Acta Scientiarum. Technology
ISSN: 1806-2563
eduem@uem.br
Universidade Estadual de Maringá
Brasil

Dias, Maria Madalena; dos Santos Pacheco, Roberto Carlos
Uma metodologia para o desenvolvimento de sistemas de descoberta de conhecimento
Acta Scientiarum. Technology, vol. 27, núm. 1, enero-junio, 2005, pp. 61-72
Universidade Estadual de Maringá
Maringá, Brasil

Uma metodologia para o desenvolvimento de sistemas de descoberta de conhecimento

Maria Madalena Dias[1]* e Roberto Carlos dos Santos Pacheco[2]

*[1]Departamento de Informática, Universidade Estadual de Maringá, Avenida Colombo, 5790, 87020-900, Maringá, Paraná, Brasil. [2]Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis, Santa Catarina, Brasil. *Autor para correspondência. e-mail: mmdias@din.uem.br*

RESUMO. Após a organização conseguir sanar seus problemas operacionais, surge a necessidade de sistemas para o suporte à tomada de decisão. A área de pesquisa de mineração de dados cresce rapidamente para atender a essas novas necessidades. No entanto, a aplicação de técnicas de mineração de dados pode tornar-se uma tarefa difícil e não confiável se não for seguida uma metodologia completa e sistemática no desenvolvimento de sistemas de descoberta de conhecimento. Este artigo apresenta uma metodologia, denominada MeDesC, que integra UML (*Unified Modeling Language*) e Linguagem E-LOTOS (*Enhancements to Language Of Temporal Ordering Specification*). O principal objetivo da utilização da metodologia MeDesC é gerar informações relevantes e confiáveis à tomada de decisão através da aplicação de técnicas de mineração de dados. A metodologia MeDesC foi utilizada no desenvolvimento de um sistema de descoberta de conhecimento, tendo como base de dados informações da pós-graduação Brasileira. O protótipo de um ambiente de descoberta de conhecimento deu suporte à implementação desse sistema.

Palavras-chave: mineração de dados, sistemas de descoberta de conhecimento em banco de dados, métodos formais, modelagem orientada a objetos.

ABSTRACT. A Methodology for the development of knowledge discovery systems. After the organization solves its operational problems, systems are necessary to support the decision making process. The data mining research area is growing quickly to assist such new needs of the organization. However, the implementation of data mining techniques may become a difficult and unreliable task unless a complete and systematic methodology is adopted in the development of knowledge discovery systems. This paper aims to introduce a methodology named MeDesC. This methodology integrates UML (Unified Modeling Language) and E-LOTOS (Enhancements to Language of Temporal Ordering Specification). The main objective is to generate relevant and reliable information for decision making, by means of the application of data mining techniques. The MeDesC methodology was used to develop a knowledge discovery system based on data from the Brazilian Post-graduation. The prototype of a knowledge discovery environment provided support to the implementation of this system.

Key words: data mining, knowledge discovery in database, formal methods, object-oriented model.

Introdução

A crescente exigência por sistemas confiáveis e de boa qualidade deu origem a muitas metodologias para o desenvolvimento de sistemas computacionais. No entanto, a maioria dessas metodologias não se aplica adequadamente a alguns tipos de sistemas, tais como: sistemas de tempo real e sistemas de descoberta de conhecimento em banco de dados. As propostas de metodologias para o desenvolvimento de sistemas de descoberta de conhecimento em banco de dados também não incluem formalismo, o que dificulta a obtenção de sistemas confiáveis e de qualidade.

Os sistemas de descoberta de conhecimento são considerados sistemas complexos. Por isto, eles exigem maior rigor no seu processo de desenvolvimento.

Os métodos formais são muito utilizados, atualmente, na especificação de sistemas complexos com o objetivo de construir sistemas de forma mais sistemática e sem ambigüidades.

O objetivo deste artigo é apresentar uma

Metodologia de Desenvolvimento de Sistemas de Descoberta de Conhecimento, denominada MeDesC, que inclui formalismo no processo de desenvolvimento desses sistemas.

Para clarificar esta apresentação, nas próximas seções são apresentados conceitos de mineração de dados, relacionadas tarefas e técnicas de mineração de dados e as fases básicas do processo de descoberta de conhecimento; são mostrados conceitos de métodos formais e relacionados os níveis de rigor que podem ser empregados no uso de métodos formais; são descritas, sucintamente, três metodologias de descoberta de conhecimento propostas por outros autores; é descrita a metodologia MeDesC; são demonstrados resultados obtidos no estudo de casos realizado e conclusões sobre a importância da inclusão de métodos formais no desenvolvimento de sistemas de descoberta de conhecimento em banco de dados e sobre os resultados obtidos na aplicação prática da metodologia MeDesC.

Mineração de Dados

"Mineração de dados é a exploração e a análise, por meio automático ou semi-automático, de grandes quantidades de dados, a fim de descobrir padrões e regras significativos" (Berry e Linoff, 1997).

A mineração de dados pode ser considerada como uma parte do processo de Descoberta de Conhecimento em Banco de Dados (*KDD – Knowledge Discovery in Databases*).

Segundo Goebel e Gruenwald (1999), o termo KDD é usado para representar o processo de tornar dados de baixo nível em conhecimento de alto nível, enquanto mineração de dados pode ser definida como a extração de padrões ou modelos de dados observados.

A mineração de dados combina métodos e ferramentas das seguintes áreas (Cratochvil, 1999): aprendizagem de máquina, estatística, banco de dados, sistemas especialistas e visualização de dados.

Os principais objetivos da mineração de dados são descobrir relacionamentos entre dados e fornecer subsídios para que possa ser feita uma previsão de tendências futuras baseada no passado.

Os resultados obtidos com a mineração de dados podem ser usados para gerenciamento de informação, tomada de decisão, controle de processo e muitas outras aplicações.

As técnicas de mineração de dados podem ser aplicadas sobre bancos de dados operacionais ou sobre *Data Warehouse* (DW) ou *Data Mart* (DM), nos quais geralmente resulta uma informação melhor, visto que os dados normalmente são preparados antes de serem armazenados no DW ou DM (Dias *et al.,* 1998). Podem ser aplicadas, inclusive, sobre um *Data Set* (DS), que contém apenas o conjunto de dados específico para um tipo de investigação a ser realizada.

"Um DW é um conjunto de dados baseado em assuntos, integrado, não-volátil e variante em relação ao tempo, de apoio às decisões gerenciais" (Inmon, 1997).

Um DM é um DW departamental, ou seja, um DW construído para uma área específica da organização (Inmon, 1997).

As técnicas de mineração de dados podem ser aplicadas a tarefas[1] como (Dias, 2001):

- Classificação: constrói um modelo de algum tipo que possa ser aplicado a dados não classificados a fim de categorizá-los em classes;
- Estimativa (ou regressão): usada para definir um valor para alguma variável contínua e desconhecida;
- Associação: usada para determinar quais itens tendem a ser adquiridos juntos em uma mesma transação;
- Segmentação (ou *clustering*): processo de partição de uma população heterogênea em vários subgrupos ou grupos mais homogêneos;
- Sumarização: envolve métodos para encontrar uma descrição compacta para um subconjunto de dados.

Harrison (1998) afirma que não há uma técnica que resolva todos os problemas de mineração de dados. Diferentes métodos servem para diferentes propósitos, e cada método oferece suas vantagens e suas desvantagens. A familiaridade com as técnicas é necessária para facilitar a escolha de uma delas de acordo com os problemas apresentados. As técnicas de mineração de dados normalmente usadas são (Dias, 2001):

- Descoberta de regras de associação: estabelece uma correlação estatística entre atributos de dados e conjuntos de dados;
- Árvore de decisão: hierarquização dos dados, baseando-se em estágios de decisão (nós) e na separação de classes e subconjuntos;
- Raciocínio baseado em casos ou MBR: baseado no método do vizinho mais próximo, combina e compara atributos para estabelecer hierarquia de semelhança;
- Algoritmos genéticos: métodos gerais de busca e otimização, inspirados na Teoria da Evolução,

[1] Neste contexto, tarefa é um tipo de problema de descoberta de conhecimento a ser solucionado.

onde a cada nova geração, soluções melhores têm mais chance de ter "descendentes";

- Redes neurais artificiais: modelos inspirados na fisiologia do cérebro, na qual o conhecimento é fruto do mapa das conexões neuronais e dos pesos dessas conexões.

A técnica descoberta de regras de associação soluciona problema de associação. A técnica árvore de decisão soluciona problemas de classificação e regressão. As demais técnicas solucionam problemas de classificação e segmentação.

O Processo de Descoberta de Conhecimento

O processo de descoberta de conhecimento é um método semi-automático complexo e iterativo (Mannila, 1996). A Figura 1 representa o processo de descoberta de conhecimento, de acordo com Groth, (1998) e Lans (1997) e sintetizado em Dias (2001), em cujo estudo as etapas desse processo são descritas. Vale lembrar que as atividades relacionadas em cada etapa podem ser adaptadas para atender à especificidade dos dados a serem analisados e da tarefa de mineração de dados escolhida.

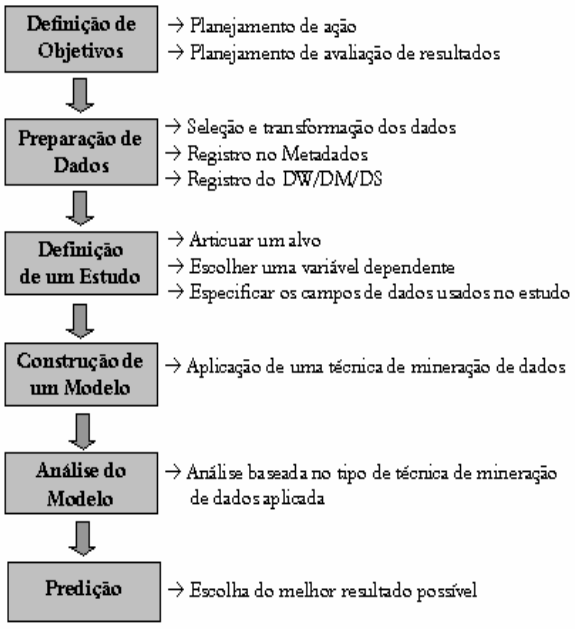

Figura 1. Processo de Descoberta de Conhecimento.

Métodos Formais

"Métodos formais são um conjunto de ferramentas e notações (com uma semântica formal) usado para especificar de forma não ambígua os requisitos de um sistema que suporta a prova ou propriedades daquela especificação e provas de corretude de uma implementação para aquela especificação" (Wiryana, 1998).

Os métodos formais podem ser aplicados em graus variados de rigor. Podem ser aplicados em estágios selecionados ou em todos os estágios do ciclo de vida do sistema, e para alguns ou todos os componentes e propriedades do sistema.

Rushby (1993) classifica métodos formais em quatro níveis de rigor crescente:

Nível 0: Nenhum uso de métodos formais;

Nível 1: Uso de conceitos e notações da matemática discreta.

Nível 2: Uso de linguagens de especificação formalizadas com algumas ferramentas de suporte mecanizadas;

Nível 3: Uso de linguagens de especificação totalmente formais, incluindo teorema mecanizado, provando ou checando provas.

A escolha do nível de rigor depende dos benefícios desejados no uso de métodos formais, do fator de segurança crítica da aplicação e dos recursos disponíveis.

Trabalhos Relacionados

O desenvolvimento de um sistema de descoberta de conhecimento em banco de dados é uma tarefa muito complexa, principalmente pela característica de não determinismo do processo de desenvolvimento desse tipo de sistema. Por conseguinte, é imprescindível o uso de uma metodologia completa e sistemática.

Os trabalhos que se propõem a apresentar uma metodologia para o desenvolvimento de sistemas de descoberta de conhecimento não incluem formalismo na especificação desses sistemas. Normalmente as metodologias propostas procuram solucionar questões relativas a determinadas etapas do processo de desenvolvimento desses sistemas e não apresentam notação para representar as características do sistema como um todo.

A seguir, são relacionados trabalhos que propõem uma metodologia para sistemas de descoberta de conhecimento em banco de dados.

a) Metodologia de Klemettinen

Klemettinen *et al.* (1997) apresentam uma metodologia que pode ser usada para automatizar aquisição de conhecimento. As fases dessa metodologia são aquelas já definidas por outros autores (Fayyad *et al.*, 1996; Mannila, 1996): pré-processamento, transformação, descoberta, apresentação e utilização (Figura 2). No entanto, maior ênfase é dada nas duas fases centrais dessa metodologia:

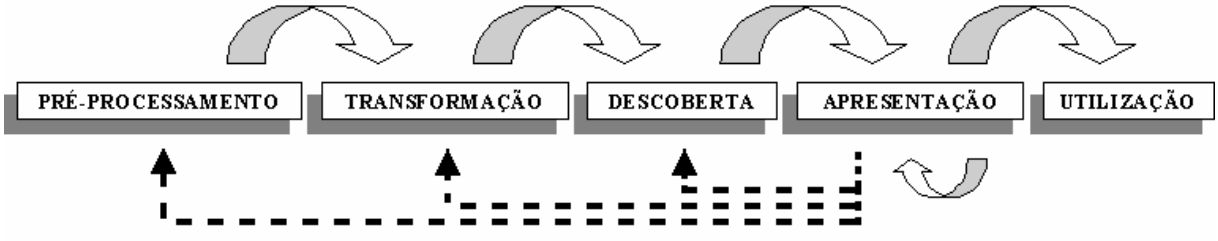

Figura 2. O Modelo do Processo KDD (Klemettinen *et al.*, 1997).

- Fase de descoberta de padrões: na qual são encontrados todos os padrões potencialmente relevantes para algum critério bastante livre;
- Fase de apresentação: em que são fornecidos métodos flexíveis para iterativa e interativamente criar diferentes visões para os padrões descobertos.

Nas duas primeiras fases do processo, os dados são coletados e preparados de forma adequada para a descoberta de padrões. Uma visão geral sobre os dados pode ser produzida nessa fase. Os atributos identificados como irrelevantes são removidos e novos atributos podem ser derivados.

Na fase de descoberta de padrões, todos os padrões potencialmente interessantes são gerados do conjunto do *data set*.

A apresentação do conhecimento descoberto é uma parte principal dessa metodologia. Nessa fase, os padrões relevantes podem ser localizados de grandes coleções de padrões potencialmente relevantes.

b) Metodologia de Feldens

Feldens *et al.* (1998) propõem uma metodologia integrada, na qual as tecnologias de mineração de dados e *data warehouse*, bem como questões de visualização têm papéis muito importantes no processo. Também supõem uma forte interação entre mineradores de dados e pessoas da organização para questões de modelagem e preparação de dados. As fases definidas para essa metodologia são: pré-processamento, mineração de dados e pós-processamento, conforme Figura 3.

A fase de pré-processamento inclui tudo o que é feito antes da mineração de dados, o que significa a análise que é feita na organização a fim de enfocar o projeto de mineração de dados, a análise dos dados existentes, a integração de fontes de dados, as transformações de dados, etc.

A fase de mineração de dados inclui a aplicação de algoritmos, possivelmente a aplicação repetida. A escolha dos algoritmos pode ser realizada baseando-se na análise feita na fase de pré-processamento.

A fase de pós-processamento pode ser definida por operações de filtragem, estruturação e classificação. Somente após essa fase o conhecimento descoberto é apresentado ao usuário.

O conhecimento descoberto pode ser filtrado por alguma medida estatística, por exemplo, suporte, confiança ou outro critério definido pelo usuário. Estruturação significa que o conhecimento pode ser organizado de forma hierárquica.

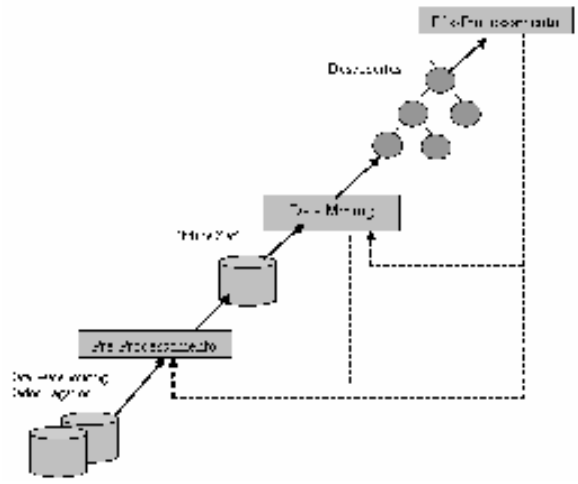

Figura 3. Processo KDD (Feldens *et al*, 1998).

c) Modelo de Processo CRISP-DM

O Modelo de Processo CRISP-DM (*CRoss-Industry Standard Process for Data Mining*) define um processo de mineração de dados não linear (CRISP-DM, 2005), conforme pode ser visto na Figura 4.

Nesse modelo, o ciclo de vida do projeto de mineração de dados consiste em seis fases. A seqüência dessas fases não é rigorosa, depende do resultado de cada fase ou de qual tarefa particular de uma fase precisa ser executada na próxima fase. As flechas indicam as dependências mais importantes e freqüentes entre as fases.

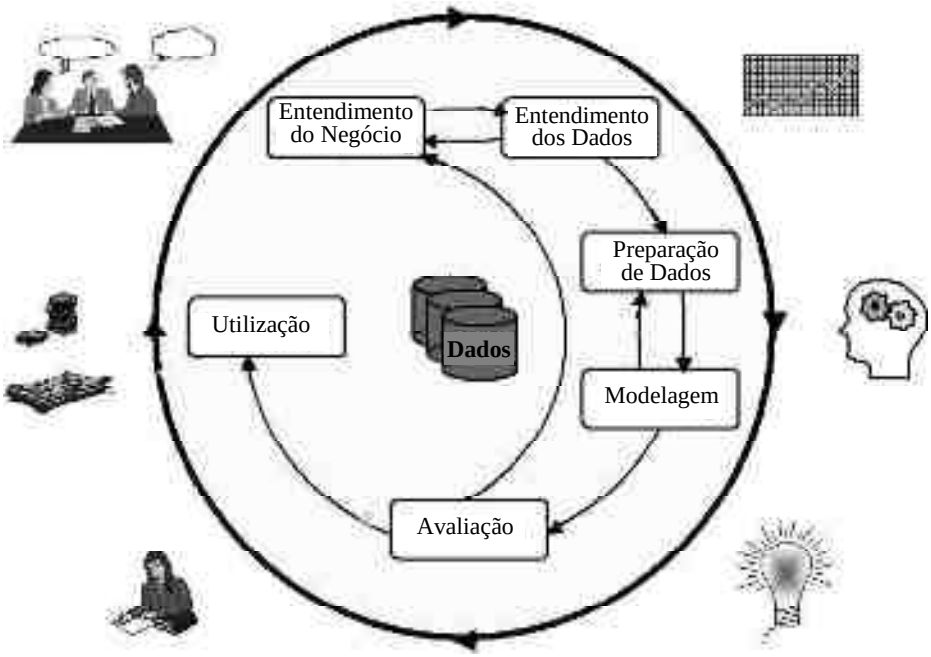

Figura 4. Fases do Modelo de Processo CRISP-DM

O círculo externo na figura simboliza a natureza cíclica da mineração de dados. Um processo de mineração de dados continua após uma solução ter sido descoberta. Os processos de mineração de dados subseqüentes se beneficiarão das experiências anteriores.

A seguir, cada fase do modelo é definida sucintamente.

A fase inicial do processo, Entendimento do Negócio (*Business Understanding*), visa o entendimento dos objetivos do projeto e dos requisitos sob o ponto de visa ao negócio. Com baseado no conhecimento adquirido, o problema de mineração de dados é definido e um plano preliminar é projetado para ativar os objetivos.

A fase Entendimento dos Dados (*Data Understanding*) inicia com uma coleção de dados e procede com atividades que visam: buscar familiaridade com os dados, identificar problemas de qualidade de dados, descobrir os primeiros discernimentos nos dados ou detectar subconjuntos interessantes para formar hipóteses da informação escondida.

A fase Preparação de Dados (Data Preparation) cobre todas as atividades de construção do dataset final. As tarefas de preparação de dados são, provavelmente, desempenhadas várias vezes e não em qualquer ordem prescrita. Essas tarefas incluem seleção de tabelas, registros e atributos, bem como transformação e limpeza dos dados para as ferramentas de modelagem.

Na fase Modelagem (Modelling), várias técnicas de modelagem são selecionadas e aplicadas e seus parâmetros são ajustados para valores ótimos. Geralmente existem várias técnicas para o mesmo tipo de problema de mineração de dados. Algumas técnicas têm requisitos específicos na formação de dados. Portanto, retornar à fase de preparação de dados é freqüentemente necessário.

Na fase Avaliação (Evaluation), o modelo (ou modelos) construído na fase anterior é avaliado e são revistos os passos executados na sua construção para se ter certeza de que o modelo representa os objetivos do negócio. O principal objetivo é determinar se existe alguma questão de negócio importante que não foi suficientemente considerada. Nessa fase, uma decisão sobre o uso dos resultados de mineração de dados deverá ser alcançada.

Após o modelo (ou modelos) ser construído e avaliado, na fase Utilização, ou Aplicação (Deployment), ele pode ser usado de duas formas. Na primeira, o analista pode recomendar ações a serem tomadas baseando-se simplesmente na visão do modelo e de seus resultados. Na segunda forma, o modelo pode ser aplicado a diferentes conjuntos de dados.

d) A Metodologia MeDesC

As principais etapas da metodologia MeDesC são baseadas no ciclo de vida clássico de desenvolvimento de sistemas (Pressman, 2000) e nos passos básicos para o desenvolvimento de sistemas de descoberta de conhecimento em banco de dados (Groth, 1998). São elas: análise do sistema, projeto informal, projeto formal, implementação e análise dos resultados. A Figura 5 representa estas etapas.

Figura 5. Etapas da Metodologia MeDesC.

Apesar das etapas de análise do sistema, projeto informal e implementação constarem na grande maioria das metodologias de desenvolvimento de sistemas computacionais, os objetivos dessas etapas na metodologia MeDesC são diferentes daqueles definidos para outras metodologias.

A metodologia proposta define um processo iterativo de desenvolvimento de sistemas de descoberta de conhecimento em banco de dados, por prever a necessidade de retornar às etapas de projeto informal e projeto formal, caso seja encontrado algum erro durante a verificação/validação do sistema. Após a análise dos resultados, pode-se retornar às etapas de análise do sistema e de projeto informal para que seja definido um novo sistema ou ajustado o sistema já desenvolvido. Nesse caso, os resultados obtidos podem ser utilizados como base.

Na etapa de análise do sistema e de projeto informal, são utilizados diagramas UML (*Unified Modeling Language*) (Booch *et al.*, 1999) para representar os objetos do sistema, seus comportamentos e suas interações.

A UML foi escolhida para ser usada na metodologia como forma de representação das características do sistema por ser uma linguagem unificada, definida como padrão, para modelagem de objetos; definir diferentes tipos de diagramas que facilitam o desenvolvimento de sistemas e por separar claramente os aspectos estáticos dos dinâmicos e funcionais, permitindo fáceis mudanças.

Além disso, segundo Pressman (2000), a atividade de modelagem é fundamental para uma boa análise.

A etapa de projeto formal foi incluída na metodologia para tornar mais rigoroso o processo de desenvolvimento de sistemas de descoberta de conhecimento em banco de dados. A formalização da especificação do sistema é realizada através do mapeamento dos modelos UML, construídos na fase de projeto informal, para a linguagem de especificação formal E-LOTOS (ISO/IEC, 2005).

E-LOTOS foi escolhida por ser definida como um padrão de linguagem de especificação formal.

A especificação formal do sistema poderá ser verificada e validada com a utilização de uma ferramenta adequada e, no caso de ser detectado algum erro, deve-se retornar à etapa de projeto informal ou à etapa de projeto formal.

Na metodologia MeDesC, foi aplicado o nível 1 de utilização de método formal.

A implementação é realizada tomando como base os diagramas UML, construídos na etapa de projeto informal, mas só após a especificação formal ser validada, para que seja garantida a obtenção de um sistema de boa qualidade.

Atualmente, está sendo desenvolvido um ambiente de descoberta de conhecimento em banco de dados, denominado ADesC, para dar suporte às etapas da metodologia MeDesC. Nesse ambiente serão implementadas as atividades necessárias para a construção de um DW/DM, para a seleção de atributos na definição de um conjunto de dados (*data set*), para a aplicação de técnicas de mineração de dados e para a análise dos resultados.

Nas próximas seções são descritas as estratégias a serem adotadas em cada etapa da metodologia, de acordo com as técnicas utilizadas.

Análise do sistema

A primeira etapa da metodologia, a análise do sistema, tem como principais objetivos: definir tipos de investigações a serem realizadas com a aplicação de técnicas de mineração de dados e identificar as fontes de dados necessárias nessas investigações. A Figura 6 representa as principais atividades desta etapa.

Projeto informal

A etapa de projeto informal consiste na seleção dos atributos, definição das transformações de dados necessárias, projeto de uma estrutura de metadados, projeto do DW/DM, definição de técnica de amostragem estatística, escolha de uma ou mais técnicas de mineração de dados, construção dos diagramas de classe e na descrição do comportamento dos objetos através dos diagramas de estado e de colaboração.

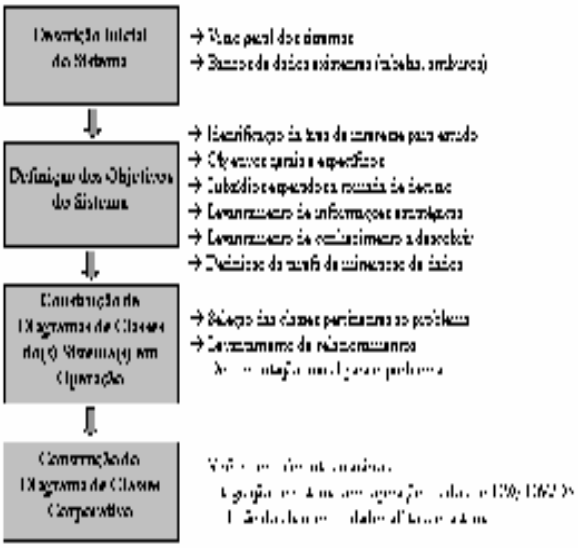

Figura 6. Atividades da Etapa Análise do Sistema.

As principais atividades dessa etapa estão representadas na Figura 7.

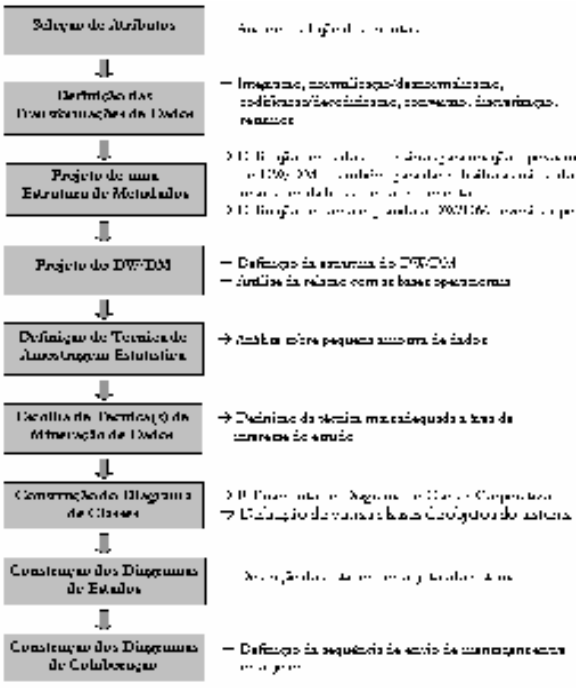

Figura 7. Atividades da Etapa Projeto Informal.

Projeto formal

O objetivo da etapa de projeto formal é dar maior rigor ao processo de desenvolvimento de sistema de descoberta de conhecimento em banco de dados. A utilização de uma técnica de descrição formal torna possível a realização de verificação e validação do sistema especificado. Nessa etapa, é adotada a linguagem de especificação formal E-LOTOS.

E-LOTOS é uma linguagem de especificação formal adequada por permitir a especificação de sistemas de forma modular, por possibilitar a representação de tempo que pode ser utilizada para mostrar, por exemplo, quando o DW/DM deverá ser povoado pelos dados atuais e, principalmente, por permitir a verificação e a validação do sistema.

No projeto formal, deve ser construído um modelo formal em E-LOTOS, manual ou automaticamente, a partir dos diagramas de UML.

A formalização de modelos orientados a objetos deve ser realizada através da tradução das características desses modelos para as representações existentes na linguagem formal utilizada.

No entanto, existem dificuldades na incorporação de TDFs (Técnicas de Descrição Formal) no desenvolvimento orientado a objetos usando uma linguagem de modelagem. A principal dificuldade é a especificação formal do comportamento dinâmico global de sistemas baseados em objetos usando as TDFs atuais. Isto ocorre porque a maioria dessas TDFs não incorpora características desses sistemas, tais como modularidade, herança, encapsulamento etc.

Essas dificuldades podem ser superadas através da definição de uma forma de mapeamento entre os conceitos da modelagem orientada a objetos e da TDF utilizada.

Com a utilização de UML e de E-LOTOS, o resultado desse mapeamento é um modelo formal que contém uma especificação do sistema em E-LOTOS, a partir de características apresentadas nos diagramas UML. O modelo formal integra, assim, as propriedades estáticas e dinâmicas do sistema. Como a especificação formal obtida é executável, pode-se usar a prototipagem para validá-la.

A Figura 8 mostra os passos para a construção do modelo formal em E-LOTOS.

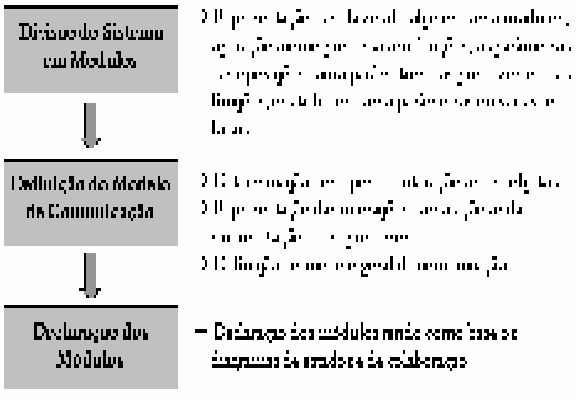

Figura 8. Atividades da Etapa Projeto Formal.

A seguir, são relacionados alguns exemplos do mapeamento de diagramas UML para E-LOTOS.

a) Herança

A Figura 9 mostra um exemplo de herança simples entre classes de objetos. Essa figura é baseada em UML (Booch *et al.*, 1999).

Figura 9. Exemplo de Herança entre Classes de Objetos.

O exemplo descreve a relação de compartilhamento de atributos e métodos de docentes e discentes, quando ambos são abstraídos como indivíduos.

A especificação na linguagem E-LOTOS desse exemplo é apresentada na seqüência.

```
module modBaseDados is
  process Basedados [...] is
     ...
  endproc
  process BuscarTabelas [...] (...) is
     ...
  endproc
  (* ... outros processos *)
endmod
module modDocente import modBaseDados is
  process Docente [...] (...) is
     ...
  endproc
  process VerAtributos [...] (...) is
     ...
  endproc
   (* ... outros processos *)
endmod
module modDiscente import modBaseDados is
  process Discente [...] (...)  is
     ...
  endproc
  process TransformarDados [...] is
     ...
   endproc
   (* ... outros processos *)
endmod
```

A cláusula "*import*" é utilizada para declarar que um módulo herda as características de outro(s) módulo(s).

Cada classe de objetos representada na Figura 9 é traduzida para E-LOTOS como um módulo, por exemplo, *modBaseDados* é a especificação da classe de objetos *BaseDados*.

Os processos *BaseDados*, *Docente* e *Discente* representam os construtores das classes e é através deles que ocorre a sincronização com outras classes de objetos.

b) Modelo de comunicação

O modelo de comunicação, apresentado na Figura 10, é descrito em E-LOTOS pelos módulos abaixo. O módulo *modDADOS* define os tipos de dados do sistema e o módulo de especificação *AmbDesco* descreve o comportamento do sistema.

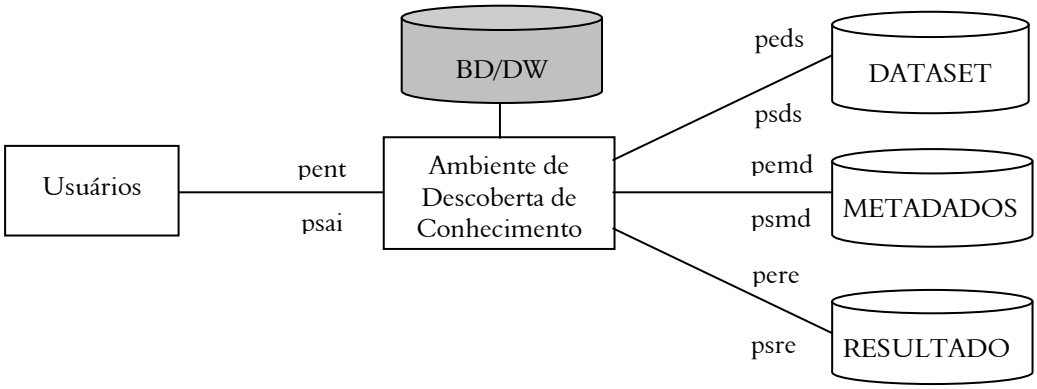

Figura 10. Modelo Geral de Comunicação.

```
module modDADOS is
  type Typent is
          ...
  endtype
  type Typesai is
          ...
  endtype
  (* ... outros tipos *)
  type Portaent is (ent: Typent) endtype
  type Portasai is (sai: Typesai) endtype
  type Portaebd is (ebd: Typebd) endtype
  type Portasbd is (sbd: Typebd) endtype
  type Portaeds is (eds: Typeds) endtype
  type Portasds is (sds: Typeds) enddtype
  type Portaemd is (emd: Typemd) endtype
  type Portasmd is (smd: Typemd) endtype
  type Portaere is (ere: Typere) endtype
  type Portasre is (sre: Typere) endtype
endmod
specification AmbDesco import modDados,
modUsuario, modAmbiente, modBaseDados,
modDataSet, modMetadados, modResultados is
  gates  pent: Portaent, psai: Portasai, pebd:
Portabd, psbd: Portabd, peds: Portads,
          psds: Portads, pemd: Portamd, psmd:
          Portamd, pere: Portare, psre: Portare
Behaviour
(
   (
     Usuario [pent, psai]  (0)
     |||
     ...
     Usuario [pent, psai] (n-1)
   )
   |[pent, psai]|
   Ambiente [pent, psai, pebd, psbd, peds, psds,
pemd, psmd, pere, psre]
   |[pebd, psbd, peds, psds, pemd, psmd, pere,
psre]|

          Bancodados [pebd, psbd]
     |||
          Dataset [peds, psds]
          |||
          Metadados [pemd, psmd]
          |||
          Resultados [pere, psre]
   )
)
endspec
```

O módulo de especificação AmbDesco inclui (usando a cláusula import) o módulo modDADOS, que define os tipos de dados, e os módulos que declaram as classes de objetos Usuario, Ambiente, Basedados, Dataset, Metadados e Resultados. Ele declara o conjunto de portas tipadas por onde ocorrem as interações entre os objetos e, também, representa os objetos da Figura 10 como processos paralelos, que são sincronizados por meio de portas de comunicação.

O conjunto de n usuários (indexado de 0 a n-1) é sincronizado com o processo Ambiente, representando a associação muitos-para-um. Da mesma forma, poderão ser representadas as associações um-para-muitos, muitos-para-muitos e um-para-um.

O conjunto de n usuários corresponde à instanciação de n objetos da classe Usuario. A instanciação de um objeto em E-LOTOS é representada, portanto, pela instanciação de um processo (ou um conjunto de processos) que define o estado e o comportamento da classe a qual ele pertence.

Implementação do sistema

Nessa etapa, o sistema é implementado a partir dos diagramas UML resultantes da fase de projeto informal, podendo ser utilizado o ambiente ADesC. No entanto, o usuário pode optar por implementá-lo em outro ambiente de programação de sua preferência ou utilizar alguma outra ferramenta disponível no mercado.

Análise dos resultados

Nessa fase, são definidas estratégias para a análise dos resultados obtidos pelo sistema de descoberta de conhecimento em banco de dados.

Uma estratégia a ser utilizada é o "Refinamento do Conjunto de Regras Descoberto" (Freitas, 1998), cujo objetivo é reduzir o número de regras (e o número de condições em cada regra) a ser apresentado ao usuário. Freitas (1998) define duas abordagens básicas para esse tipo de estratégia: subjetiva e objetiva.

A abordagem subjetiva (controlada pelo usuário) procura selecionar regras de classificação mais interessantes, baseando-se no conhecimento do usuário.

A abordagem objetiva (autônoma, controlada pelo sistema) também procura selecionar regras mais interessantes, mas baseando-se nos dados, em vez de basear-se no conhecimento do usuário.

Nessa etapa, o modelo de mineração de dados construído deve ser validado. Isto implica a avaliação de seus resultados e na interpretação de seu significado.

Estudo de casos

O estudo de casos realizado teve como principal objetivo estabelecer a construção de sistemas de descoberta de conhecimento segundo a metodologia proposta e verificar a efetividade de seus resultados.

As etapas da metodologia MeDesC foram seguidas passo a passo e o sistema foi implementado através do protótipo do ambiente ADesC.

A realização das atividades definidas em cada etapa da metodologia MeDesC garantiram a segurança dos resultados obtidos.

No desenvolvimento de sistemas computacionais, é importante o uso de modelos para expressar suas características, como está previsto na metodologia MeDesC, o que não ocorre nas outras metodologias apresentadas.

A técnica de mineração de dados Regras de Associação foi aplicada tendo como base os dados contidos em bancos de dados da Capes dos Programas de Pós-Graduação do Brasil no ano de 1998.

A seguir, são apresentados os resultados obtidos em três investigações realizadas.

1) Estudo da relação fomento x produtividade

O objetivo desse estudo foi verificar se existe alguma relação entre quantidade de bolsas fornecidas ao Programa e a Produtividade do seu corpo docente. A Tabela 1 mostra as regras geradas neste estudo.

Tabela 1. Regras geradas para a primeira investigação.

Regra	Suporte	Confiança
Se o Programa possui de 19 a 24 meses de financiamento por aluno concluído, então ele produz de 2 a 3 publicações em média por pesquisador	19,62%	53,76%
Se o Programa possui de 25 a 30 meses de financiamento por aluno concluído, então ele produz de 2 a 3 publicações por pesquisador	16,88%	48,19%

Pode-se observar que o aumento relativo na quantidade de bolsas de um Programa implica aumento na Produção, mas há um nível de saturação para esta regra (quando média de bolsa por aluno está entre 25 e 30 meses). É interessante observar que essa constatação vai ao encontro à política de redução do tempo máximo de bolsa adotada pela Capes.

2) Estudo da relação entre a carga de orientação e tempo de titulação dos orientandos

O objetivo desse estudo foi verificar se existe alguma relação entre o tempo de formação do aluno e o total de orientandos do seu orientador. A Tabela 2 apresenta as regras geradas nesse estudo.

Tabela 2. Regras geradas para a segunda investigação.

Regra	Suporte	Confiança
Se o orientador possui de 4 a 7 orientandos, então a média de tempo de formação de seus orientandos de mestrado é maior que 36 meses	16,92%	49,21%
Se o orientador possui 1 orientando, então o tempo de formação de seu orientando de mestrado é maior que 36 meses	8,43%	41,71%

O estudo mostra que há um equilíbrio nos valores de confiança entre o número de orientandos de um orientador e o tempo médio de formação de seus orientandos. Isto indica que não há suporte para a afirmação que alunos de um professor com mais orientandos tenham maior tempo médio de titulação.

3) Estudo da relação entre o tempo de titulação com a disponibilidade do Fomento (bolsas) no programa ou com a participação discente em projetos

O objetivo deste estudo foi verificar se existe alguma relação entre o tempo de formação do aluno com o fato do mesmo possuir ou não bolsa e com a vinculação de sua dissertação a projetos. A Tabela 3 relaciona as regras obtidas neste estudo.

Tabela 3. Regras geradas para a terceira investigação.

Regra	Suporte	Confiança
Se o aluno possui bolsa, então sua dissertação está vinculada a projeto de pesquisa	36,88%	53,27%
Se o aluno possui bolsa, então seu tempo de formação é maior que 36 meses	28,34%	40,94%
Se a dissertação está vinculada a projeto, então o tempo de formação do aluno é maior que 36 meses	21,30%	42,72%
Se o aluno possui bolsa e sua dissertação está vinculada a projeto, então seu tempo de formação é maior que 36 meses	14,11%	38,26%
Se a área é Ciências Exatas e da Terra, então o aluno possui bolsa	10,16%	82,66%
Se a área é Ciências Humanas, então o aluno possui bolsa	13,06%	72,70%
Se a área é Engenharia, então o aluno possui bolsa	11,44%	65,78%
Se a área é Engenharia, então a dissertação possui vínculo com projeto	10,04%	57,70%

Através do terceiro estudo, pode-se constatar que há um aumento no tempo médio de titulação quando o aluno possui bolsa ou quando sua dissertação está relacionada a projeto de pesquisa.

No estudo também foi comparada a implicação de participação de bolsistas em projetos. A regra permite afirmar que o impacto da concessão de bolsa na vinculação com projeto é maior do que no tempo médio de titulação.

Constatou-se ainda que os Programas das áreas de Ciências Exatas e da Terra, Ciências Humanas e Engenharia são aqueles que possuem o maior número de bolsas, e também que mais da metade das dissertações da área de Engenharia possui vínculo com projeto.

Conclusão

O desenvolvimento e a implementação de sistemas de descoberta de conhecimento em banco de dados são tarefas cuja complexidade depende da natureza de seus sistemas e da necessidade do alto grau de conhecimento do negócio da empresa pelo analista de sistemas.

A característica de não determinismo presente no processo de desenvolvimento de sistemas de descoberta de conhecimento faz com que esses sistemas se diferenciem de outros tipos de sistemas. Por isto, o uso indiscriminado de metodologias clássicas de desenvolvimento de sistemas torna-se inadequado. É necessária a utilização de metodologia específica.

A má especificação de qualquer tipo de produto de *software* pode levar a resultados incorretos, que podem causar graves conseqüências. No caso de sistemas de descoberta de conhecimento em banco de dados, os resultados incorretos podem levar a tomadas de decisões também incorretas. Essas decisões podem causar grandes prejuízos financeiros à empresa, ou um diagnóstico errado (ex: no caso da aplicação de técnicas de mineração de dados na área da saúde).

As metodologias de implantação de sistemas de Descoberta de Conhecimento são baseadas em iterações e interações de diferentes etapas (da análise do negócio à apresentação do conhecimento descoberto). Uma das principais características das metodologias propostas por outros autores é a ausência de técnicas de formalização que possam impedir a ambigüidade ou ineficácia de tais processos, considerando o não cumprimento de objetivos iniciais do sistema de Descoberta de Conhecimento.

A metodologia MeDesC define um processo de formalização, completo e sistemático, de desenvolvimento de sistemas de descoberta de conhecimento em banco de dados que aplicam técnicas de mineração de dados.

Além de garantir a confiabilidade do sistema através da formalização do processo de desenvolvimento, a metodologia MeDesC faz uso das vantagens da modelagem orientada a objetos (herança, encapsulamento, reuso) e utiliza UML para representar as características do sistema.

As demais metodologias descritas neste artigo não apresentam formalismo na especificação do sistema e, também, não definem ou utilizam modelos para representar características do sistema.

Foram apresentados alguns exemplos na etapa de projeto formal da metodologia proposta para que o leitor possa ter um melhor entendimento sobre como os modelos UML podem ser mapeados para a linguagem E-LOTOS.

As etapas da metodologia MeDesC, se seguidas corretamente, levam a especificações de sistemas de descoberta de conhecimento em banco de dados corretas, verificadas e validadas, contribuindo, assim, na construção de sistemas confiáveis e de qualidade.

As conclusões do estudo de casos permitiram mostrar a relevância da metodologia MeDesC na obtenção de resultados de mineração de dados partindo-se de hipóteses levantadas por usuários e buscando-se, passo a passo, meios de se chegar à prova verdadeira ou falsa dessas hipóteses.

Referências

BERRY, M.J.A.; LINOFF, G. *Data mining techniques*. New York: John Wiley & Sons, Inc., 1997.

BOOCH, G. *et al. The unified modeling language user guide.* New York: Addison Wesley, 1999.

CRATOCHVIL, A. *Data mining techniques in supporting decision making*. 1999. Master Thesis - Universiteit Leiden, Leiden, 1999.

CRISP-DM (CRoss Industry Standard Process for Data Mining). CRISP-DM Process Model. Disponível em: *http: //www.crisp-dm.org/Process/index.htm*. Acesso em Junho/2005.

DIAS, M.M. *et al.* Data warehouse – presente e futuro. *Revista Tecnológica*, Maringá, n. 7, p. 59-73, 1998.

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

FAYYAD, U. *et al.* From data mining to knowledge discovery: an overview. *In:* ADVANCES IN KNOWLEDGE DISCOVERY AND DATA MINING, 1996, Menlo Park. *Proceedings…* Menlo Park: AAAI Press, 1996, p. 1-34.

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, Equador. *Proceedings...* Equador, 1998.

FREITAS, A.A. *Data mining – mineração de dados*. Apostila de curso ministrado no CEFET-PR, Curitiba, DAINF, 1998.

GOEBEL, M.; GRUENWALD, L. A survey of data mining and knowledge discovery software tools. *ACM SIGKDD*, San Diego, v.1, n.1, p.20-33, 1999.

GROTH, R. *Data mining*. New Jersey: Prentice Hall, Inc., 1998.

HARRISON, T.H. *Intranet data warehouse*. São Paulo: Editora Berkeley, 1998.

INMON, W.H. *Como construir o Data Warehouse*. Editora Campus, 1997.

ISO/IEC JTC1/SC21/WG7. Final commite draft on

Enhancements to LOTOS. Editor: "Enhancement to LOTOS" (1.21.20.2.3). Disponível em: *http://lotos.site.uottawa.ca/elotos/*. Acesso em Junho/2005.

KLEMETTINEN, M. *et al.* A data mining methodology and its application to semi-automatic knowledge acquisition. *In:* INTERNATIONAL CONFERENCE AND WORKSHOP ON DATABASE AND EXPERT SYSTEMS APPLICATIONS (DEXA'97), 8., 1997, Toulouse. *Proceedings…* Toulouse, 1997, p. 670-677.

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, São Paulo. *Anais ...* São Paulo, 1997. p. 1-38.

MANNILA, H. Data mining: machine learning, statistics, and databases. *In:* INTERNATIONAL CONFERENCE ON SCIENTIFIC AND DATABASE MANAGEMENT, 8., 1996, Stockholm. *Proceedings…* Stockholm, 1996. p.1-8.

PRESSMAN, R. *Software Engineering: a practtioner's approach*. 5th ed. McGraw-Hill (Series in Computer Science), 2000.

RUSHBY, J. Formal methods and the certification of critical systems. Computer Science Laboratory SRI International, Menlo Park CA 94025 USA. Technical Report CSL-93-7, 1993.

WIRYANA, M. *Information system development: an interdiscipline approach*, Disponível em: *http://wiryana.pandu.org/artikel/paper_issm/*. Acesso em março/2005.

Received on March 10, 2005.
Accepted on June 13, 2005.