese Nula (H0):

Os ECs usados pelos desenvolvedores do UPKDD são diferentes dos ECs considerados no Conjunto Comum como sendo fundamental para o desenvolvimento de soluções KDD.

Hipótese Alternativa (H1):

Os ECs usados pelos desenvolvedores do UPKDD são similares aos ECs considerados no Conjunto Comum como sendo fundamental para o desenvolvimento de soluções KDD.

Hipótese Alternativa (H2):

No conjunto de ECs usados do UPKDD e que fazem parte do Conjunto Comum existem ECs que os desenvolvedores consideram úteis para o desenvolvimento de soluções KDD.

Hipótese Alternativa (H3):

No conjunto de ECs usados do UPKDD, que fazem parte do Conjunto Comum e considerados úteis para o desenvolvimento de soluções KDD, existem ECs cuja descrição deve ser modificada para atingir o nível esperado pelos desenvolvedores.

Hipótese Alternativa (H4):

No conjunto de ECs não usados do UPKDD existem ECs que os desenvolvedores gostariam de usar.

2) Descrição da instrumentação

Esta instrumentação servirá então para definir para cada EC, aplicando o teste estatístico qui-quadrado: (1) Se pode considerar que o EC é usado; (2) Se pode considerar que o EC é útil e (3) Se pode considerar que a descrição do EC não precisa de modificação.

Para cada EC fundamental para o desenvolvimento de soluções KDD, oferecer a avaliação conforme modelo do Quadro 4.2, a seguir:

Utilização significa a Aplicação do elemento-chave durante o desenvolvimento da solução KDD, considerando inclusive o conceito aplicado.	**Utilidade** significa que o elemento-chave em questão pode ser Eficaz, em situações específicas para a solução KDD.	**Adequação do nível de descrição** significa que determinado elemento-chave possui explicação Conveniente quanto a sua definição e uso.
Utilização (S)	**Utilidade (U)**	**Adequação do nível de descrição (A)**
1.não usado e não gostaria de usar. 2.não usado, mas gostaria de usar. 3.usado.	1. não se demonstrou útil. 2. provavelmente é útil. 3. é útil.	1.a descrição deve ser aumentada. 2.a descrição deve ser diminuída 3.a descrição não precisa ser modificada.

Quadro 4.2 – Modelo de instrumentação

O resultado obtido, conforme Tabela 4.1, será *n* ECs com valores diferentes para (S,U,A), sendo que:

S: Utilização receberá valores

{0 – não usado [respostas = 1 ou 2] /1 – usado [resposta = 3]}

U: Utilidade receberá valores

{0 – não é útil [resposta = 1]/ 1 – é útil [respostas = 2 ou 3] }

A: Adequação receberá valores

{0 – nível é adequado [resposta = 3]/1 – nível não é adequado [respostas = 1 ou 2]}

Tabela 4.1 – Métricas para os ECs

N	S	U	A	Descrição do EC	Questões
1	0	0	0	Não é usado, não é útil, a modificação não é necessária.	Q1, Q4
2	0	0	1	Não é usado, não é útil, a modificação é necessária.	N/A
3	0	1	0	Não é usado, é útil, a modificação não é necessária.	Q1, Q4
4	0	1	1	Não é usado, é útil, a modificação é necessária.	Q1, Q4
5	1	0	0	É usado, não é útil, a modificação não é necessária.	Q2
6	1	0	1	É usado, não é útil, a modificação é necessária.	Q2
7	1	1	0	É usado, é útil, a modificação não é necessária.	Q2
8	1	1	1	É usado, é útil, a modificação é necessária.	Q2

3) Seleção do contexto

O contexto pode ser caracterizado conforme quatro dimensões:

1. Processo {*on-line*/*off-line*}
2. Participantes {alunos/profissionais}
3. Realidade {o problema real/modelado}
4. Generalidade {específico/geral}

O processo é caracterizado como *on-line*, porque durante a execução do estudo de caso os participantes são observados e eventualmente entrevistados.

Os participantes são alunos. Este grupo é composto por 8 (oito) profissionais com formação em Tecnologia em Processamento de Dados e Bacharelado em Informática, a maioria com mais de 10 anos de experiência, 3 (três) profissionais com formação em Ciência da Computação com experiência em torno de 3 anos e um profissional de Engenharia de Produção com 1 ano de experiência em desenvolvimento de sistemas (Apêndice F).

O problema é real, ou seja, não simulado. O estudo conduz à identificação de ECs inseridos no processo de desenvolvimento de soluções KDD, de maneira significativa visto que depende da presença ou não das atividades previstas no processo.

O contexto possui caráter específico porque é focado em mecanismos que facilitem o trabalho do desenvolvedor de soluções KDD, representado pela investigação comparativa de ECs usados pelos desenvolvedores com o Conjunto Comum de ECs.

4) Seleção dos indivíduos

Os indivíduos participantes do estudo de caso são alunos matriculados na disciplina de Banco de Dados, do Programa de Pós-graduação em Ciência da Computação da UEM, tendo a maioria deles experiência, com nível médio, em desenvolvimento de software. Portanto a escolha dos participantes não foi de maneira aleatória, pois esta disciplina tem como um dos principais objetivos o ensino de tecnologias de *data warehousing*. Nesse sentido, foi solicitado aos alunos, por meio de estudo de caso, a construção de um *data warehouse* juntamente com a construção de aplicações analíticas, a partir de uma base de dados relacional de um sistema real, ou seja, não hipotético.

Como estratégia de comparação das atividades desempenhadas pelos alunos participantes, por meio da observação, os alunos foram divididos em dois grupos de maneira aleatória e foi definido que cada grupo usaria uma abordagem diferente de desenvolvimento de sistemas de descoberta de conhecimento em banco de dados. Uma das abordagens segue diretrizes ministradas por meio de treinamentos, que enfatizaram uso de documentos específicos, responsabilidades e atividades pré-determinadas. A abordagem alternativa procura não conduzir rigidamente o desenvolvedor, deixando-o livre para busca de soluções, documentos e responsabilidades conforme demanda.

5) Variáveis

Variável independente:

A lista de ECs do Conjunto Comum.

Variáveis dependentes: (Similaridade, Utilidade e Adequação)

1. Similaridade entre os ECs usados do UPKDD e os ECs do Conjunto Comum
   Pode receber os valores:
   Igual: quando todos os ECs têm o valor SUA = {1, X,X} (métricas 5-8);
   Diferente: quando todos os ECs têm o valor SUA = {0,X,X} (métricas 1-4);
   Similar: quando não se cumprem as condições de "Igual" e "Diferente". O grau de similaridade pode ser avaliado como: {1, X, X } / {0, X, X} + {1, X, X} *100%.
2. Utilidade de ECs similares. Mostra a parte útil dos ECs usados do UPKDD
   Parte Útil: {1,1,X} / {1,X,X} *100%;
   Parte Inútil: {1,0,X} / {1,X,X} *100%.
3. Adequação de ECs similares. Mostra a parte adequada dos ECs usados do UPKDD:
   Parte Adequada: {1,X,0} / {1,X,X}*100 %;
   Parte Inadequada: {1,X,1} / {1,X,X}*100%.

6) Análise qualitativa

Para analisar a informação referente aos ECs não usados pelos desenvolvedores, mas que eles gostariam de usar, se propôs aplicar a análise qualitativa. Essa análise deve apresentar o conjunto de ECs do Conjunto Comum que não são usados pelos desenvolvedores, mas que consideram necessários para o desenvolvimento de soluções KDD.

Esta análise depende dos ECs com valor SUA = {0,X,X} (métricas 1-4 da Tabela 4.1) e da opção "não usado, mas gostaria de usar" para a Utilização do EC.

7) Validade

A validade dos resultados depende da população de participantes do experimento e pode ser classificada em: interna, de conclusão, de construção e externa. A seguir esses tipos de validades são descritos.

a) Validade Interna - estabelece o relacionamento de causalidade e diferencia relacionamentos falsos.

Neste caso, os participantes foram selecionados a partir da matrícula na disciplina de banco de dados e também pelo nível de experiência em desenvolvimento de software que os mesmos possuem. Assim uma instrumentação adequada registra os perfis dos participantes para a aplicação do estudo de caso. A maioria dos participantes selecionados possui experiência em torno de 10 anos em desenvolvimento de sistemas. Portanto, assume-se que os mesmos são representativos para a população.

Para redução da influência de fatores externos como a persuasão de metodologias de desenvolvimento de software individuais aos participantes dos grupos, fez-se necessária a realização de oito horas de treinamento do processo proposto e nivelamento dos conceitos do domínio da aplicação, assim como a estratégia de divisão de reuniões dos grupos com horários diferenciados.

Os ECs foram definidos tendo como base o Conjunto Comum de Elementos-Chave para o desenvolvimento de soluções KDD. Este conjunto foi elaborado por meio de pesquisa sistemática em artigos e livros de autores que possuem forte influência tanto em tecnologias de *data warehousing*, quanto em aplicações analíticas como mineração de dados e OLAP.

A disciplina de Banco de Dados em que os participantes estão matriculados oferece subsídios para o desenvolvimento e uso de tecnologias de KDD. Portanto, assume-se que a condução dos trabalhos também foi representativa.

Para reduzir a influência de fatores externos e a falta de motivação para terminar o estudo, a investigação foi direcionada para a análise das tarefas desempenhadas por um grupo de participantes que seguiu o processo proposto, em comparação com qualquer outra seqüência de tarefas realizadas pelo outro grupo de participantes, ambas no mesmo contexto de soluções KDD.

b) Validade de Conclusão (Confiabilidade do Experimento) – demonstra que o estudo pode ser repetido com os mesmos resultados.

Para receber os valores de Utilização, Utilidade e Adequação do nível de descrição, o teste binomial foi utilizado. A verificação de hipótese foi feita por meio de simples demonstração de utilização ou não de ECs no conjunto que representa as variáveis independentes.

c) Validade de Construção - estabelece corretamente medidas operacionais para os conceitos a serem estudados.

Este estudo está caracterizado pela conformidade dos ECs do Conjunto Comum com os ECs usados pelos desenvolvedores do UPKDD. O Conjunto Comum representa um consenso dos autores sobre as tarefas, responsabilidades e documentos pertencentes ao contexto de aplicações KDD.

d) Validade Externa - estabelece o domínio no qual o estudo se encontra e que poderá ser generalizado.

Os participantes podem ser considerados representativos, já que os mesmos possuem experiência em desenvolvimento de sistemas. A instrumentação foi adequadamente elaborada, pois se compõe de ECs do Conjunto Comum. E, finalmente, as características temporais exigem tempo para aplicação dos instrumentos de avaliação, em torno de 40 minutos. Além disso, os participantes estão envolvidos na disciplina de banco de dados.

4.3.3. Fase Operação

Esta fase é interessada na coleta de dados de maneira imparcial.

1) Questionários aplicados

Os questionários foram aplicados em dois momentos diferentes. No início do estudo de caso, com o intuito de traçar o perfil do grupo de participantes. E depois no final do estudo de caso para caracterização das dificuldades encontradas e do processo utilizado, assim como elaboração da análise dos casos em que houve interesse em usar outra seqüência qualquer de atividades para o desenvolvimento da solução KDD.

Os questionários aplicados aos grupos de participantes do estudo de caso, realizado por este trabalho de mestrado, estão apresentados em apêndices, conforme relacionados a seguir:

- caracterização do participante (Apêndice C);
- caracterização do processo (Apêndice D);
- caracterização do estudo de caso (Apêndice E).

2) Resultado do estudo – dados crus

Os questionários foram aplicados e os dados tabulados de maneiras diferentes, a fim de facilitar a análise e interpretação dos dados, conforme pode ser visto no Apêndice F.

4.3.4. Fase Interpretação

Esta fase é responsável por explicar os resultados do experimento, usando para tal a estatística descritiva, a verificação de hipóteses por meio da aplicação do testes estatísticos (paramétricos ou não) e, também, considerando os aspectos de validação dos dados e resultados.

1) Validação dos dados

Houve apenas um participante com resposta incorreta do ponto de vista dos valores válidos de SUA (Utilização, Utilidade e Adequação). Foi o participante identificado como "9", a resposta inválida foi a "8" (Especialista KDD), por que o valor de SUA: {0,0,1} não deve existir para este contexto.

2) Estatística descritiva

Conforme Montgomery e Runger (2003), elaborar adequadamente a sumarização e apresentação de dados são fundamentais ao bom julgamento estatístico, permitindo focar em características relevantes a serem usadas na solução de problemas. Neste sentido, as medidas de tendência central, como a mediana e moda, organizam a amostra destacando os acontecimentos.

Mediana de uma amostra é uma medida de tendência central, que divide os dados em duas partes iguais, metade abaixo da mediana e metade acima. Se o número de observações for par, a mediana estará na metade da distância entre os dois valores centrais. Se o número de observações for ímpar, a mediana será o valor central. Moda da amostra é o valor da observação que ocorre com mais freqüência (Montegomery e Runger, 2003). Estas são medidas também chamadas de estimadores não tendenciosos de parâmetros.

Segundo Barros et al. (2006), a estatística descritiva acontece após a coleta de dados e sugere que a agregação da análise gráfica, junto às medidas de tendência central, apóia a visão geral do conjunto de dados. O autor indica quatro medidas estatísticas para este tipo de escala, a saber: Faixa, Correlação de Spearman, Mediana e Moda.

A medida Faixa indica a dispersão ou concentração de dados em relação ao valor central, por meio da diferença entre os maiores e os menores valores. Correlação de Spearman mostra a relação existente entre duas variáveis do conjunto de dados, identificando a posição relativa do elemento à ordenação de dados existentes. Para análise preliminar dos dados deste experimento, não é relevante o uso destas medidas estatísticas visto que não existe interesse em "rankear" o conjunto de dados e também em identificar a diferença entre os valores coletados. Neste experimento, as medidas estatísticas como Mediana e Moda são calculadas para os valores de Utilização, Utilidade e Adequação, sendo estas da escala ordinal[21]. As Tabelas 4.2, 4.3 e 4.4 ilustram os valores de mediana e moda obtidos para Utilização, Utilidade e Adequação, respectivamente. Nos itens a seguir esses resultados são discutidos.

a) *Mediana para Utilização* - Cada EC foi avaliado por 12 participantes diferentes e a mediana para Utilização demonstrou que o elemento central, para quase todos os ECs, é "3". Portanto todos os elementos do conjunto de dados acima do elemento central, também recebem o valor "3"; reparando que este é o maior número possível de resposta. Ou seja, pelo menos metade dos elementos ocorridos do conjunto de dados é "3", logo cinqüenta porcento dos participantes responderam que o EC foi usado.

b) *Moda para Utilização* - Notou-se que o valor que mais ocorre é o "3", demonstrando *alta utilização* para quase todos os ECs. Conforme consta no documento de instrumentação, os valores são: (1) não usado e não gostaria de usar, (2) não usado, mas gostaria de usar e (3) usado.

Tabela 4.2 – Medidas de tendência central do aspecto de utilização

Utilização	ECs													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Mediana	3	3	3	3	3	3	3	2,5	3	3	3	3	3	3
Moda	3	3	3	3	3	3	3	{2,3}	3	3	3	3	3	3

[21] Os valores de uma escala ordinal representam diferentes tipos nos quais um elemento pode ser ordenado, ainda que sem qualquer interpretação numérica.

c) *Mediana para Utilidade* - Cada EC foi avaliado por 12 participantes diferentes e a mediana para Utilidade demonstrou que o elemento central, para quase todos os ECs, é "3". Portanto todos os elementos do conjunto de dados acima do elemento central, também recebem o valor "3"; reparando que este é o maior número possível de resposta. Ou seja, pelo menos metade dos elementos ocorridos do conjunto de dados é "3", logo cinqüenta porcento dos participantes responderam que o EC foi considerado útil.

d) *Moda para Utilidade* - Notou-se que o valor que mais ocorre é o "3", demonstrando *alta utilidade* para quase todos os ECs. Conforme consta no documento de instrumentação, os valores são: (1) não se demonstrou útil, (2) provavelmente é útil e (3) é útil.

Tabela 4.3 – Medidas de tendência central do aspecto de utilidade

Utilidade	ECs													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Mediana	3	3	3	3	3	2,5	3	3	3	3	3	3	3	3
moda	3	3	3	3	3	{2,3}	3	3	3	3	3	3	3	3

e) *Mediana para Adequação* - Cada EC foi avaliado por 12 participantes diferentes e a mediana para Adequação demonstrou que o elemento central, para quase todos os ECs, é "3". Portanto todos os elementos do conjunto de dados acima do elemento central, também recebem o valor "3"; reparando que este é o maior número possível de resposta. Ou seja, pelo menos metade dos elementos ocorridos do conjunto de dados é "3", logo cinqüenta porcento dos participantes responderam que o EC teve nível de descrição suficiente. Para os casos em que a mediana é "2,5", o elemento central ficou entre um elemento com valor "2" e um elemento com valor "3", por conseguinte, os elementos ocorridos depois da mediana podem receber apenas o valor "3". Para o caso em que a mediana é "1", não se pode assegurar que metade superior da mediana receberá valor somente igual a "3".

f) *Moda para Adequação* - Para o aspecto de Adequação do Nível de Descrição notou-se que o valor que mais ocorre é o "3", demonstrando *alto nível de descrição*, ou seja, não precisa de modificações, para quase todos os ECs. Conforme consta no documento de instrumentação, os valores são: (1) a descrição deve ser aumentada, (2) a descrição deve ser diminuída e (3) a descrição não precisa ser modificada.

Tabela 4.4 – Medidas de tendência central do aspecto de adequação

Adequação	ECs													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Mediana	3	2,5	3	2,5	3	1	3	3	3	3	3	3	2	3
Moda	3	3	3	3	3	1	3	3	3	3	3	3	{1,3}	3

3) Teste binomial para destaque de agrupamentos

- Grupo de ECs que foram considerados *Usados* e também *Úteis*.

Este grupo caracterizou-se por obter na escala discretizada[22] pré-estabelecida, os valores de *alta* utilização, *alta* utilidade e *média* necessidade de adequação, conforme mostra a Tabela 4.5.

Tabela 4.5 – Teste binomial do grupo de ECs Usados e Úteis

ECs/Características	Utilização	Utilidade	Adequação	Características
Lista de Inferência	9:3	12:0	8:4	Mesmo os ECs tendo alto índice de utilização e utilidade, o detalhamento deve ser modificado porque foi considerado insuficiente para o entendimento. Para todos os ECs não usados, houve interesse dos participantes em usá-los. Independente dos ECs terem sido usados ou não, a maioria dos participantes considerou que esses ECs são úteis.
Lista de Informações de Negócio	11:1	12:0	6:6	
Matriz de Barramento	10:2	11:1	8:4	
Modelo Dimensional	11:1	12:0	6:6	
Analisar o Contexto de Dados	10:2	12:0	8:4	
Projetar Arquitetura de DW	9:3	12:0	6:6	
Construir o Modelo de Dados	11:1	12:0	7:5	

- Grupo de ECs que foram considerados *Úteis* e *Mal-Compreendidos*.

Este grupo caracterizou-se por obter, na escala discretizada pré-estabelecida, os valores de *média* utilização, *alta* utilidade e *alta* necessidade de adequação. A Tabela 4.6 ilustra esses valores.

[22] Onde os valores de *Alta* (Utilização, Utilidade ou Adequação), estão compreendidos no intervalo de {9 a 12} elementos positivos existentes, da mesma forma os valores de *Média* estão no intervalo de {6 a 8} e os de *Baixa* estão no intervalo de {0 a 5}.

Tabela 4.6 – Teste binomial do grupo de ECs Úteis e Mal-Compreendidos

ECs/Características	Utilização	Utilidade	Adequação	Características
Descrição de Base de Dados Existente	7:5	11:1	9:3	Todos esses ECs exigem melhor descrição do seu conteúdo, provavelmente para o aumento de sua utilização, já que eles são considerados de alta utilidade. Apesar desses ECs não terem sido usados pela maioria dos participantes, houve interesse em sua utilização.
Especialista de KDD	6:6	11:1	9:3	
Usuário Final (tomador de decisão)	7:5	11:1	9:3	
Gerar Lista de Informações de Negócio	8:4	12:0	9:3	

- Grupo de ECs que foram considerados *Úteis*.

Este grupo caracterizou-se por obter na escala discretizada pré-estabelecida, os valores de *média* utilização, *alta* utilidade e *média* necessidade de adequação, conforme pode ser visto na Tabela 4.7.

Tabela 4.7 – Teste binomial do grupo de ECs Úteis

ECs/Características	Utilização	Utilidade	Adequação	Características
Engenheiro de Conhecimento	7:5	11:1	8:4	Mesmo esses ECs terem sido considerados de alta utilidade, provavelmente a sua utilização poderia ter sido maior, se houvesse melhor descrição do seu conteúdo.
Analisar o Contexto de Ferramentas	7:5	12:0	8:4	

- Grupo de ECs que foram considerados *Compreendidos*.

Este grupo caracterizou-se por obter na escala discretizada pré-estabelecida, os valores de *média* utilização, *alta* utilidade e *baixa* necessidade de adequação. Esses valores são mostrados na Tabela 4.8.

Tabela 4.8 – Teste binomial do grupo de ECs Compreendidos

ECs/Características	Utilização	Utilidade	Adequação	Características
Caracterização de Ferramentas Existentes	7:5	12:0	5:7	Esse EC foi compreendido e considerado altamente útil, ainda que não tenha sido usado por alguns participantes. Mesmo para aqueles participantes que não usaram este EC, houve interesse em utilizá-lo.

4) Aplicação do teste estatístico não-paramétrico – Qui-Quadrado

Segundo Montgomery e Runger (2003), uma teoria científica é julgada por meio de métodos estatísticos, que desempenham um papel importante no planejamento, condução e análise de dados de experimentos. A experimentação requer a decisão de aceitar ou rejeitar uma afirmação acerca de um parâmetro, esta afirmação é chamada de hipótese e o procedimento de tomada de decisão sobre a hipótese é chamado teste de hipóteses.

O teste de hipóteses escolhido foi o qui-quadrado, este é usado quando se deseja comparar freqüências observadas com freqüências esperadas. A forma de aplicação deste teste é chamada de Tabela de Contingência ou teste de independência, que está ilustrada na Tabela 4.9.

Tabela 4.9 – Tabela de contingência *r(row) versus c(column)*

		Colunas			
		1	2	...	*c*
linhas	1	O_{11}	O_{12}	...	O_{1c}
	2	O_{21}	O_{22}	...	O_{2c}
	...	...	...	...	...
	r	O_{r1}	O_{r2}	...	O_{rc}

Fonte: Montgomery e Runger (2003, p.92)

Para Montgomery e Runger (2003), os *n* elementos de uma amostra provenientes de uma população podem ser classificados de acordo com dois critérios diferentes. E torna-se interessante saber se os dois métodos de classificação são estatisticamente independentes. Os procedimentos para este teste são explicados a seguir.

Seja p_{ij} a probabilidade de um elemento selecionado aleatoriamente cair a $ij-$*éssima* célula, dado que as duas classificações sejam independentes.

Então $p_{ij} = u_i v_j$, em que u_i é a probabilidade de um elemento selecionado aleatoriamente cair na linha classe i e v_j é a probabilidade de um elemento selecionado aleatoriamente cair na coluna classe j.

Agora, supondo independência, os estimadores de u_i e v_j são:

$$\hat{u}_i = \frac{1}{n}\sum_{j=1}^{c} Oij \qquad (4.1)$$

$$\hat{v}_j = \frac{1}{n}\sum_{i=1}^{r} Oij \tag{4.2}$$

Logo, a freqüência esperada de cada célula é:

$$Eij = n\hat{u}_i\hat{v}_j = \frac{1}{n}\sum_{j=1}^{c} Oij \sum_{i=1}^{r} Oij \tag{4.3}$$

Assim, para *n* grande, a estatística:

$$X_0^2 = \sum_{i=1}^{r}\sum_{j=1}^{c} (Oij - Eij)^2 / Eij \tag{4.4}$$

tem uma distribuição aproximada qui-quadrado com $(r-1)(c-1)$ graus de liberdade, se a hipótese nula for verdadeira. Por conseguinte, rejeitaríamos a hipótese de independência, se o valor observado da estatística de teste X_0^2 excedesse $X^2_{\alpha,(r-1)(c-1)}$, onde α é o grau de liberdade, ou margem de erro suportada pelo teste estatístico.

Aplicação do Teste - Para cada EC foi aplicado o teste estatístico qui-quadrado para determinar: (1) Se pode considerar que o EC é usado, (2) Se pode considerar que o EC é útil e (3) Se pode considerar que a descrição do EC não precisa de modificação. Esta situação representa a ideal, ou seja, valores de SUA devem receber os valores {1,1,0}.

A seguir estão descritos os passos realizados na aplicação do teste qui-quadrado.

Primeiro Passo *– Buscar a quantidade de valores SUA {1,1,0} no conjunto de resposta para cada EC.*

Resultado Obtido: está resumido na Tabela 4.10, que mostra as várias distribuições diferentes para a condição ideal.

Tabela 4.10 – Resumo das condições ideais encontradas

Elementos-Chave (ECs)	Condições Ideais Encontradas
Lista de Inferência	6 (seis)
Lista de Informações de Negócio	6 (seis)
Matriz de Barramento	6 (seis)
Modelo Dimensional	5 (cinco)
Descrição de Base(s) de Dados Existente(s)	6 (seis)
Caracterização de Ferramentas Existentes	3 (três)
Engenheiro de Conhecimento	3 (três)
Especialista de KDD	4 (quatro)
Usuário Final (Tomador de Decisão)	5 (cinco)
Analisar o Contexto de Dados	6 (seis)
Analisar o Contexto de Ferramentas	4 (quatro)
Gerar Informação de Negócio	6 (seis)
Projetar Arquitetura de DW	3 (três)
Construir a Modelagem de Dados	7 (sete)

Segundo Passo – *Calcular* u_i e v_j *, usando as equações (4.1) e (4.2).*

Dada a Tabela de Contingência elaborada para este experimento, conforme mostrado na Tabela 4.11., foram calculados os valores de u_i e v_j para todos os ECs.

Tabela 4.11 – Freqüências ocorridas para a Lista de Inferência

Tabela de Contingência		Respostas		
		Positivas	Negativas	Total
Distribuição	Ideal	*12*	*0*	*12*
	Real	*6*	*6*	*12*
	Total	*18*	*6*	*24*

Resultado Obtido: valores de u_i e v_j

$u_1 = (12/24) = 0{,}50$

$u_2 = (12/24) = 0{,}50$

$v_1 = (18/24) = 0{,}75$

$v_2 = (6/24) = 0{,}25$

Terceiro Passo *– Calcular as freqüências esperadas para cada EC, usando a equação (4.3).*

As freqüências esperadas podem agora ser calculadas.

$$E_{11} = n\hat{u}_1\hat{v}_1 = 24(0{,}50)(0{,}75) = 9$$
$$E_{12} = n\hat{u}_1\hat{v}_2 = 24(0{,}50)(0{,}25) = 3$$
$$E_{21} = n\hat{u}_2\hat{v}_1 = 24(0{,}50)(0{,}75) = 9$$
$$E_{22} = n\hat{u}_2\hat{v}_2 = 24(0{,}50)(0{,}25) = 3$$

Resultado Obtido: está resumido na Tabela 4.12, mostrando as freqüências esperadas para o EC - Lista de Inferência.

Tabela 4.12 – Freqüências esperadas para a Lista de Inferência

Tabela de Contingência		Respostas		
		Positivas	Negativas	Total
Distribuição	Ideal	*9*	*3*	*12*
	Real	*9*	*3*	*12*
	Total	*18*	*6*	*24*

Quarto Passo *– Determinar a margem de erro*

Resultado Obtido: deseja-se usar a margem de erro como o valor de $\alpha = 0{,}05$ (5%).

Quinto Passo *– Determinar o grau de liberdade [* $X^2_{\alpha;(r-1)(c-1)}$ *]*

Resultado Obtido: $X^2_{0,05;(2-1)(2-1)} = X^2_{0,05;(1)(1)} = X^2_{0,05;(1)} = 3{,}84$

Sexto Passo *– Calcular o valor de* X_0^2 *para cada EC, usando a equação (4.4).*

Resultado Obtido: $X_0^2 = \sum_{i=1}^{2} \sum_{j=1}^{2} (Oij - Eij)^2 / Eij$, para o EC – Lista de Inferência o valor será 8, conforme é demonstrado a seguir:

$$= \frac{(12-9)^2}{9} + \frac{(0-3)^2}{3} + \frac{(6-9)^2}{9} + \frac{(6-3)^2}{3} = \frac{9}{9} + \frac{9}{3} + \frac{9}{9} + \frac{9}{3} = 1 + 3 + 1 + 3 = 8$$

Assim obteve-se para cada EC o valor qui-quadrado correspondente, o resultado está resumido na Tabela 4.13.

Tabela 4.13 – Valor qui-quadrado de cada EC

Distribuições	Valor qui-quadrado	Elementos-Chave (ECs)
(+12): (- 0)	0,0	
(+11): (- 1)	1,04	
(+10): (- 2)	2,18	
(+ 9): (- 3)	3,43	
$X^2_{0,05;(1)}$	**3,84**	**qui-quadrado para o grau de liberdade 1. (verificação na tabela pré-estabelecida, pela estatística)**
(+ 8): (- 4)	4,80	
(+ 7): (- 5)	6,32	Construir a Modelagem de Dados
(+ 6): (- 6)	8,00	Lista de Inferência, Lista de Informação de Negócio, Matriz de Barramento Descrição de Bases de Dados Existentes Analisar o Contexto de Dados Gerar Informação de Negócio
(+ 5): (- 7)	9,88	Modelo Dimensional Usuário Final (tomador de decisão)
(+ 4): (- 8)	12,00	Especialista de KDD Analisar o Contexto de Ferramentas
(+ 3): (- 9)	14,40	Caracterização das Ferramentas Existentes Engenheiro de Conhecimento Projetar Arquitetura de DW
(+ 2): (-10)	17,14	
(+ 1): (-11)	20,31	
(+ 0): (-12)	24,00	

Sétimo Passo *– Julgar as Hipóteses*

Resultado Obtido: devido o grau de liberdade ser 1, os valores de X_0^2 encontrados para todos os EC são maiores que $X^2_{0,05,(1)}$, conforme é mostrado na Tabela 4.13.

Portanto, rejeita-se a H0. [$X_0^2 > X^2_{0,05,(1)} = 3,84$].

5) Análise quantitativa

Para verificar a similaridade entre os ECs usados pelos desenvolvedores do UPKDD e os ECs do Conjunto Comum, é necessário calcular o valor da variável dependente 1 (similaridade entre os ECs usados do UPKDD e os ECs do Conjunto Comum). O conjunto de ECs usados pelos desenvolvedores do UPKDD e os ECs do Conjunto Comum podem ser considerados iguais, pois existe o valor SUA = {1,X,X} presente em todos os ECs. Portanto o grau de similaridade é de 100%.

Para verificar a utilidade dos ECs similares, é necessário calcular o valor de variável dependente 2 (utilidade de ECs similares). Os ECs usados do UPKDD podem ser considerados úteis com um grau de 90% a 100%, pois existe o valor de SUA ={1,1,X} presente em todos os ECs.

Para verificar o nível de descrição dos ECs usados, é necessário calcular o valor da variável dependente 3 (adequação de ECs similares). Os ECs usados do UPKDD apresentam a seguinte distribuição: 72% não precisa de modificação, 3% a modificação é indiferente e 7% precisa de modificação, o valor de SUA = {1,X,0} está presente em todos os ECs.

6) Análise qualitativa

Para analisar a informação referente aos ECs não usados, mas que os desenvolvedores gostariam de usar, foi elaborada uma lista a partir dos valores de SUA = {0, X, X}. O resultado obtido foi que todos os ECs, que não foram usados, os desenvolvedores gostariam de usar. Os 14 ECs foram avaliados por 12 participantes, totalizando 168 resultados, destes 48 ECs foram considerados como não usados e 120 ECs considerados como usados.

7) Verificação das hipóteses

Para fazer conclusões relevantes sobre as hipóteses *H0* e *H1*, é necessário entender inicialmente os resultados apresentados nas Tabelas 4.10 e 4.13. Nestas tabelas é demonstrado que todas os ECs possuem 7 ou menos condições ideais no conjunto de respostas e, conseqüentemente, todos os ECs possuem valores de distribuição qui-quadrado maiores que o valor qui-quadrado com grau e liberdade 1 e margem de erro de 5%. Portanto, todos os 14 ECs avaliados pelos 12 participantes possuem condições ideais em relação aos valores de SUA={1,1,0}. Isto prova que se determinado EC obteve como resposta *alguma* condição ideal, ele faz parte do Conjunto Comum de ECs. Portanto todos os 14 ECs usados do UPKDD fazem parte do Conjunto Comum, logo **rejeita-se *H0***, **aceitando-se *H1***.

Neste experimento adotou-se o Conjunto Comum de ECs como referência de elementos fundamentais para o desenvolvimento de soluções KDD, pois todos os ECs pertencentes a este conjunto possuem condições ideais para os valores de SUA e representam um consenso dos autores de arquitetura e implementação do processo KDD e OLAP.

Como visto anteriormente, todos os ECs usados do UPKDD fazem parte do Conjunto Comum. A partir disto, *H2* investiga se os ECs foram considerados inúteis para o

desenvolvimento de soluções KDD. Para a análise de *H2* é necessário entender os resultados obtidos nas Tabelas 4.5.; 4.6; 4.7 e 4.8. Nestas tabelas todos os grupos de ECs apresentam a variação de 91,6% a 100% de Utilidade. Portanto, não existindo ECs considerados inúteis pelos desenvolvedores do UPKDD, devido a baixa porcentagem obtida, logo **aceita-se *H2***.

Estabelecido que todos os ECs usados do UPKDD fazem parte do Conjunto Comum e que de 90% a 100 % destes são úteis, *H3* procura ECs cuja a descrição deva ser modificada, para atingir o nível esperado pelos desenvolvedores. Para analisar *H3* é necessário compreender as Tabelas 4.1; 4.5; 4.6; 4.7 e 4.8. Na Tabela 4.8 é possível visualizar que apenas 7% dos ECs (um EC – Caracterização das Ferramentas Existentes), apresenta necessidade de mudança do nível de descrição do seu conteúdo descritivo. Nas Tabelas, 4.5, 4.6 e 4.7 foi demonstrado que a modificação não é necessária para 10 ECs (71%) e também que a modificação é indiferente para esta amostra para 3 ECs (22%); logo **aceita-se *H3.***

Conforme resultados obtidos na análise qualitativa dos dados, descritos na Seção 4.3.4 (item 6) e Tabelas 4.5.; 4.6; 4.7 e 4.8, existem apenas ECs que os desenvolvedores não usaram e que gostariam de usar, logo **aceita-se *H4.***

4.4. CONSIDERAÇÕES FINAIS

Este capítulo apresentou o estudo de caso realizado, a partir do modelo e experimentação em engenharia de software. Nele é possível identificar que a mensuração do processo proposto seguiu passos bem delineados, a fim de quantificar variáveis que mostram o efeito da mudança como a utilização, utilidade e adequação do UPKDD.

Para que o estudo de caso quantitativo pudesse acontecer, foram envolvidas duas equipes de desenvolvedores na produção de uma mesma solução KDD. Sem estas equipes, a realização do estudo de caso não seria possível, pelo menos não para os objetivos de medição identificados.

Durante a fase de interpretação do estudo de caso, foi necessário entender as particularidades do teste de hipótese estatística qui-quadrado. Este teste pode ser aplicado de formas diferentes, sendo que em algumas delas as freqüências esperadas é calculada sobre técnicas de porcentagem. No caso da tabela de contingência, o cálculo das freqüências esperadas não é baseado em porcentagem como foi mostrado neste capítulo.

No próximo capítulo são sumarizadas as principais contribuições e identificados alguns trabalhos futuros para continuidade da pesquisa realizada e apresentada nesta dissertação de mestrado.

5 CONCLUSÕES E TRABALHOS FUTUROS

A intenção de formalizar o processo de desenvolvimento de aplicações KDD, investigada por Dias (2001), prova que a ordenação rigorosa de atividades para a descoberta de conhecimento diminui a característica de indeterminismo desses sistemas.

Neste sentido, este trabalho de mestrado propôs um *Processo de Software para aplicações KDD,* denominado **UPKDD**. A seqüência de atividades do processo proposto foi modelada usando os recursos do metamodelo SPEM, já que este serve como modelo unificado das metodologias existentes para a modelagem de processos.

Para a separação semântica de fases e *workflows* deste modelo de processo, foi escolhida a estrutura dinâmica e estática do UP, visto que o mesmo é um processo de software estabelecido e que permite personalizações, como foi o caso do UPKDD.

Tanto para modelar o processo usando uma notação específica como o SPEM, quanto para rever os elementos-chave do UP, foi necessário empenho considerável na configuração de um ambiente que permitisse o *design* do UPKDD e no entendimento dos detalhes do processo tradicional de desenvolvimento de software.

Para uma aplicação KDD ter sucesso, esta depende principalmente de uma estrutura de dados formatada para suportar aplicações analíticas e um controle para a condução da descoberta de conhecimento, especialmente em banco de dados.

Por meio da avaliação do processo proposto, observou-se que o mesmo é adequado para aplicações deste tipo, pois os resultados quantitativos do estudo de caso realizado mostram alta utilização dos elementos-chave propostos comprovadamente similares ao Conjunto Comum de ECs.

Juntamente ao acompanhamento dos grupos de participantes, durante o exercício de desenvolvimento de uma solução KDD para um laboratório de análises clínicas, comprovou-se a necessidade de um processo realmente focado nas características deste tipo de aplicação como o UPKDD.

A escolha por um dos nove métodos de experimentação em engenharia de software dependeu principalmente do objetivo de medição, que foi representar o grau de qualidade do UPKDD por meio do valor dado à utilidade do mesmo.

As contribuições deste trabalho de mestrado tiveram impacto nos elementos-chave do UP, para abranger a realidade de descoberta de conhecimento em banco de dados.

Algumas visões foram agregadas ao processo de software tradicional, como a visão de *condução do processo KDD*, representado pela Lista de Inferências. Esta representa o direcionamento ou objetivo que norteia a busca por conhecimento em sistemas de apoio a decisão. Em complemento à Lista de Inferência, que trata de questões hipotéticas sugeridas pelos usuários finais, surgiu um modelo para mapear estas expectativas em situações realísticas dadas às condições das bases de dados existentes, mapeamento este denominado como Lista de Informações de Negócio.

Outra visão incorporada ao processo tradicional foi a visão da *estrutura dos dados* empresariais. Para este tipo de aplicação tem-se por base que os sistemas operacionais tratam perfeitamente situações transacionais da empresa. Portanto, durante as atividades de desenvolvimento não existe preocupação em construir o banco de dados relacional. Esta visão nova está focada em como explorar os dados, tornando-os úteis, para isso é necessário transformá-los em uma estrutura dimensional, possibilitando prever eventos futuros usando dados históricos da empresa. Esta estrutura deve ser favorável ao uso de tecnologias analíticas que apóiam o tomador de decisões.

Então, pode-se afirmar que o UPKDD é dirigido por **Objetivos de Descoberta de Conhecimento** e centrado em **Arquitetura e Tecnologia Analíticas**.

Logo como diferencial este trabalho destaca-se pela personalização do UP para a caracterização dos sistemas de KDD, incluindo o rigor da modelagem de processos, por meio da adoção dos estereótipos do metamodelo SPEM. Assim como pelo rigor matemático inserido no estudo de caso apoiado pelos testes de hipóteses estatísticas.

5.1. TRABALHOS FUTUROS

Como trabalhos futuros, podem-se destacar:

- A investigação das definições do UP não tratados por este trabalho. Como as Fases de Construção e Transição e, também, os *Workflows* de Implementação e Testes. De maneira a incorporar novos elementos-chave representativos para aplicações KDD.
- A realização de outros exercícios para outras soluções e domínios KDD. Possibilitando a replicação de projetos com equipes de trabalho diferenciadas e contando com mais tempo para obtenção de resultados significativos.

- A implementação de um ambiente de apoio ao processo, conhecido por PSEE. Para implementar a configuração realizada na ferramenta Rational Rose também em outras ferramentas, ou ainda, implementar funcionalidades novas como modelagem de processo usando SPEM, em uma nova ferramenta.
- O refinamento do processo modelado seguindo o metamodelo SPEM.
- A elaboração de mecanismos de rastreabilidade entre fases e *workflows* do UPKDD.
- O uso de outros métodos de avaliação de processos, recursos e ferramentas. Por exemplo, realizar um experimento formal para soluções KDD.

REFERÊNCIAS

ACUÑA, S. T.; FERRÉ, X. Software process modelling. In: WORLD MULTI-CONFERENCE ON SYSTEMICS, CYBERNETICS AND INFORMATICS (ISAS-SCI2001), 5, 2001, Orlando. **Proceedings…** Orlando, Florida, USA, 2001. p.1-6. Disponível em:< http://is.ls.fi.upm.es/udis/miembros/xavier/papers/processmodelling.pdf >. Acesso em: set. 2006.

ALMEIDA, M. S. O. **MGUP:** RUP aplicado a jogos móveis. 2006. 118f. Dissertação (Mestrado em Ciência da Computação) – Departamento de Informática (DIN), Universidade Estadual de Maringá (UEM), Maringá, PR, 2006.

ÁLVARES, P. M. R. S. A definição de um processo. In: **WebPraxis:** um processo personalizado para projetos de desenvolvimento para a web. 2001. Dissertação (Mestrado em Ciência da Computação) – Instituto de Ciências Exatas, Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, MG, 2006. p. 31-38.

BARANAUSKAS, J. A. **Extração automática de conhecimento por múltiplos indutores**. 2001. Tese (Doutorado em Ciências – Ciências da Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação (ICMC-USP), Universidade de São Paulo (USP), São Carlos, SP, 2001.

BARROS, M. O. et al. **Métodos estatísticos aplicados à engenharia de software experimental.** 2006. Apresentação em Power-point. Disponível em <http://www.uniriotec.br/~marcio.barros/>. Acesso em: 04 jul. 2007.

BASILI, V. R.; SELBY, R. W. Jr.; HUTCHENS, D. H. **Experimentation in software engineering**, Departamento de Computer Science – College Park, Universidade de Maryland, USA, (Relatório Técnico/Científico TR1575), nov. 1985, 32p.

BASILI, V. R.; ROMBACH, H. D.; SELBY, R. W. Jr. The experimental paradigm in software engineering. In: DAGSTUHL-WORKSHOP, 706, 1992. Experimental software engineering issues: critical assessment and future directives, Lecture Notes in Computer Software. **Proceedings…** Springer-Verlag, ago. 1993.

BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. The goal question metric approach. In: MARCINIAK, J.J. (ed.). **Encyclopedia of Software Engineering**. New York: John Wiley & Sons, 1994. p.528-532.

BORBA, S. F. P. Metodologias de desenvolvimento do modelo multidimensional. In: **Metodologia para implantação de modelos multidimensionais em banco de dados**

orientado a objetos. 2006. Tese (Doutorado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC, 2006. p. 89-109.

BRACHMAN, R. J.; ANAND, T. The process of knowledge discovery in databases: a human-centered approach. In: FAYYAD, U. M.; PIATETSKY-SHAPIRO,G.; SMYTH, P.; UTHURUSAMY, R., (editores). **Advances in Knowledge Discovery and Data Mining**. Menlo Park, Calif.: American Association for Artificial Intelligence (AAAI)/MIT Press, 1996. p. 37-57.

BULCÃO NETO, R. F. Um processo para aplicações sensíveis a contexto. In: **Um processo de software e um modelo ontológico para apoio ao desenvolvimento de aplicações sensíveis a contexto**. 2006. Tese (Doutorado em Ciências – Ciências da Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação (ICMC-USP), Universidade de São Paulo (USP), São Carlos, SP, 2006. p. 111-122.

CEREJA JUNIOR, M. G.; SANT'ANNA, N.; BORREGO FILHO, L. F. UML e PML: uma exploração de abordagens para a modelagem de processos. In: SIMPÓSIO INTERNACIONAL DE MELHORIA DE PROCESSO DE SOFTWARE (SIMPROS), 4, Recife, 2002. **Anais eletrônicos...** 2002. Disponível em: <http://www.simpros.com.br>. Acesso em: out. de 2006.

CHAPMAN, P. et al. **CRoss Industry Standard Process for Data Mining (CRISP-DM 1.0**): step-by-step data mining guide, CRISP-DM Consortiun, USA, (Relatório Técnico CRISP-DM), 2000. 77p. Disponível em: <http://www.crisp-dm.org/process2.htm>. Acesso em: 05 maio 2006.

CURTIS, B.; KELLNER, M. I.; OVER, J. Process modeling. **Communications of the ACM**, v. 35, n. 9, p. 75-90, set. 1992.

DATE, C. J. Apoio à decisão. In: **Introdução a sistemas de banco de dados**. 8. ed. Rio de Janeiro: Elsevier, 2003. p. 590-620.

DIAS, M. M. **Um modelo de formalização do processo de desenvolvimento de sistemas de descoberta de conhecimento em banco de dados**. 2001. 197f. Tese (Doutorado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC, 2001.

FARIAS, L. L. Estudo experimental da relação entre fatos e riscos de projeto de software. In: **Planejamento de riscos em ambientes de desenvolvimento de software orientados à organização**. 2002. Dissertação (Mestrado em Ciências em Engenharia de Sistemas e Computação) – COPPE/UFRJ, Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, RJ, 2002. p. 106-136.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in database. In: **Advances in Knowledge Discovery and Data Mining**. Menlo Park, Calif: American Association for Artificial Intelligence (AAAI Press): MIT Press, 1996. p. 37–54.

FERREIRA, A. B. de H. **Novo dicionário da língua portuguesa**. 2. ed. Rio de Janeiro: Nova Fronteira, 1986.

FELDENS, M. A. et al. Towards a methodology for the discovery of useful knowledge combining data mining, data warehousing and visualization. In: CLEI – CONFERÊNCIA LATINO-AMERICANA DE INFORMÁTICA, 24, 1998, Quito. **Anais eletrônicos...** Quito, Equador. 1998. Disponível em: <http://jacui.inf.ufrgs.br/~feldens/clei98.html>. Acesso em 22 abril 2006.

FUGGETTA, A. Software process: a roadmap. In: FINKELSTEIN, A. (ed.). CONFERENCE ON THE FUTURE OF SOFTWARE ENGINEERING (ICSE00), Limerick, 2000. **Proceedings…** Limerick: ACM Press, jun. 2000. p. 25–34.

GENVIGIR, E. C. **Um modelo de processo da engenharia de requisitos aplicável a ambientes de engenharia de software centrados em processo**. 2004. 251f. Dissertação (Mestrado em Computação Aplicada), Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, SP, 2004.

GOEBEL, M.; GRUENWALD, Le. A survey of data mining and knowledge discovery software tools. **ACM SIGKDD Explorations Newsletter**, v. 1, n. 1, p. 20-33, jun. 1999.

GUPTA, S. K.; BHATNAGAR, V.; WASAN, S. K. **Architecture for knowledge discovery and knowledge management**. London: Springer-Verlag, Knowledge and Information Systems Journal, v. 7, n. 3, p. 310–336, 2005.

GROTH, R. **Data mining:** a hands-on approach for business professionals. Prentice-Hall PTR, 1998.

HAN, J.; KAMBER, M. **Data mining**: concepts and techniques. In: GRAY, Jim (ed.). USA: The Morgan Kaufmann Series, 2001. 500p. (Series in Data Management Systems).

HAUCK, J. C. R. ; WANGENHEIM, C. G. V. Modelando o processo de software em uma pequena empresa - o caso void caz. In: SIMPÓSIO INTERNACIONAL DE MELHORIA DE PROCESSOS DE SOFTWARE (SIMPROS), 6, 2004. **Anais eletrônicos...** Disponível em: <http://www.simpros.com.br>. Acesso em: outubro de 2006. p. 251-262.

IBM. Disponível em: <http://www.ibm.com/news/br/pr/2006/02/20-02-2006.htm>. Acesso em maio 2006.

INMON, W.H. **Como construir o data warehouse**. Rio de Janeiro: Campus, 1997. 388p.

JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. **The unified software development process**. 2.ed. Canadá: Addison-Wesley, 1999. 463p., (Object Technology Series).

JÄGER, D.; SCHLEICHER, A.; WESTFECHTEL, B. Using UML of software process modeling. In: EUROPEAN SOFTWARE ENGINEERING CONFERENCE e ACM SIGSOFT INTERNATIONAL SYMPOSIUM ON THE FOUNDATIONS OF SOFTWARE ENGINEERING (ESEC/FSE'99), 7, set. 1999, Toulouse, França. **Proceedings…** 1999. p. 91-108.

KELLNER, M. I.; HANSEN, G. A. **Software process modeling**. Carnegie Mellon University, Software Engineering Institute (SEI), Pittsburgh, Pennsylvania, (Relatório Técnico CMU/SEI-88-TR-009 ESD-TR-88-010), 1988. 52p.

KIMBALL, R.; ROSS, M. **Data warehouse toolkit:** o guia completo para modelagem multidimensional. Rio de Janeiro: Campus, 2002. 494p.

KIMBALL, R. et al. Introducing data warehouse architecture. In: **The data warehouse lifecycle toolkit:** expert methods for designing, developing, and deploying data warehouses. USA: Wiley Computer Publishing, 1998. p. 317-334.

KITCHENHAM, B. A.; PICKARD, L.; PFLEEGER, S. L. Case studies for method and tool evaluation. **IEEE Software**. v. 12. n. 4. jul. 1995, p. 52-62.

KITCHENHAM, B. A. Evaluating software engineering methods and tools: part 1: the evaluation context and evaluation methods. **ACM SIGSOFT – Software Engineering Notes**. v. 21, n. 1, jan. 1996a, p. 11-14.

KITCHENHAM, B. A. Evaluating software engineering methods and tools: part 2: selecting an appropriate evaluation method – technical criteria. **ACM SIGSOFT – Software Engineering Notes**. v. 21, n. 2, mar. 1996b, p. 11-15.

KITCHENHAM, B. A. Evaluating software engineering methods and tools: part 3: selecting an appropriate evaluation method – pratical issues. **ACM SIGSOFT – Software Engineering Notes**. v. 21, n. 4, jul. 1996c, p. 9-12.

KITCHENHAM, B. A. ; LINKMAN, S.; LAW, D. DESMET: a methodology for evaluating software engineering methods and tools. **Computing & Control Engineering Journal**. v. 8, n. 3, jun. 1997, p.120-126.

KITCHENHAM, B. A.; PICKARD, L. M. Evaluating software engineering methods and tools: part 9: quantitative case study methodology. **ACM SIGSOFT – Software Engineering Notes**. v. 23, n. 1, jan. 1998, p. 24-26.

KLEMETTINEN, M.; MANNILA, H.; TOIVONEN, H. A data mining methodology and its application to semi-automatic knowledge acquisition. In: DEXA WORKSHOP, 1997. **Proceedings...** Toulouse, França, 1997. p. 670-677.

KRUCHTEN, P. **Introdução ao RUP:** rational unified process. Rio de Janeiro: Ciência Moderna, 2003. 255p.

LANS, R. V. O que é data mining e uma análise do mercado de produtos. In: CONGRESSO NACIONAL DE NOVAS TECNOLOGIAS E APLICAÇÕES EM BANCO DE DADOS, 8, 1997. **Anais…**,1997.

LUJÁN-MORA, S. **Data warehouse design with UML**. 2005. 200f. Tese (Doutorado), Universidade de Alicante, Departamento de Sistemas de Computação e Software. Argentina, 2005.

MACHADO, F. N. R. **Tecnologia e projeto de data warehouse:** uma visão multidimensional. São Paulo: Érica, 2004. 318p.

MANNILA, H. **Data mining**: machine learning, statistic and databases. Departamento de Computer Science, Universidade de Helsinki,1997.

MARTINS, P. V.; SILVA, A. R. Comparação de metamodelos de processos de desenvolvimento de software. In: CONFERÊNCIA PARA A QUALIDADE NAS TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÕES, 5, 2004. Instituto Português de Qualidade. 2004. **Proceedings...**, p. 179-186. Disponível em:< http://berlin.inesc.pt/alb/static/papers/2004/pv-quatic2004.pdf >. Acesso em: set. 2006.

MENOLLI, A. L. A. **Definição de uma arquitetura de data warehousing para gestão em ciência e tecnologia no Brasil**. 2004. 109f. Dissertação (Mestrado em Ciência da Computação) – Departamento de Informática (DIN), Universidade Estadual de Maringá (UEM), Maringá, PR, 2004.

MITRA, S.; PAL, S. K.; MITRA, P. Data mining in soft computing framework: a survey. **IEEE Transactions on Neural Networks**. v. 13, n. 1, jan. 2002, p. 3-14.

MONTGOMERY D. C.; RUNGER G. C. Inferência estatística para uma única amostra. In: **Estatística aplicada e probabilidade para engenheiros**. Rio de Janeiro: LTC, 2003. p. 142-178.

OLIVEIRA JUNIOR, E. A. de. Avaliação do processo proposto. In: **Um processo de gerenciamento de variabilidade para linha de produto de software**. 2005. Dissertação (Mestrado em Ciência da Computação) – Departamento de Informática (DIN), Universidade Estadual de Maringá (UEM), Maringá, PR, 2005, p. 99-138.

OMG Object Management Group. **Software process engineering metamodel specification (SPEM)**. (Relatório Técnico - OMG document number formal/05-01-06), 2005. Disponível em: <http://www.omg.org>. Acesso em: set. 2006.

ORACLE. Oracle Business Intelligence Discoverer, versão 10g. 2007. Disponível em: <http://www.oracle.com/technology/products/discoverer/index.html>. Acesso em: abril 2007.

ORACLE. Oracle Warehouse Builder (OWB), versão 11g. 2007. Disponível em: < http://www.oracle.com/technology/products/warehouse/index.html >. Acesso em: abril 2007.

PASSOS, E.; GOLDSCHMIDT, R. **Data mining**: um guia prático. Rio de Janeiro: Elsevier, 2005. 261p.

PERRY, D. E.; PORTER, A. A.; VOTTA, L. G. Empirical studies of software engineering: a roadmap. In: CONFERENCE ON THE FUTURE OF SOFTWARE ENGINEERING (ICSE00), Limerick, 2000. **Proceedings…** Limerick,: ACM, 2000. p. 345-355.

PETERS, J. F; PEDRYCZ, W. O processo de software. In: **Engenharia de Software**. Rio de Janeiro: Campus, 2001. p. 29-66.

PFLEEGER, S. L. Design and analysis in software engineering: the language of case studies and formal experiments. **ACM SIGSOFT – Software Engineering Notes**. v. 19, n. 4, out. 1994, p. 16-20.

PFLEEGER, S. L. Experimental design and analysis in software engineering: part2: how to set up and experiment. **ACM SIGSOFT – Software Engineering Notes**. v. 20, n. 1, jan. 1995, p. 22-26.

PFLEEGER, S. L. Avaliando produtos, processos e recursos. In: **Engenharia de software**: teoria e prática. 2. ed. São Paulo: Prentice Hall, 2004. p. 415-457.

PRESSMAN, R. S. **Engenharia de software**. 6. ed. São Paulo: McGraw-Hill, 2006. 720p.

REINARTZ, T. Focusing solutions for data mining: analytical studies and experimental results in real-world domains. In: SIEKMANN, J; CARBONELL, J. G. **Lecture Notes in Artificial Intelligence (LNAI)**. New York: Springer-Verlag, 1999.

REZENDE, S. O. et al. Mineração de Dados. In: REZENDE, S. O. (Org.). **Sistemas inteligentes:** fundamentos e aplicações. Barueri: Manole, 2005. p.307-335.

SAITTA, L., NERI, F. Learning in the "Real World". In: KOHAVI, R. e PROVOST, F. (editores). **Special issue on applications of machine learning and the knowledge discovery process**. Netherlands: Kluwer Academic Publishers. (Machine Learning) v. 30, p. 133-163, 1998.

SCOTT, K. **O processo unificado explicado**. Porto Alegre: Bookman, 2003. 160p.

SINGH, H. S. **Data warehouse**: conceitos, tecnologias, implementação e gerenciamento. São Paulo: Makron Books, 2001. 382p.

SOMMERVILLE, I. **Engenharia de software**. São Paulo: Addison Wesley, 2003. 592p.

SOUSA, G. M. C. Adaptando o processo unificado para o desenvolvimento de software orientado a aspectos. In: **Uma abordagem direcionada a casos de uso para o desenvolvimento de software orientado a aspectos**. 2004. Dissertação (Mestrado em Ciência da Computação) – Centro de Informática, Universidade Federal de Pernambuco (UFPE), Recife, 2004, p.69-107.

SPSS. Disponível em: <http://www.spss.com.br/press/receitarecorde.htm>. Acesso em maio 2006.

TRAVASSOS, G. H.; GUROV, D.; AMARAL, E. A. G. **Introdução à engenharia de software experimental**. Programa de Engenharia de Sistemas e Computação – (COPPE/UFRJ), Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil, (Relatório Técnico RT-ES-509/02), 2002, 52p.

VALENTIN, L. G. **Uma arquitetura de software para sistemas de descoberta de conhecimento em banco de dados**. 2006. 169f. Dissertação (Mestrado em Ciência da Computação) – Departamento de Informática (DIN), Universidade Estadual de Maringá (UEM), Maringá, PR, 2006.

WEISS S. M.; INDURKHYA N. **Predictive data mining:** a pratical guide. Canadá: Morgan Kaufmann Publishers Inc., 1998.

WEKA. versão 3. open source software. Disponível em: <http://www.cs.waikato.ac.nz/ml/weka/>. Acesso em: abril 2007.

WERNER, C. M. L; BRAGA, R. M. M. Engenharia de domínio e o desenvolvimento baseado em componentes. In: HUZITA, E. H. M.; GIMENES, I. M. S. (Org.). **Desenvolvimento baseado em componentes**: conceitos e técnicas. Rio de Janeiro: Ciência Moderna, 2005. p. 57-103.

APÊNDICE A

- CONJUNTO COMUM

O Conjunto Comum, mostrado no Quadro A1, é composto pelos Elementos-Chave indicados por autores da área de processo, arquitetura e implementação de soluções KDD (*Knowledge Discovery in Database* ou Descoberta de Conhecimento em Banco de Dados). A letra X indica posicionamento favorável à presença do Elemento-Chave.

Elementos-Chave (ECs)	**Kimball e Ross (2002)**	**Inmon (1997)**	**Singh (2001)**	**Rezende et al. (2005)**	**Passos e Goldschmidt (2005)**	**Fayyad, Piatetsky-Shapiro e Smyth (1996)**	**Brachman e Anand (1996)**
Lista de Inferência				X	X		X
Lista de Informações de Negócio	X	X	X	X	X	X	X
Matriz de Barramento	X	X					
Modelo Dimensional	X	X	X			X	
Descrição de Base(s) de Dados Existente(s)					X	X	X
Caracterização de Ferramentas Existentes	X				X		
Engenheiro de Conhecimento	X		X	X		X	X
Especialista de KDD	X		X	X	X	X	X
Usuário Final (Tomador de Decisão)	X	X	X	X	X	X	X
Analisar o Contexto de Dados						X	X
Analisar o Contexto de Ferramentas	X				X		
Gerar Informações de Negócio	X	X	X	X	X	X	X
Projetar Arquitetura de Data Warehouse	X	X					
Construir a Modelagem de Dados	X	X	X			X	

Quadro A1 – Conjunto Comum

APÊNDICE B

- ARQUIVO DE CONFIGURAÇÃO E TELA DO RATIONAL ROSE

Este Apêndice mostra como o arquivo de configuração foi alterado para aceitar os estereótipos do metamodelo SPEM. O Quadro B1 mostra um exemplo de cada linha alterada que é precedida de seu comentário. No Quadro B2 são listadas todas as alterações feitas.

Trecho Comentado – para criação de classes no ambiente

```
[Stereotyped Items]
Class:Interface
Class:enumeration
%%classe do ambiente recebe um valor novo%%
Class:processo
[Class:Interface]
Item=Class
Stereotype=Interface
[Class:enumeration]
Item=Class
Stereotype=enumeration
%%%%%criação de uma nova classe do estereotipo SPEM para instanciação%%%%%
[Class:processo]
Item=Class
Stereotype=processo
%%path da figura%%
Metafile=&\MyStereotypeIcons\processo.wmf
%%posicionamento padrao para a figura%%
ExtentX=50
%%posicionamento padrao para a figura%%
ExtentY=50
%%posicionamento padrao para o texto da figura%%
TextXPos=0
%%posicionamento padrao para o texto da figura%%
TextYPos=0
%path do icone%
ListImages=&\MyStereotypeIcons\processo_l.bmp
%%reserva um espaco na lsta de itens%%
ListIndex=1
%path do icone%
SmallPaletteImages=&\MyStereotypeIcons\processo_s.bmp
%%reserva um espaco na lista de itens%%
SmallPaletteIndex=1
%path do icone%
MediumPaletteImages=&\MyStereotypeIcons\processo_m.bmp
%%reserva um espaco na lista de itens%%
MediumPaletteIndex=1
%% nao auto redimensionamento da figura%%
AutomaticResize=NO
%%altura do texto%%
TextWidth=100
%%alinhamento do texto%
TextJust=Center
% numero de linhas para nome do estereótipo instanciado%
NameLines=2
% não proporcional%
Proportional=NO
% não redirecionamento centralizado%
CenteredResize=NO
%dica do nome do esteriotipo, Hint%
ToolTip=processo
```

Quadro B1 – Trecho de código de configuração da ferramenta

Tela do Ambiente Rose com os Estereótipos do SPEM instanciados

Na Figura B2 é mostrado um exemplo de instanciação dos onze estereótipos do SPEM na ferramenta Rational Rose.

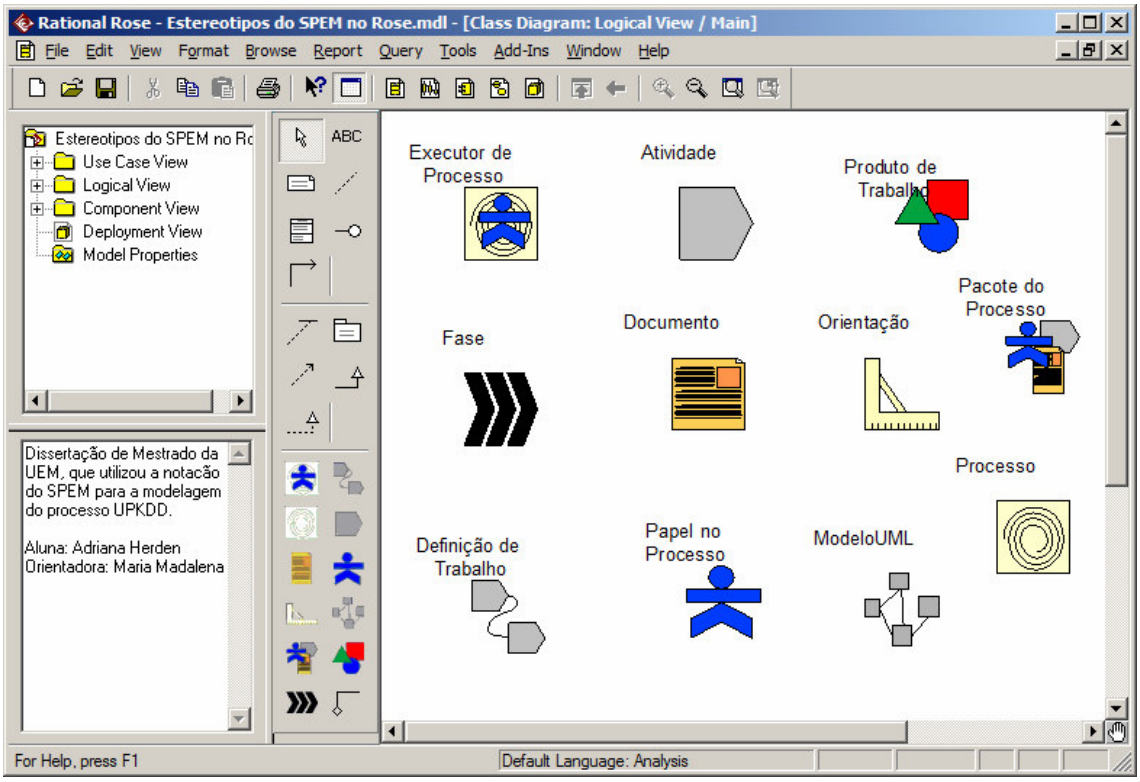


Figura B2 – Tela da ferramenta Rational Rose

Arquivo Original – defaultstereotypes.ini

```
[Stereotyped Items]
Class:Interface
Class:enumeration
Class:processo
Class:atividade
Class:definicaodetrabalho
Class:documento
Class:modelouml
Class:orientacao
Class:pacotedeprocesso
Class:papelnoprocesso
Class:executornoprocesso
Class:produtodetrabalho
Class:fase

[Class:Interface]
Item=Class
Stereotype=Interface

[Class:enumeration]
Item=Class
Stereotype=enumeration

[Class:processo]
Item=Class
Stereotype=processo
Metafile=&\MyStereotypeIcons\processo.wmf
ExtentX=50
ExtentY=50
TextXPos=0
TextYPos=0
ListImages=&\MyStereotypeIcons\processo_l.bmp
ListIndex=1
SmallPaletteImages=&\MyStereotypeIcons\processo_s.bmp
SmallPaletteIndex=1
MediumPaletteImages=&\MyStereotypeIcons\processo_m.bmp
MediumPaletteIndex=1
AutomaticResize=NO
TextWidth=100%
TextJust=Center
NameLines=2
Proportional=NO
CenteredResize=NO
ToolTip=processo

[Class:atividade]
Item=Class
Stereotype=atividade
Metafile=&\MyStereotypeIcons\atividade.wmf
ExtentX=50
ExtentY=50
TextXPos=0
TextYPos=0
ListImages=&\MyStereotypeIcons\atividade_l.bmp
ListIndex=1
SmallPaletteImages=&\MyStereotypeIcons\atividade_s.bmp
SmallPaletteIndex=1
MediumPaletteImages=&\MyStereotypeIcons\atividade_m.bmp
```

continua

continuação

```
MediumPaletteIndex=1
AutomaticResize=NO
TextWidth=100%
TextJust=Center
NameLines=2
Proportional=NO
CenteredResize=NO
ToolTip=atividade

[Class:definicaodetrabalho]
Item=Class
Stereotype=definicaodetrabalho
Metafile=&\MyStereotypeIcons\definicaodetrabalho.wmf
ExtentX=50
ExtentY=50
TextXPos=0
TextYPos=0
ListImages=&\MyStereotypeIcons\definicaodetrabalho_l.bmp
ListIndex=1
SmallPaletteImages=&\MyStereotypeIcons\definicaodetrabalho_s.bmp
SmallPaletteIndex=1
MediumPaletteImages=&\MyStereotypeIcons\definicaodetrabalho_m.bmp
MediumPaletteIndex=1
AutomaticResize=NO
TextWidth=100%
TextJust=Center
NameLines=2
Proportional=NO
CenteredResize=NO
ToolTip=definicaodetrabalho

[Class:documento]
Item=Class
Stereotype=documento
Metafile=&\MyStereotypeIcons\documento.wmf
ExtentX=50
ExtentY=50
TextXPos=0
TextYPos=0
ListImages=&\MyStereotypeIcons\documento_l.bmp
ListIndex=1
SmallPaletteImages=&\MyStereotypeIcons\documento_s.bmp
SmallPaletteIndex=1
MediumPaletteImages=&\MyStereotypeIcons\documento_m.bmp
MediumPaletteIndex=1
AutomaticResize=NO
TextWidth=100%
TextJust=Center
NameLines=2
Proportional=NO
CenteredResize=NO
ToolTip=documento

[Class:modelouml]
Item=Class
Stereotype=modelouml
Metafile=&\MyStereotypeIcons\modelouml.wmf
ExtentX=50
ExtentY=50
```

continua

continuação

```
TextXPos=0
TextYPos=0
ListImages=&\MyStereotypeIcons\modelouml_l.bmp
ListIndex=1
SmallPaletteImages=&\MyStereotypeIcons\modelouml_s.bmp
SmallPaletteIndex=1
MediumPaletteImages=&\MyStereotypeIcons\modelouml_m.bmp
MediumPaletteIndex=1
AutomaticResize=NO
TextWidth=100%
TextJust=Center
NameLines=2
Proportional=NO
CenteredResize=NO
ToolTip=modelouml

[Class:orientacao]
Item=Class
Stereotype=orientacao
Metafile=&\MyStereotypeIcons\orientacao.wmf
ExtentX=50
ExtentY=50
TextXPos=0
TextYPos=0
ListImages=&\MyStereotypeIcons\orientacao_l.bmp
ListIndex=1
SmallPaletteImages=&\MyStereotypeIcons\orientacao_s.bmp
SmallPaletteIndex=1
MediumPaletteImages=&\MyStereotypeIcons\orientacao_m.bmp
MediumPaletteIndex=1
AutomaticResize=NO
TextWidth=100%
TextJust=Center
NameLines=2
Proportional=NO
CenteredResize=NO
ToolTip=orientacao

[Class:pacotedeprocesso]
Item=Class
Stereotype=pacotedeprocesso
Metafile=&\MyStereotypeIcons\PacotedeProcesso.wmf
ExtentX=50
ExtentY=50
TextXPos=0
TextYPos=0
ListImages=&\MyStereotypeIcons\pacotedeprocesso_l.bmp
ListIndex=1
SmallPaletteImages=&\MyStereotypeIcons\pacotedeprocesso_s.bmp
SmallPaletteIndex=1
MediumPaletteImages=&\MyStereotypeIcons\pacotedeprocesso_m.bmp
MediumPaletteIndex=1
AutomaticResize=NO
TextWidth=100%
TextJust=Center
NameLines=2
Proportional=NO
CenteredResize=NO
ToolTip=pacotedeprocesso
```

continua

continuação

```
[Class:papelnoprocesso]
Item=Class
Stereotype=papeldoprocesso
Metafile=&\MyStereotypeIcons\papelnoprocesso.wmf
ExtentX=50
ExtentY=50
TextXPos=0
TextYPos=0
ListImages=&\MyStereotypeIcons\papelnoprocesso_l.bmp
ListIndex=1
SmallPaletteImages=&\MyStereotypeIcons\papelnoprocesso_s.bmp
SmallPaletteIndex=1
MediumPaletteImages=&\MyStereotypeIcons\papelnoprocesso_m.bmp
MediumPaletteIndex=1
AutomaticResize=NO
TextWidth=100%
TextJust=Center
NameLines=2
Proportional=NO
CenteredResize=NO
ToolTip=papelnoprocesso

[Class:executornoprocesso]
Item=Class
Stereotype=executornoprocesso
Metafile=&\MyStereotypeIcons\executornoprocesso.wmf
ExtentX=50
ExtentY=50
TextXPos=0
TextYPos=0
ListImages=&\MyStereotypeIcons\executornoprocesso_l.bmp
ListIndex=1
SmallPaletteImages=&\MyStereotypeIcons\executornoprocesso_s.bmp
SmallPaletteIndex=1
MediumPaletteImages=&\MyStereotypeIcons\executornoprocesso_m.bmp
MediumPaletteIndex=1
AutomaticResize=NO
TextWidth=100%
TextJust=Center
NameLines=2
Proportional=NO
CenteredResize=NO
ToolTip=executornoprocesso

[Class:produtodetrabalho]
Item=Class
Stereotype=produtodetrabalho
Metafile=&\MyStereotypeIcons\produtodetrabalho.wmf
ExtentX=50
ExtentY=50
TextXPos=0
TextYPos=0
ListImages=&\MyStereotypeIcons\produtodetrabalho_l.bmp
ListIndex=1
SmallPaletteImages=&\MyStereotypeIcons\produtodetrabalho_s.bmp
SmallPaletteIndex=1
MediumPaletteImages=&\MyStereotypeIcons\produtodetrabalho_m.bmp
MediumPaletteIndex=1
AutomaticResize=NO
```

continua

```
                                                                    continuação
TextWidth=100%
TextJust=Center
NameLines=2
Proportional=NO
CenteredResize=NO
ToolTip=produtodetrabalho

[Class:fase]
Item=Class
Stereotype=fase
Metafile=&\MyStereotypeIcons\fase.wmf
ExtentX=50
ExtentY=50
TextXPos=0
TextYPos=0
ListImages=&\MyStereotypeIcons\fase_l.bmp
ListIndex=1
SmallPaletteImages=&\MyStereotypeIcons\fase_s.bmp
SmallPaletteIndex=1
MediumPaletteImages=&\MyStereotypeIcons\fase_m.bmp
MediumPaletteIndex=1
AutomaticResize=NO
TextWidth=100%
TextJust=Center
NameLines=2
Proportional=NO
CenteredResize=NO
ToolTip=fase
                                                                fim do arquivo
```

Quadro B2 – Código de configuração

APÊNDICE C

- QUESTIONÁRIOS PARA CARACTERIZAÇÃO DOS PARTICIPANTES

Este Apêndice mostra no Quadro C1 e no Quadro C2 os questionários aplicados para a pesquisa, com a finalidade de caracterizar os participantes.

Questionário 1 – aplicado em 04 de abril de 2007.

QUESTÕES DIRIGIDAS AOS MEMBROS e LÍDERES DOS GRUPOS

1) Qual a sua idade?
( ) entre 20 e 24 anos
( ) entre 25 e 29 anos
( ) entre 30 e 34 anos
( ) entre 35 e 39 anos
( ) mais de 39

2) Pertence a qual grupo de trabalho?
( ) grupo 1
( ) grupo 2

3) Qual a formação profissional (graduação)?
( ) Ciência da Computação
( ) Engenharia Elétrica
( ) Matemática
( ) Física
( ) outro _____

4) Qual o tempo de formado na graduação?
( ) menos de 1 ano
( ) de 1 à 3 anos
( ) de 3 à 5 anos
( ) de 5 à 10 anos
( ) acima de 10 anos

5) Qual a área de especialização?
( ) Inteligência Artificial
( ) Processamento de Imagens
( ) Engenharia de Software
( ) Computação Científica e Processamento de Alto Desempenho
( ) Outro_____

6) Quanto tempo está na área de Computação?
( ) menos de 1 ano
( ) de 1 à 3 anos
( ) de 3 à 5 anos
( ) de 5 à 10 anos
( ) acima de 10 anos

continua

continuação

7) Quanto tempo de experiência em desenvolvimento de sistemas?
( ) menos de 1 ano
( ) de 1 à 3 anos
( ) de 3 à 5 anos
( ) de 5 à 10 anos
( ) acima de 10 anos

8) Quanto tempo em programação?
( ) menos de 1 ano
( ) de 1 à 3 anos
( ) de 3 à 5 anos
( ) de 5 à 10 anos
( ) acima de 10 anos

9) Em quantos projetos de pequeno porte (de 1 até 20 pessoas), trabalhou?
( ) Apenas 1.
( ) de 1 à 3 projetos.
( ) de 3 à 5 projetos.
( ) de 5 à 10 projetos.
( ) acima de 10 projetos.

10) Em quantos projetos de médio porte (de 20 até 50 pessoas), trabalhou?
( ) Apenas 1.
( ) de 1 à 3 projetos.
( ) de 3 à 5 projetos.
( ) de 5 à 10 projetos.
( ) acima de 10 projetos.

11) Em quantos projetos de grande porte (mais de 50 pessoas), trabalhou?
( ) Apenas 1.
( ) de 1 à 3 projetos.
( ) de 3 à 5 projetos.
( ) de 5 à 10 projetos.
( ) acima de 10 projetos.

12) Há quanto tempo conhece metodologias para qualidade do processo de software?
( ) primeira experiência
( ) menos de 1 ano
( ) de 1 à 3 anos
( ) de 3 à 5 trabalhos
( ) acima de 5 anos

13) Há quanto tempo conhece metodologias para sistemas de descoberta de conhecimento em banco de dados?
( ) primeira experiência
( ) menos de 1 ano
( ) de 1 à 3 anos
( ) de 3 à 5 trabalhos
( ) acima de 5 anos

continua

continuação
14) Há quanto tempo conhece modelagem orientada a objetos?
( ) menos de 1 ano ( ) de 1 à 3 anos ( ) de 3 à 5 anos ( ) de 5 à 10 anos ( ) acima de 10 anos
15) Há quanto tempo conhece gerenciador de banco de dados? ( ) menos de 1 ano ( ) de 1 à 3 anos ( ) de 3 à 5 anos ( ) de 5 à 10 anos ( ) acima de 10 anos

Quadro C1 – Questionário para caracterização das equipes

Questionário 2 – aplicado em 23 de julho de 2007.

Descrição da Instrumentação

Verificação da presença relevante de Elementos-Chave, para o desenvolvimento de Soluções de Descoberta de Conhecimento em Banco de Dados (KDD[23]).

Esta pesquisa faz parte de uma dissertação de mestrado da Universidade Estadual de Maringá, Departamento de Informática.

Um dos propósitos do Desenvolvimento de Sistemas é oferecer ao usuário soluções informatizadas que facilitem, acima de tudo, o trabalho a ser desempenhado pelos mesmos. O domínio da aplicação é a primeira preocupação do desenvolvedor, e no caso de aplicações KDD esta tarefa torna-se complexa, visto a natureza do processo para tomada de decisões, em que na maioria das vezes envolve decisões sobre dinheiro, tempo e recursos da organização.
Sob o ponto de vista do Participante, do Estudo de Caso proposto, responda o questionário a seguir:
Caracterização do Participante:

Formação	
Curso:	
Experiência em Desenvolvimento de Sistemas	
	Excelente
	Alto
	Médio
	Baixo
	Nenhum
Membro do Grupo	
	Grupo 1 (Gerente Aldo)
	Grupo 2 (Gerente Francisco)

Quadro C2 – Questionário para formação dos participantes

[23] KDD sigla de *Knowledge Discovery in Database* ou Descoberta de Conhecimento em Banco de Dados.

APÊNDICE D

- QUESTIONÁRIO PARA CARACTERIZAÇÃO DO PROCESSO

Este Apêndice trata da caracterização do processo proposto, mostrado no Quadro D1.

Questionário – aplicado em 23 de julho de 2007.

Descrição da Instrumentação

Verificação da presença relevante de Elementos-Chave, para o desenvolvimento de Soluções de Descoberta de Conhecimento em Banco de Dados (KDD[24]).

Caracterização dos Elementos-Chave:

Este questionário visa verificar a importância dos Elementos-Chave no desenvolvimento de soluções KDD. Elemento-Chave entende-se por:

- Artefatos (porção significativa de informação produzida ou consumida por uma atividade),
- Papéis (responsabilidade ou competência individual para realizar uma atividade) e
- Atividades (descreve o que um papel executa no processo de engenharia de software, mostrando também o progresso da aplicação).

Considere que as colunas são independentes, portanto pede-se para responder todas as questões para todos os Elementos-Chave. Sob o ponto de vista do desenvolvedor de aplicações KDD, avalie e marque com um X as colunas correspondentes segundo as escalas abaixo:

Utilização significa a Aplicação do elemento-chave durante o desenvolvimento da solução KDD, considerando inclusive o conceito aplicado.	**Utilidade** significa que o elemento-chave em questão pode ser Eficaz, em situações específicas para a solução KDD.	**Adequação do nível de descrição** significa que determinado elemento-chave possui explicação Conveniente quanto a sua definição e uso.
Utilização	**Utilidade**	**Adequação do nível de descrição**
1. não usado e não gostaria de usar. 2. não usado, mas gostaria de usar. 3. usado.	1.não se demonstrou útil. 2.provavelmente é útil. 3.é útil.	4.a descrição deve ser aumentada. 5.a descrição deve ser diminuída 6.a descrição não precisa ser modificada.

N	Elementos-Chave	Descrição	Utilização			Utilidade			Adequação		
			1	2	3	1	2	3	1	2	3
	Artefatos										
1	Lista de Inferência	São as proposições/questões elaboradas a partir da base de dados transacional, a serem provadas pelas investigações na base de dados em modelo dimensional.									
2	Lista de Informações de Negócio	São os mapeamentos candidatos das proposições/questões de investigação com as informações de negócio, que fazem parte da base de dados transacional.									

[24] KDD sigla de *Knowledge Discovery in Database* ou Descoberta de Conhecimento em Banco de Dados.

N	Elementos-Chave	Descrição	Utilização			Utilidade			Adequação		
			1	2	3	1	2	3	1	2	3
3	Matriz de Barramento	É um modelo de arquitetura de DW[25] do tipo *Bus* (Barramento), que tem o propósito de separar as dimensões padrões (nas colunas) e fatos paralelos (nas linhas), para futuras inclusões de novos *data marts*[26].									
4	Modelo Dimensional	É um modelo lógico de dados, composto de uma tabela com chave múltipla (fato), e um conjunto de pequenas tabelas (dimensão).									
5	Descrição de Base(s) de Dados Existente(s)	É o resumo da estrutura de dados existentes, que tem por propósito entendimento dos dados do negócio, por exemplo, o destaque das principais tabelas e campos.									
6	Caracterização de Ferramentas Existentes	É o resumo das características dos softwares de KDD (por exemplo, Oracle Warehouse Builder, Discoverer, Weka), enfatizando a capacidade e limitação operacional, os requisitos mínimos para instalação entre outros.									
	Papéis										
7	Engenheiro de Conhecimento	Responsabilidade ou Competência que realiza aquisição de conhecimento a partir das informações de negócio (conhecimento do domínio) da empresa.									
8	Especialista de KDD	Responsabilidade ou Competência que realiza aplicações analíticas (mining ou olap), em resposta ao contexto da tomada de decisão.									
9	Usuário Final (Tomador de Decisão)	Responsabilidade ou Competência que realiza visualizações de informações extraídas, assim como determina quais motivações/necessidades de busca.									
	Atividades										
10	Analisar o Contexto de Dados	Realiza a ação de construir a lista de inferência para fundamentar e direcionar o processo KDD, a partir da base de dados transacional.									

[25] DW é a sigla usada para o *Data Warehouse*.
[26] *Data marts* implementa um determinado assunto, enquanto um DW implementa e integra vários assuntos da organização.

N	Elementos-Chave	Descrição	Utilização			Utilidade			Adequação		
			1	2	3	1	2	3	1	2	3
11	Analisar o Contexto de Ferramentas	Realiza a ação de caracterizar limitações e capacidades operacionais de softwares de KDD, com o intuito de orientar o desenvolvimento apenas para as funcionalidades não atendidas, para determinada solução.									
12	Gerar Informações de Negócio	Realiza a ação de construir os mapeamentos candidatos das proposições/questões de investigação, com as informações de negócio, que fazem parte da base de dados transacional.									
13	Projetar Arquitetura de DW	Realiza a ação de construir a matriz de barramento, assim como o *framework* arquitetural com representação para todos os níveis de detalhe (Requisitos de Negócio, Modelos e Documentos da Arquitetura, Modelos Detalhados e Implementação).									
14	Construir a Modelagem de Dados	Realiza a ação de construir o modelo dimensional para a solução específica de KDD.									

Quadro D1 – Questionário de caracterização do processo proposto

Obrigada por sua colaboração à nossa pesquisa.

APÊNDICE E

- QUESTIONÁRIO PARA CARACTERIZAÇÃO DO ESTUDO DE CASO

Este Apêndice mostra como foi caracterizada a realização do estudo de caso. O Quadro E1 mostra o questionário aplicado aos participantes da pesquisa.

Questionário – aplicado em 23 de julho de 2007.

Caracterização do Estudo de Caso:

Considere que o processo de desenvolvimento de software ou de uma solução define *quem* está fazendo o *que*, *quando* e *como* para alcançar certo objetivo.

Uma vez que, o Estudo de Caso proposto está finalizado, por favor, marque com X, em quais aspectos as dificuldades foram encontradas no decorrer do trabalho. Caso encontre alguma dificuldade que não tenha na lista abaixo, acrescente sua descrição no final da tabela.

Dificuldades Encontradas	**Incluir**
1. Tempo Insuficiente para o Desenvolvimento	
2. Modelagem de Sistema (UML)	
3. Equipe não Experiente	
4. Ferramentas Não Disponíveis	
5. Atribuição Inadequada de Atividades entre os Participantes	
6. Usuário Não Presente	
7. Processo KDD	
8. Ambiente de Desenvolvimento Inadequado	
9. Domínio da Aplicação	
10. Modelagem de Dados para Tomada de Decisão	
11. Equipe Dispersa Geograficamente	
12. Conflito entre Participantes	
13. Tempo Insuficiente para Entendimento do Contexto	
14. Participantes com Nível de Experiência Diferente	
15. Processo de Desenvolvimento Desconhecido	
16. Processo de Desenvolvimento Ausente	
17.	

Quadro E1 – Questionário das dificuldades encontradas

APÊNDICE F

- RESULTADO DO ESTUDO DE CASO - DADOS CRUS

Este Apêndice mostra os resultados alcançados pela pesquisa. Os dados foram trabalhos a fim de comprovar as hipóteses estatísticas para a realização do estudo de caso. Na Tabela F1 são mostrados dados dos participantes. Os dados das variáveis dependentes são mostrados nas Tabelas F2, F3 e F4. E os dados sobre as dificuldades estão na Tabela F5.

Questionário – aplicado em 23 de julho de 2007.

Tabulação da Caracterização do Participante

Tabela F1 – Dados dos participantes

		pessoas													0	1		
	Experiência em desenvolvimento de sistemas	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**		**1**	**2**	**3**	
1	Excelente														0	0	0	
2	Alto						3	3	3			3			0	0	4	
3	Medio	3		3		3				3	3		3		0	0	6	
4	Baixo		3		3										0	0	2	
5	Nenhum														0	0	0	
	Grupo1 - Aldo																	
	Alto	3																
	Medio	3																
	Baixo	0																
	Grupo2 - Francisco																	
	Alto	1																
	Medio	3																
	Baixo	2																
															0	1		
	Formacao	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**		**1**	**2**	**3**	
1	Informatica e Processamento de Dados	3		3			3	3	3	3		3	3		0	0	8	
2	Ciencia da Computacao		3			3					3				0	0	3	
3	Engenharia da Producao				3										0	0	1	
	Grupo1 - Aldo																	
	Informatica e Processamento de Dados	5																
	Ciencia da Computacao	1																
	Engenharia da Producao	0																
	Grupo2 - Francisco																	
	Informatica e Processamento de Dados	3																
	Ciencia da Computacao	2																
	Engenharia da Producao	1																

Tabulação da Caracterização dos Elementos-Chave

Tabela F2 – Dados da variável utilização

		pessoas												mediana	0		1					
	UTILIZAÇÃO	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**		**1**	**2**	**3**					
1	Lista de Inferência	3	2	3	3	3	2	3	3	2	3	3	3	**3**	0	3	9					
2	Lista de Informações de Negócio	3	3	3	3	3	3	3	3	2	3	3	3	**3**	0	1	11					
3	Matriz de Barramento	3	3	3	3	3	3	3	3	3	2	3	2	**3**	0	2	10					
4	Modelagem Dimensional	3	3	3	3	3	3	3	3	3	3	3	2	**3**	0	1	11					
5	Descrição de Base(s) de Dados Existente(s)	3	2	2	2	2	2	3	3	3	3	3	3	**3**	0	5	7					
6	Caracterização de Ferramentas Existentes	2	3	3	2	2	3	3	3	3	2	3	2	**3**	0	5	7					
7	Engenheiro de Conhecimento	3	2	2	3	2	2	3	3	2	3	3	3	**3**	0	5	7					
8	Especialista de KDD	3	2	3	3	2	2	2	2	2	3	3	3	**2,5**	0	**6**	**6**					
9	Usuário Final (Tomador de Decisão)	3	3	2	2	2	2	3	3	2	3	3	3	**3**	0	5	7					
10	Analisar o Contexto de Dados	3	3	3	3	2	3	3	3	3	2	3	3	**3**	0	2	10					
11	Analisar o Contexto de Ferramentas	3	2	3	2	2	2	3	3	3	2	3	3	**3**	0	5	7					
12	Gerar Informações de Negócio	3	2	3	2	3	2	3	3	3	2	3	3	**3**	0	4	8					
13	Projetar Arquitetura de DW	3	2	3	3	3	2	3	3	2	3	3	3	**3**	0	3	9					
14	Construir a Modelagem de Dados	3	3	2	3	3	3	3	3	3	3	3	3	**3**	0	1	11					
																				Continua		

Tabela F3 – Dados da variável utilidade

		pessoas												mediana	0	1					
	UTILIDADE	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**		**1**	**2**	**3**				
1	Lista de Inferência	3	2	2	3	3	2	3	3	2	3	3	3	**3**	0	4	8				
2	Lista de Informações de Negócio	3	3	3	3	3	3	3	3	2	2	3	3	**3**	0	2	10				
3	Matriz de Barramento	3	3	3	3	3	3	3	1	3	2	3	2	**3**	**1**	2	9				
4	Modelagem Dimensional	3	3	3	3	3	3	3	3	2	3	3	2	**3**	0	2	10				
5	Descrição de Base(s) de Dados Existente(s)	3	3	3	2	3	3	3	3	1	3	3	3	**3**	**1**	1	10				
6	Caracterização de Ferramentas Existentes	2	3	2	2	3	2	3	3	2	2	3	3	**2,5**	0	6	6				
7	Engenheiro de Conhecimento	3	3	2	3	3	3	3	3	1	3	3	3	**3**	**1**	1	10				
8	Especialista de KDD	3	3	3	3	3	3	3	3	1	3	3	3	**3**	**1**	0	11				
9	Usuário Final (Tomador de Decisão)	3	3	3	2	3	3	3	1	2	3	3	3	**3**	**1**	2	9				
10	Analisar o Contexto de Dados	3	3	3	3	3	3	3	3	2	2	3	3	**3**	0	2	10				
11	Analisar o Contexto de Ferramentas	3	3	3	2	3	2	3	3	3	2	3	3	**3**	0	3	9				
12	Gerar Informações de Negócio	3	3	3	2	3	3	2	3	3	3	3	3	**3**	0	2	10				
13	Projetar Arquitetura de DW	3	3	2	3	3	3	2	3	3	3	3	3	**3**	0	2	10				
14	Construir a Modelagem de Dados	3	3	2	3	3	3	2	3	3	3	3	3	**3**	0	2	10				
																				continua	

Tabela F4 – Dados da variável adequação

		pessoas												mediana	1		0						
	ADEQUAÇÃO	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**		**1**	**2**	**3**						
1	Lista de Inferência	3	3	1	3	1	3	3	1	1	3	3	3	**3**	4	0	8						
2	Lista de Informações de Negócio	3	1	3	1	3	1	3	1	1	2	3	3	**2,5**	5	1	6						
3	Matriz de Barramento	3	3	3	2	1	3	3	3	1	3	3	1	**3**	**3**	**1**	8						
4	Modelagem Dimensional	3	1	3	2	1	1	3	1	1	3	3	3	**2,5**	5	1	6						
5	Descrição de Base(s) de Dados Existente(s)	3	3	1	1	3	3	3	3	1	3	3	3	**3**	**3**	**0**	9						
6	Caracterização de Ferramentas Existentes	1	3	1	1	1	3	3	3	1	3	1	1	**1**	7	0	5						
7	Engenheiro de Conhecimento	3	3	3	1	3	3	3	3	3	1	1	1	**3**	**4**	**0**	8						
8	Especialista de KDD	3	3	3	1	3	3	3	3	1	3	3	1	**3**	**3**	**0**	9						
9	Usuário Final (Tomador de Decisão)	3	3	3	1	2	3	3	3	3	3	1	3	**3**	**2**	**1**	9						
10	Analisar o Contexto de Dados	3	3	3	1	3	3	3	3	1	3	1	1	**3**	4	0	8						
11	Analisar o Contexto de Ferramentas	1	3	3	1	3	3	3	3	1	3	1	3	**3**	4	0	8						
12	Gerar Informações de Negócio	3	3	1	1	3	3	1	3	3	3	3	3	**3**	3	0	9						
13	Projetar Arquitetura de DW	1	3	1	1	3	3	1	1	3	3	3	1	**2**	6	0	6						
14	Construir a Modelagem de Dados	1	3	1	1	3	3	1	1	3	3	3	3	**3**	5	0	7						

Tabulação da Caracterização do Estudo de Caso

As respostas com valores igual a 3 contam como Dificuldade Presente (valendo 1). Para as respostas com valores igual a 1 contam como Dificuldade Não Presente (valendo 0).

Tabela F5 – Dados das dificuldades encontradas

		pessoas												0	1
	DIFICULDADES ENCONTRADAS	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**	**3**
1	Tempo Insuficiente para o Desenvolvimento	3	3	3	3	3	3	1	1	3	1	3	3	3	0	9
2	Modelagem de Sistema (UML)	3	1	1	1	1	1	3	3	3	1	3	1	7	0	5
3	Equipe não Experiente	3	3	1	3	3	3	1	1	3	1	3	1	5	0	7
4	Ferramentas Não Disponíveis	3	3	3	3	1	3	1	3	3	3	3	3	2	0	10
5	Atribuição Inadequada de Atividades entre os Participantes	1	1	1	1	1	1	1	1	1	1	3	1	11	0	1
6	Usuário Não Presente	1	3	3	3	1	3	1	1	3	1	3	1	6	0	6
7	Uso do Processo KDD	3	1	3	1	1	1	1	3	3	1	3	1	7	0	5
8	Ambiente de Desenvolvimento Inadequado	1	3	3	1	1	1	1	1	1	1	3	1	9	0	3
9	Domínio da Aplicação	1	3	3	3	3	1	3	3	3	1	3	3	3	0	9
10	Modelagem de Dados para Tomada de Decisão	1	1	1	1	1	1	3	1	3	3	3	1	8	0	4
11	Equipe Dispersa Geograficamente	3	1	1	1	1	1	1	3	1	3	3	3	7	0	5
12	Conflito entre Participantes	1	1	1	1	1	1	1	3	1	1	3	1	10	0	2
13	Tempo Insuficiente para Entendimento do Contexto	3	1	1	1	1	1	1	1	3	1	3	3	8	0	4
14	Participantes com Nível de Experiência Diferente	1	3	1	3	1	3	1	3	3	1	3	1	6	0	6
15	Processo de Desenvolvimento Desconhecido	3	3	1	3	1	3	1	1	3	1	3	1	6	0	6
16	Processo de Desenvolvimento Ausente	1	1	1	3	1	3	1	1	1	1	1	1	10	0	2
17	Excesso no Retardo para saber o objetivo	3												0	0	1
18	Falta de Experiencia no uso da Ferramenta		3			3	3							0	0	3
19	Material de Apoio para uso de Ferramentas		3		3		3							0	0	3
20	Bases de Dados de Origem Incompativel com as Ferramentas		3		3									0	0	2
21	Base de Dados da Origem não contem informacoes suficientes para TD		3		3									0	0	2
22	No Inicio o grupo não entendeu o que era para fazer							3						0	0	1
23	Assimilacao de Alguns Conceitos								3					0	0	1
24	Demora para definicao da Base/Objetivo								3					0	0	1
		Sim	**Não**	**%**												
	Ferramentas Não Disponíveis	10	2	83												
	Tempo Insuficiente para o Desenvolvimento	9	3	75												
	Domínio da Aplicação	9	3	75												
	Equipe não Experiente	7	5	58												
	Usuário Não Presente	6	6	50												
	Participantes com Nível de Experiência Diferente	6	6	50												
	Processo de Dsv Desconhecido	6	6	50												

ANEXO I

- ARTEFATO (LISTA DE INFERÊNCIA)

Grupo 1 – (com treinamento)

INFERÊNCIA REFERENTE AO OLAP

1. Qual é o convênio mais utilizado dentro de um determinado período?

2. De qual localidade os meus pacientes estão vindo?

3. Qual é a percentagem de resultados positivos para o exame de TIG (Teste Imunológico de Gravidez) dentro de determinada faixa etária?

4. Qual é o perfil (sexo, estado civil, localidade, idade e período do ano em que ocorre com maior freqüência) do cliente que realiza um determinado exame?

5. Qual é o médico que mais encaminha pacientes para este laboratório?

6. Qual é o percentual de exames enviados para a internet dentro de um determinado período?

7. Qual a porcentagem de pacientes de sexo masculino/Feminino?

8. Qual setor é o maior realizador de número de exames?

9. Qual a média de retorno de paciente que tenham a necessidade de realizar exames periodicamente (por exemplo: glicose, colesterol...)?

10. Quais exames mais requisitados? Quais exames que dão maior volume no faturamento?

11. Qual o número de exames enviados para laboratório de apoio?

12. Verificar quantidade de exames efetuados por alguns grupos de usuários, tais como, geriátricos, pediátricos, masculino ou feminino estes e/ou outros grupos?

13.Comparar a média diária de exames realizados ente o período de um determinado ano e o mesmo período de outro ano imediatamente posterior. Analisar ao menos 3 anos consecutivos.

INFERÊNCIA REFERENTE À MINERAÇÃO DE DADOS

1. Por que determinado convênio é mais utilizado dentro de um determinado período?
2. Por que os meus pacientes estão vindo de determinada localidade?
3. Por que a percentagem de resultados positivos para o exame de TIG (Teste Imunológico de Gravidez) é maior dentro de determinada faixa etária?
4. Por que o cliente de determinado perfil (sexo, estado civil, localidade, idade e período do ano em que ocorre com maior freqüência) realiza um determinado exame?
5. Por que determinado o médico encaminha pacientes para este laboratório?
6. Por que determinado setor é o maior realizador de exames?
7. Por que alguns pacientes realizam alguns exames periodicamente? Quais exames?
8. Por que alguns exames são mais requisitados?
9. Por que alguns exames tiveram sua procura aumentada e outros diminuídos ente o período de um determinado ano e o mesmo período de outro ano imediatamente posterior. Analisar ao menos 3 anos consecutivos.

Grupo 2 – (sem treinamento)

A *lista de Inferência* foi chamada pelo Grupo 2 de Identificação das Necessidades de Informação no contexto do Tomador de Decisão.

Porém o artefato construído tem o mesmo objetivo.

IDENTIFICAÇÃO DAS NECESSIDADES DE INFORMAÇÃO NO CONTEXTO DO TOMADOR DE DECISÃO

Considerando o contexto do Tomador de Decisão, o grupo elencou diversas informações a serem buscadas:

a) Com o intuito de identificar a necessidade de aumento de recursos como equipamentos, materiais e pessoal, o grupo entendeu que um levantamento estatístico dos exames mais realizados pode ser útil;

b) Para identificar a necessidade de aumento ou redução de recursos, o grupo concluiu que é necessário um levantamento do tempo médio de entrega dos exames;

c) Para se verificar a necessidade de abertura de filiais ou postos de coleta mais próximos dos pacientes, o grupo definiu que estatísticas dos clientes por localidade de domicílio devem ser feitas;

d) Com o objetivo de se identificar a oportunidade de realização interna dos exames atualmente terceirizados, foi proposto o levantamento estatístico de tais exames;

e) Verificou-se a necessidade de se identificar quais formas de pagamento são mais utilizadas com o propósito de se avaliar a necessidade de trabalhar com novos convênios;

f) Afim de avaliar a necessidade de aumento ou redução de recursos, especialmente de *kits* para exame, se propôs um levantamento estatístico relacionando tipos de exames e períodos nos quais são solicitados, podendo ser dias, semanas, meses ou estações do ano;

g) A verificação da quantidade de pacientes que utilizam dos serviços prestados pelo laboratório, em função dos médicos que solicitam os exames, pode apontar quais

consultórios merecem maior atenção e quais devem ser alvo de trabalho que objetive conquistar mais clientes;

h) A identificação do perfil dos pacientes por sexo pode servir como base para a personalização do atendimento;

i) A identificação do perfil dos pacientes por faixas de idade pode apontar a necessidade de que um atendimento mais adequado para determinadas faixas etárias;

j) A baixa taxa de retorno dos pacientes pode ser um indicativo de possíveis perdas de clientes;

k) A baixa taxa de novos pacientes pode indicar perda de mercado, inclusive pela incorporação de novos tipos de exames ainda não realizados pelo laboratório.

ANEXO II

- ARTEFATO (DESCRIÇÃO DE BASE DE DADOS EXISTENTE)

Grupo 1 – (com treinamento)

DESCRIÇÃO DA BASE DE DADOS EXISTENTE

A base de dados existente armazena os dados de laboratório de análises clínicas. Possui 31 entidades. A tabela mais importante é a REQUISIÇÃO, através desta é possível fazer todos os levantamentos de dados existentes. O SGBD é o access da microsoft e se encontra em uma versão 2.0. Este é um SGBD obsoleto, lançado para a arquitetura de 16 bits e que funciona normalmente em sistemas operacionais de 32 bits da microsoft, devido a compatibilidade preservada. Esta base ocupa um tamanho pouco maior que 100 MB. Dentre os relacionamentos mais importantes encontra-se:

Tabela requisição. Possui uma entrada para cada atendimento. Gerando assim um número de protocolo;

Tabela RolExames. Possui uma entrada para cada exame da tabela de requisição;

Tabela ResultadoExame. Possui uma entrada para cada atributo de RolExames;

Tabela TipoExame. Possui uma entrada para cada entrada na RolExames. Esta tabela possui a descrição e características dos exames realizados.

Grupo 2 – (sem treinamento)

A *Descrição de Base de Dados Existente* foi chamada pelo Grupo 2 de Análise dos Dados Existentes

Porém o artefato construído tem o mesmo objetivo.

ANÁLISE DOS DADOS EXISTENTES

A partir do modelo apresentado e dos arquivos disponibilizados, o grupo analisou os dados existentes na base afim de identificar as informações que poderiam ser buscadas pelas ferramentas Weka e Discoverer.

Verificou-se que existem trinta tabelas. Dentre elas a que mais se destaca é a tabela Requisição. Pela análise feita, a requisição tende a ser o ponto de partida para descoberta de conhecimentos interessantes ao Tomador de Decisão. Com esse intuito foram levantadas as necessidades.

ANEXO III

- ARTEFATO (CARACTERIZAÇÃO DE FERRAMENTAS EXISTENTES)

Grupo 1 – (com treinamento)

CARACTERIZAÇÃO DAS FERRAMENTAS EXISTENTES

1 WEKA

O sistema Weka é um *software* livre (de código aberto) para mineração de dados, desenvolvido em Java, dentro das especificações da GNU (*General Public License*). As suas características, bem como as técnicas nele implementadas são descritas de forma detalhada em [Witten e Frank 2005], cujos autores são os responsáveis pela implementação da ferramenta. O *software* está disponível para *Windows*, *Linux* e outras plataformas.

A ferramenta Weka possui como ponto forte a extração de **classificadores** em bases de dados. Um classificador (ou **modelo de classificação**) é utilizado para identificar a classe à qual pertence uma determinada observação de uma base de dados, a partir de suas características (seus atributos).

A ferramenta Weka trabalha com arquivos de entrada no formato ARFF, que corresponde a um arquivo texto contendo um conjunto de observações, precedido por um pequeno cabeçalho. O cabeçalho é utilizado para fornecer informações a respeito dos campos que compõem o conjunto de observações. Dessa forma, antes da mineração de dados, a ferramenta pode verificar alguma inconsistência na base de dados e sinalizá-la. A Figura 1 ilustra um exemplo de arquivo ARFF, contendo um cabeçalho e um conjunto de 8 registros que representam a base de dados apresentada na Tabela 1. Observe que o cabeçalho contém a declaração da relação que o arquivo representa (comando @relation), uma lista de atributos (comando @attribute) e a relação de valores que os mesmos podem assumir. O conjunto de observações é precedido por um comando @data. Cada observação é representada por uma linha. Os valores dos campos dentro de uma observação devem ser separados utilizando a vírgula.

Tabela 1 Base de Dados Censitários

NOME	ESCOLARIDADE	IDADE	RICO (*atributo classe*)
Alva	Mestrado	>30	Sim
Amanda	Doutorado	<=30	Sim
Ana	Mestrado	<=30	Não
Eduardo	Doutorado	>30	Sim
Inês	Graduação	<=30	Não
Joaquim	Graduação	>30	Não
Maria	Mestrado	>30	Sim
Raphael	Mestrado	<=30	Não

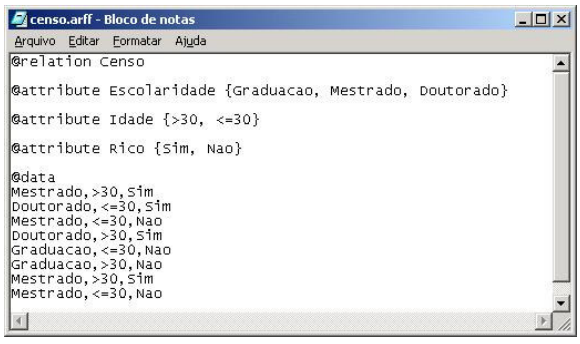


Figura 1 Arquivo ARFF.

O instalador da ferramenta Weka pode ser obtido de maneira gratuita (juntamente com seu código fonte) no site http://www.cs.waikato.ac.nz/~ml/weka. Uma vez instalado, o sistema Weka pode ser utilizado para minerar árvores de decisão através da execução dos seguintes passos:

PASSO 1: Executar o programa. A partir do menu Iniciar / Programas, selecione WEKA e clique em Weka 3-4 (versão atual do sistema). O menu principal Weka GUI Chooser será exibido na tela. Clique no botão "Explorer" (Figura 2).

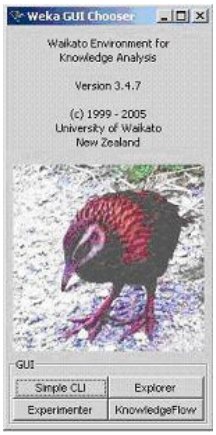


Figura 2 Weka GUI Chooser

PASSO 2: Importar o arquivo ARFF. Após iniciar o Weka Explorer, a opção "*Open File*" deve ser utilizada para abrir o arquivo ARFF que será minerado.

PASSO 3: Selecionar os Atributos. Em seguida, o Weka abrirá uma tela que permite com que o usuário possa definir qual o atributo da base que será utilizado como classe e quais os atributos que serão utilizados como preditivos (Figura 3). No momento da importação, por *default*, o Weka irá considerar o **último atributo** especificado no cabeçalho do arquivo ARFF, como o atributo classe, enquanto os **demais atributos** serão tratados como atributos preditivos. Observe que, nesta tela (aba **Preprocess**), também é possível consultar gráficos de barra que indicam os cruzamentos de freqüência envolvendo todos os atributos preditivos e o atributo classe.

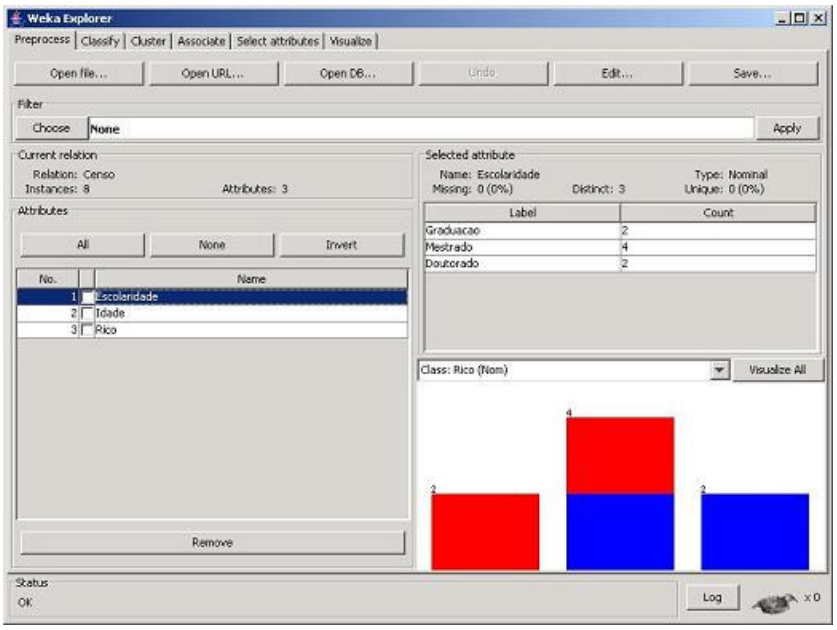


Figura 3 Seleção da Classe e dos Atributos Preditivos

PASSO 4: Selecionar o Algoritmo de Mineração. Clique na aba "*Classify*". A partir desta tela é possível escolher e executar um algoritmo de classificação sobre a base de dados importada. Os resultados da mineração também poderão ser consultados neste mesmo local. Clique no botão "*Choose*". Será aberta uma janela que permitirá a escolha do algoritmo de mineração de dados. Clique na pasta "*trees*" (algoritmos de árvore de decisão) e selecione a opção "*Id3*" (Figura 4).

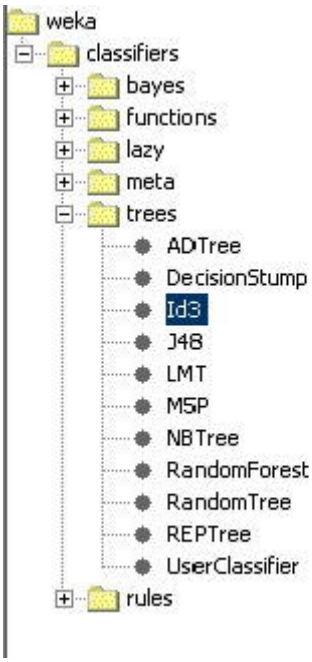


Figura 4 Seleção do Algoritmo de Mineração de Dados

PASSO 5: Executar o Algoritmo de Mineração. No painel "*Test options*" selecione a opção "*Use training set*". Esta seleção indica ao Weka que toda a base de dados será utilizada como base de treinamento durante o processo de mineração. A seguir clique no botão "*Start*". A árvore de decisão gerada pelo algoritmo ID3 é apresentada no canto direito da tela do Weka, conforme ilustra a área destacada no círculo vermelho da Figura 5. Na mesma tela são apresentadas algumas medidas de interesse que indicam a qualidade da árvore minerada.

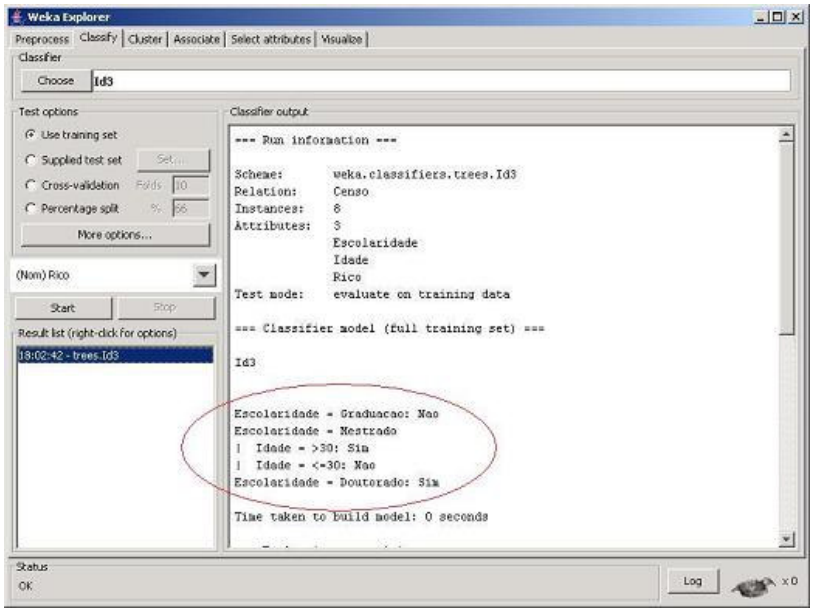


Figura 5 Árvore de Decisão Minerada pelo Weka

Além disso, existem outras capacidades do sistema Weka, como a mineração de regras de associação e *clusters* de dados e a obtenção de modelos de classificação através de outros algoritmos diferentes do ID3.

2 Oracle Warehouse Builder

O Oracle Warehouse Builder é uma ferramenta que faz parte do conjunto de soluções para banco de dados da empresa Oracle. Este ambiente permite o acesso a bases de dados, a definição de transformações, a criação de novas bases de dados e à operação com bases de dados multidimensionais. Possui uma interface gráfica bastante elaborada, com recursos avançados para edição visual do modelo de DW construído. Todos os metadados sobre o modelo são armazenados em um repositório de metadados, que possui um controle de acesso por usuários. Todo o ambiente é integrado com o banco de dados Oracle. O ambiente possui extensões para carga de dados em outras bases de dados não Oracle e arquivos textos.

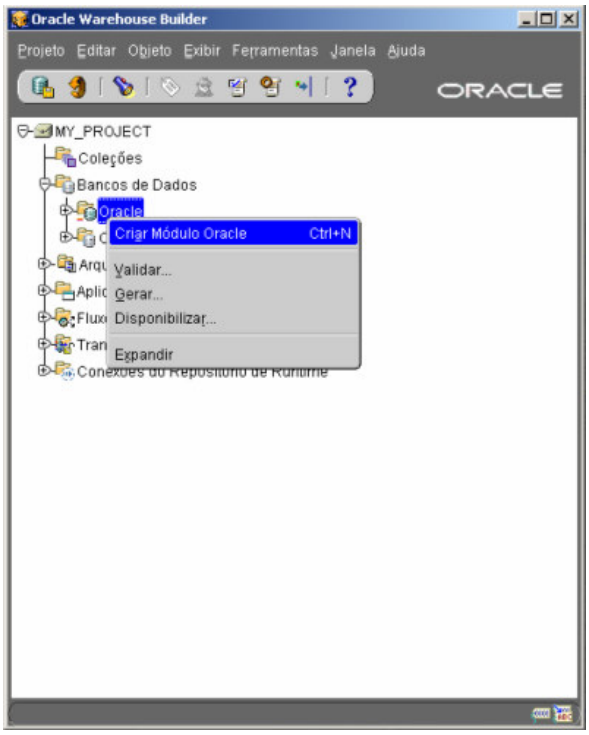


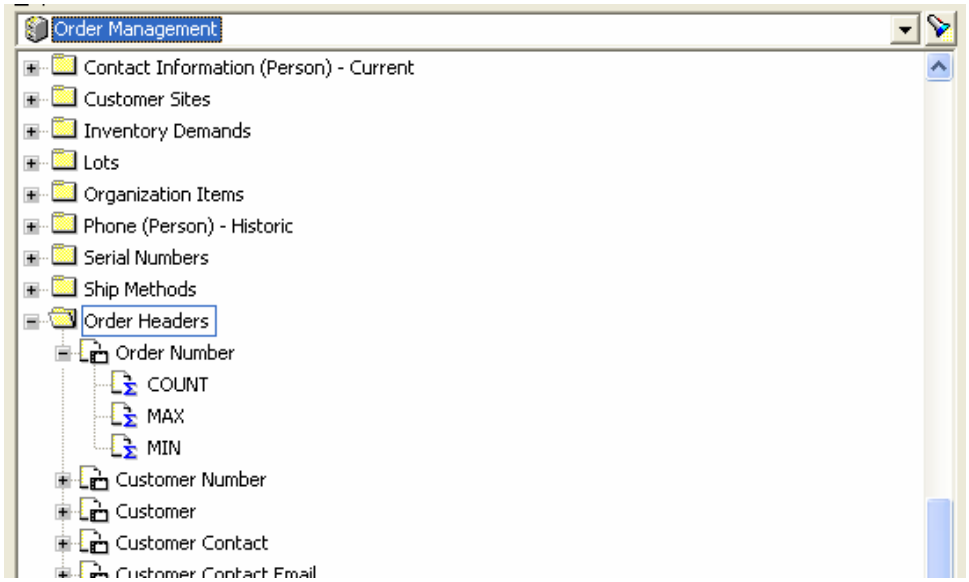
Figura 1: Tela inicial do Warehouse Builder

3 Oracle Discoverer

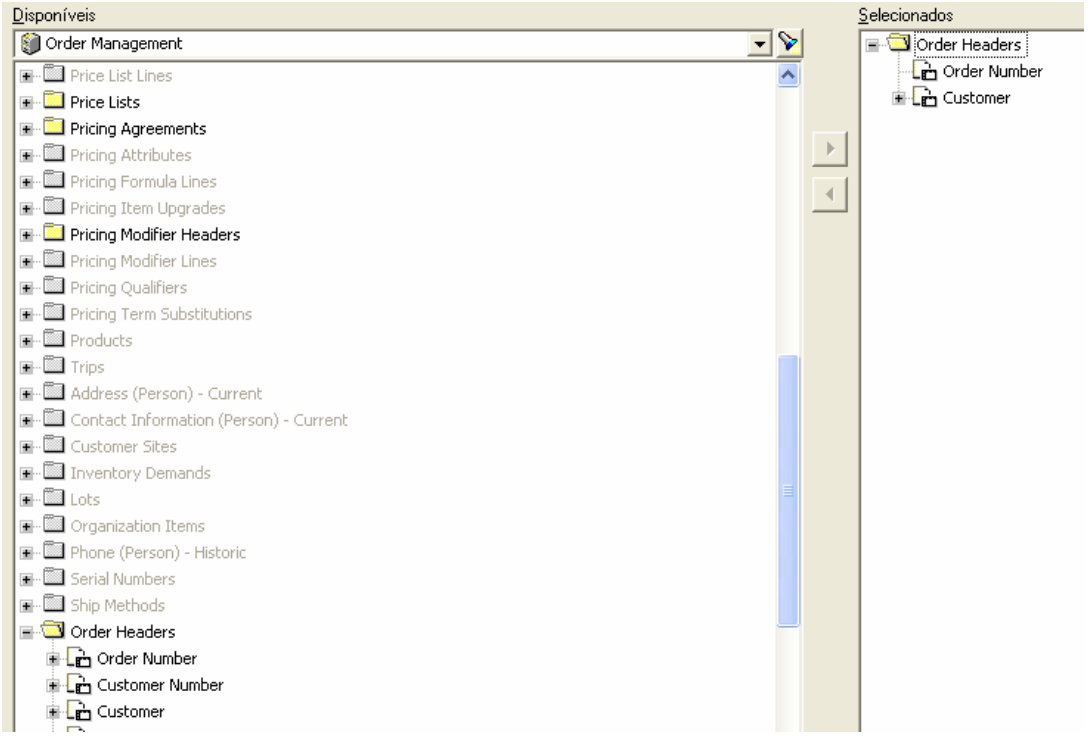
O Oracle Discoverer é uma ferramenta de BI (Business Intelligence) para apoio à tomada de decisão empresarial. Utiliza um repositório próprio, chamado de EUL (End User Layer), que têm como origem as informações armazenadas e mantidas por aplicações transacionais. A EUL tem como principal objetivo ocultar a complexidade e os detalhes do banco de dados transacional, organizando os dados de modo a refletir as áreas de negócio específicas da empresa, facilitando e agilizando as consultas.

O Oracle E-Business Suite (EBS), também chamado de Oracle Applications ou Oracle Financials, através do módulo de BI, possui um mapeamento pré-configurado de suas bases de dados transacionais para uma EUL própria. Desta maneira o Oracle Discoverer pode ser usado como ferramenta de suporte à decisão sobre as informações armazenadas no Oracle EBS.

Usando o mesmo esquema de segurança do Oracle EBS, definido através de responsabilidades que controlam os direitos de acesso, o Oracle Discoverer apresenta a estrutura mapeada das tabelas transacionais dos aplicativos EBS em forma amigável, com pastas específicas por área de negócio que listam os itens disponíveis para consulta. Na figura abaixo está um exemplo onde são listadas as pastas do módulo de gerenciamento de ordens de venda (Order Management):



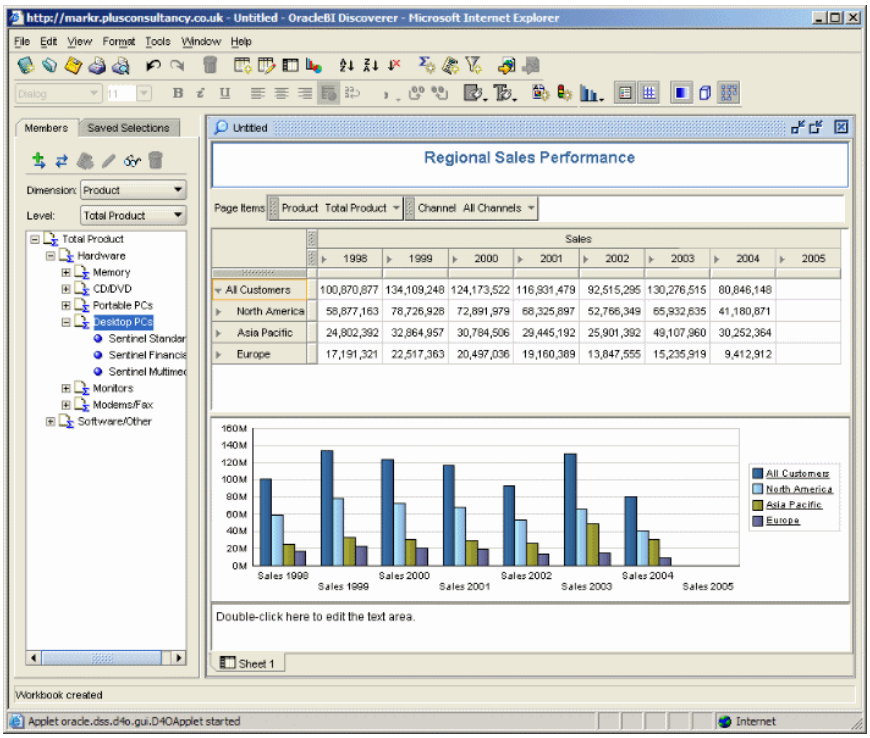
Como todos os relacionamentos e dependências entre os itens já estão mapeados, a partir da seleção de um item o Oracle Discoverer deixará disponível para seleção somente os outros itens com algum relacionamento com o primeiro selecionado. Essa característica pode ser observada na figura abaixo, onde os itens que não são passíveis de seleção (que não têm relacionamento com "Order Headers") estão com as pastas na cor cinza:

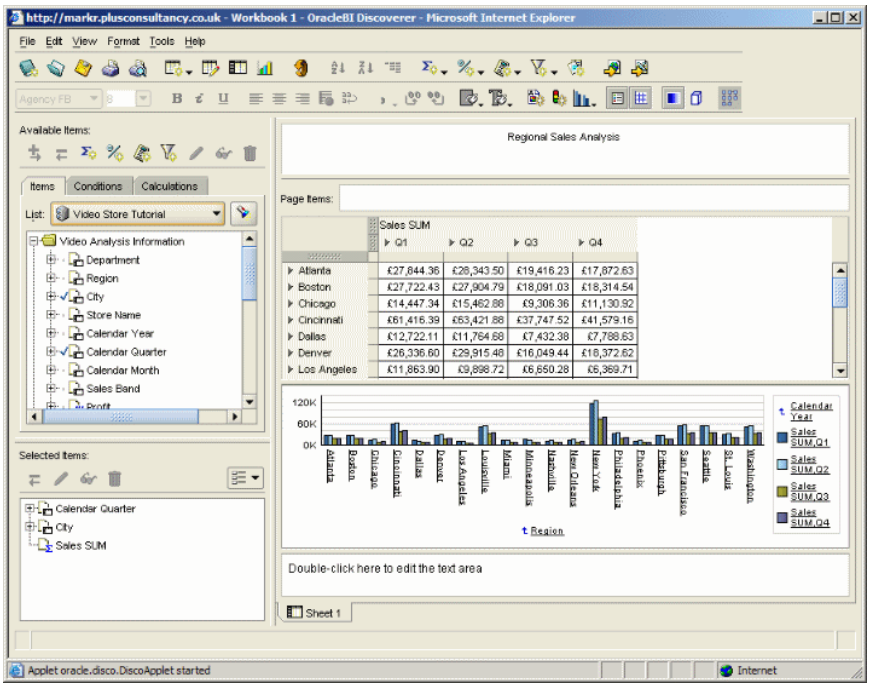


As consultas realizadas no Oracle Discoverer podem ser exibidas em formato de tabela ou matriz, permitindo a definição de campos de seleção de página para o filtro dos dados apresentados. Várias facilidades de formatação de campos e modificação de layout das consultas estão disponíveis, permitindo assim, que formatações condicionais de fontes e cores sejam definidas para cada uma das colunas consultadas. Além disso, existe a possibilidade de movimentação, ordenação e agrupamento das colunas de dados.

O Oracle Discoverer oferece a possibilidade de realização das consultas via um aplicativo instalado localmente nos computadores (Oracle Discoverer Desktop Edition) ou via browser (Oracle Discoverer Plus), além de permitir a exportação dos relatórios formatados para arquivos Html e para o Microsoft Excel, além de outros formatos possíveis.

Por estas características o Oracle Discoverer, além de outras opções como Oracle Reports e XML Publisher, é uma ótima alternativa para o desenvolvimento de relatórios e consultas customizadas para o Oracle EBS. Se sua empresa tem a necessidade de criação de relatórios com requisitos específicos utilizando alguma destas ferramentas, entre em contato conosco, pois nossa fábrica de software, especializada na customização de relatórios para o Oracle E-Business Suite, pode oferecer as soluções mais adequadas às suas necessidades.





Grupo 2 – (sem treinamento)

```
Nota-se que houve preocupação com ferramentas existentes como
ERwin, OWB, Postgres, Weka.
```

DESENVOLVIMENTO DAS ATIVIDADES

Definiu-se que a base de dados a ser explorada seria a de um laboratório de exames médicos. Os arquivos de tal base, originalmente no Access 2, foram exportados em formato de texto, extensão .txt. O Modelo Relacional foi disponibilizado em arquivo do ERWin. Uma breve apresentação foi feita pelo fornecedor da base.

Para que o trabalho pudesse ser executado, foi solicitada a instalação do banco de dados Oracle e também do Oracle Warehouse Builder (OWB) numa máquina do laboratório do Mestrado em Ciência da Computação.

A instalação dos programas somente foi efetuada no início do mês de maio. Porém, problemas foram detectados nas tentativas de utilização do OWB. Tais dificuldades foram repassadas. Somente no início do mês de junho a equipe teve de fato a ferramenta à sua disposição juntamente com uma senha que concedia direitos de administrador, exigidos pela ferramenta.

Paralelamente às tentativas de utilização do OWB, a equipe verificou que seria necessário carregar os arquivos em um outro banco de dados, já que o formato original não era compatível com o OWB e nem com a ferramenta Weka. Assim, os arquivos, um a um, foram carregados no Postgres através de programas implementados em Java.

ANEXO IV

- ARTEFATO (LISTA DE INFORMAÇÕES DE NEGÓCIO)

Grupo 1 – (com treinamento)

LISTA DE INFORMAÇÕES DE NEGÓCIO

A motivação pela investigação na base dados existente pode ser representada pela necessidade de maior suporte à tomada de decisões de caráter financeiro, gerencial e expansão do negócio. Para isto, serão investigadas características do perfil dos pacientes e exames efetuados, além do faturamento agregado a tais características em determinado período.

1. Necessidade ou Motivação do Tomador de Decisão:
Necessita de apoio para determinar onde será o novo posto de atendimento. Desta maneira, ficar mais próximo a estes pacientes, facilitando o acesso ao laboratório. Esta resposta também facilitará o marketing direcionado a determinado perfil.

1.1 Inferências Propostas:
1.1.1 De qual localidade os pacientes estão vindo.
1.1.2 Qual é o perfil (sexo, estado civil, localidade, idade e período do ano em que ocorre com maior freqüência) do cliente que realiza um determinado exame?

1.2 Informações de Negócio:
A investigação será em dados cadastrais dos endereços dos clientes versus a data da requisição e tipo de exame.

Tabela	Campo
Paciente	Bairro
Paciente	Cep
Paciente	Cidade
Paciente	UF
Paciente	DataNascimento
Paciente	Sexo
Requisição	DataRequisição
Requisição	IdadeInformada
RolExames	CodExame
TipoExame	CodExame
TipoExame	Nome

2. Necessidade ou Motivação do Tomador de Decisão:
Aumentar o faturamento, criando promoções com gratificações para os médicos mais enviar pacientes ao laboratório.

2.1 Inferências Propostas:
2.1.1 Quais médicos mais encaminham pacientes ao laboratório.

2.2 Informações de Negócio:
A investigação será nos dados da requisição do exame.

Tabela	Campo
Requisição	DataRequisição
Requisição	CodMedico

3. Necessidade ou Motivação do Tomador de Decisão:
Melhorar o faturamento através da fidelização de clientes que necessitam realizar exames periodicamente através de promoções ou campanhas para grupos específicos de pacientes.

3.1 Inferências Propostas:
3.1.1 Qual a média de retorno de pacientes que tenham a necessidade de realizar exames periodicamente (por exemplo: glicose, colesterol...).
3.1.2 Verificar quantidade de exames efetuados por alguns grupos de usuários, tais como, geriátricos, pediátricos, masculino ou feminino dentre outros grupos.

3.2 Informações de Negócio:
A investigação será nos dados do paciente, exame e requisição.

Tabela	Campo
Paciente	NumPaciente
Paciente	DataNascimento
RolExames	CodExame
Requisição	DataRequisicao
Requisição	Sexo

4. Necessidade ou Motivação do Tomador de Decisão:
Abrir contratação de funcionário de acordo com o sexo dos pacientes em sua devida proporção, pois, há exames em que o paciente pode ficar constrangido caso o funcionário que estiver passando as instruções ou efetuando coleta seja de sexo oposto.

4.1 Inferências Propostas:
4.1.1 Qual a porcentagem de pacientes de sexo masculino/feminino.

4.2 Informações de Negócio:
A investigação será nos dados do paciente e requisição.

Tabela	Campo
Paciente	Sexo
Requisição	DataRequisicao

5. Necessidade ou Motivação do Tomador de Decisão:
Devo adquirir equipamentos e kits para realizar exames que hoje são enviados a outros laboratórios, portanto, deixando de realizar exames que são rentáveis, devido a imaginar que o volume não seja o suficiente para a aquisição dos kits.

5.1 Inferências Propostas:
5.1.1 Quais exames e suas quantidades são enviados para laboratórios de apoio.

5.2 Informações de Negócio:
A investigação será nos dados do exame, requisição e laboratório de apoio.

Tabela	Campo
LabApoio	NomeApoio
TipoExame	Nome
Requisição	DataRequisicao

6. Necessidade ou Motivação do Tomador de Decisão:
Para alguns exames sazonais, devo adquirir quantidades diferenciadas de materiais e efetuar contratações para períodos de pico.

6.1 Inferências Propostas:
6.1.1 Comparar a média diária de exames realizados entre o período de um determinado ano e o mesmo período de outro ano imediatamente posterior. Analisar ao menos 3 anos consecutivos.

6.2 Informações de Negócio:
A investigação será nos dados do exame, requisição e paciente.

Tabela	Campo
Paciente	DataNascimento
Requisição	DataRequisicao
Requisição	Sexo
RolExames	CodExame

Grupo 2 – (sem treinamento)

A *Lista de Informações de Negócio* foi chamada pelo Grupo 2 de Informações a serem levantadas.

Porém o artefato foi construído com o mesmo objetivo.

A partir das informações relacionadas, foi elaborado o Quadro 1.

Quadro 1: Informações a serem levantadas

	DESCRIÇÃO DAS INFORMAÇÕES A SEREM LEVANTADAS	OBJETIVOS
1	Estatísticas de exames mais realizados.	Identificar a necessidade de aumento de recursos (equipamentos, materiais e pessoal)
2	Tempo médio de entrega dos exames.	Identificar a necessidade de aumento ou redução de recursos (equipamentos, materiais e pessoal)
3	Estatísticas dos clientes por localidade onde eles residem.	Identificar a necessidade de abertura de filiais ou postos de coleta.
4	Estatísticas de exames mais terceirizados.	Identificar a oportunidade ou não da realização interna dos exames.
5	Estatísticas de qual forma de pagamento (convênio, particular, SUS) é mais realizada.	Identificar a viabilidade de novos convênios.
6	Estatísticas relacionando tipos de exames e períodos nos quais são solicitados em dias, semanas, meses, estações do ano, meses.	Identificar a necessidade de aumento ou redução de recursos (kits para exames)
7	Estatísticas relacionando médicos e requisições.	Identificar que médicos solicitam mais ou menos exames
8	Estatísticas por sexo.	Identificar o perfil dos pacientes por sexo
9	Estatísticas por faixa etária.	Identificar o perfil dos pacientes por idade
10	Estatísticas da taxa de retorno de antigos pacientes.	Identificar possíveis perdas de pacientes
11	Estatísticas da taxa de chegada de novos pacientes.	Identificar possíveis novos tipos de exames e possível perda de espaço no mercado

ANEXO V

- ARTEFATO (MATRIZ DE BARRAMENTO)

Grupo 1 – (com treinamento)

MATRIZ DE BARRAMENTO

Processos de negócio	Paciente	GrupoConvenio	TipoExame	Tempo	LabApoio	SeguradoraConvenio	RolExames
Atendimento	x	x	x	x	x	x	x

Grupo 2 – (sem treinamento)

ELABORAÇÃO DA MATRIZ DE BARRAMENTO

A partir do levantamento das informações a serem buscadas, foi elaborada a Matriz de Barramento, constante na Figura 1.

	DIMENSÕES						
PROCESSOS DE NEGÓCIOS	Rol	Médico	Convênio	Paciente	Tipos de exames	Laboratório de apoio	Resultado de exame
Requisição de Exames	X	X	X	X	X	X	X

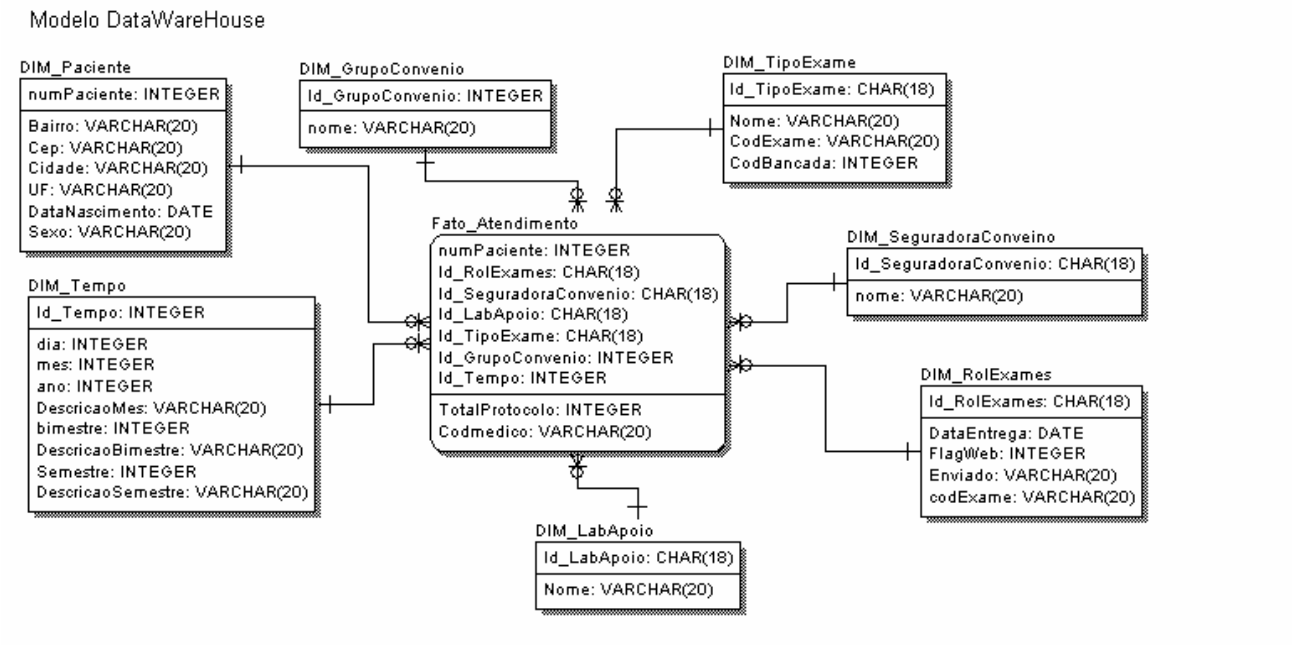
Figura 1: Matriz de barramento.

ANEXO VI

- ARTEFATO (MODELO DIMENSIONAL)

Grupo 1 – (com treinamento)

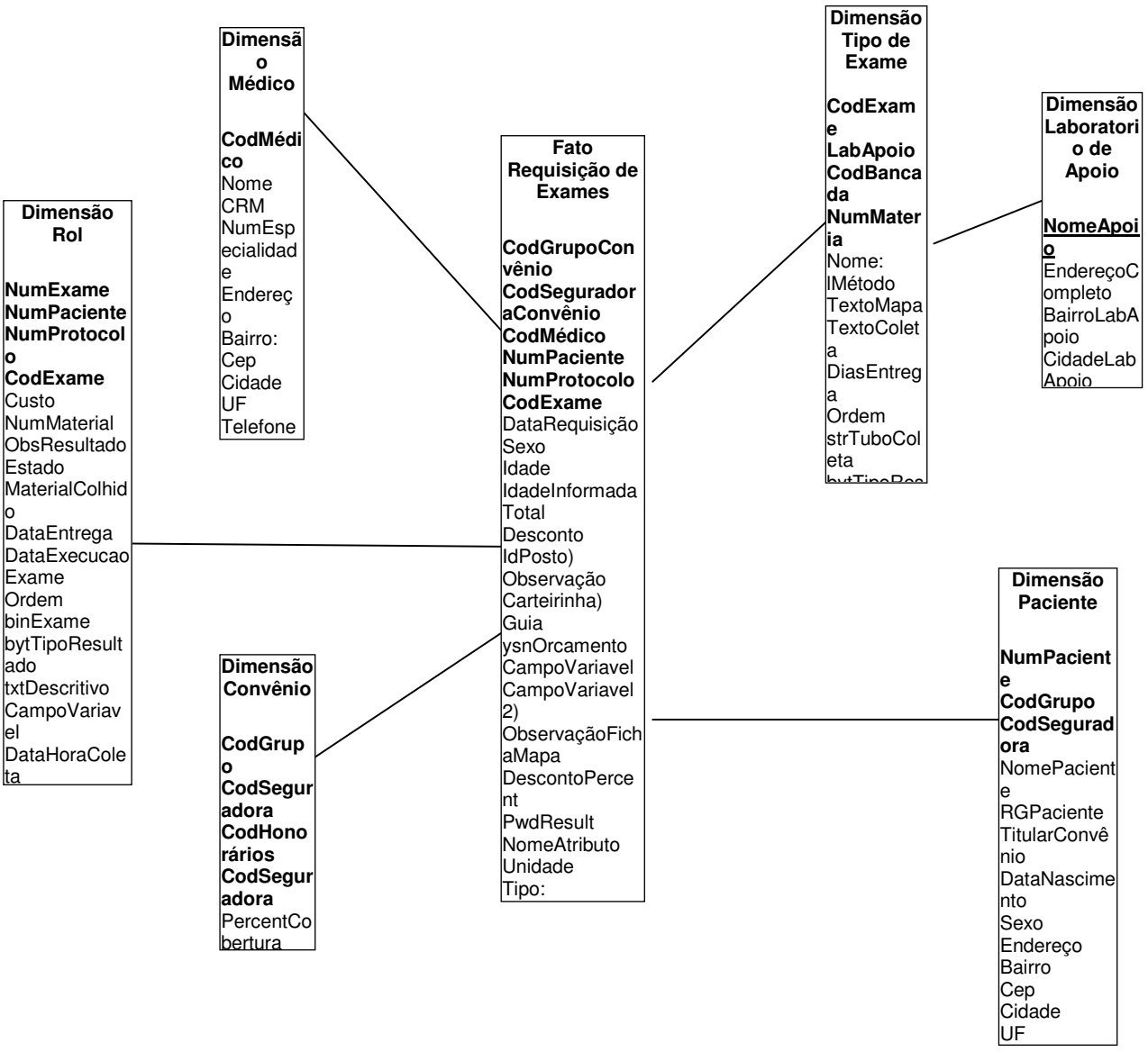
MODELO DIMENSIONAL



Grupo 2 – (sem treinamento)

ELABORAÇÃO DO MODELO DIMENSIONAL

Considerando as informações a serem levantadas e a Matriz de Barramento, foi elaborado o Modelo Dimensional.



Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo