

PlotDBase – Integração de Narrativas e Dados de Aplicativos na forma de enredos para Descoberta de Conhecimento

¹Lucas Mortimer Macedo

Antonio L. Furtado

{mortimer, furtado}@inf.puc-rio.br

PUC-RioInf.MCC20/02 Julho, 2002

Abstract:

Data information affecting people and companies may be grouped as narratives of real situations and converted into plots, composed by characters or actors, objects or resources, events or conditions, sequence of actions, goal inference rules and constraints. Plots can be described in format of logical predicates in defined clauses, as used by Prolog.

In this paper we propose the creation of PlotDBase, a relational database of plots, which contains the models and instances, composed by sets of terms and their relationships, based on criteria and properties defined by expert users. The data used to build plots are picked up from application databases and several narratives.

The PlotDBase is useful to integrate specific application systems and non-standard narratives. We propose the development of a tool that has features for data entry, data preparing and data processing as: selection, comparison and grouping of parts of several plots, automation transferring data from databases to plots, search for standards in action sequences, argument values in functions and rules, file data preparation to be used by Prolog programs and KDD programs.

Keyword: Plot, Narrative, Interactive Plot Generator, PlotDBase, Temporal Database, KDD, Knowledge Discovery, Data Mining, Programming in Logic, Prolog.

Resumo:

Informações relacionadas às pessoas e às empresas podem ser agrupadas como narrativas de situações reais e convertidas em enredos, compostos por personagens ou atores, objetos ou recursos, eventos ou condições, seqüências de ações, regras de inferência de objetivos e restrições. Enredos podem ser descritos na formas de predicados lógicos em cláusulas definidas, como usadas em Prolog.

Neste trabalho propomos a criação de PlotDBase, banco de dados relacional de enredos, onde são armazenados os modelos e suas instâncias compostas por conjuntos de termos e das relações entre eles, baseados em critérios e propriedades definidos por usuários especialistas. Os dados que compõem os enredos são provenientes de bancos de dados de aplicações e narrativas diversas.

O PlotDBase é útil para a integração de sistemas aplicativos desenvolvidos para atender áreas específicas e narrativas não padronizadas. É proposto o desenvolvimento de uma ferramenta com funções para entrada, preparação e processamento dos dados tais como: seleção, comparação e agrupamento de partes de enredos, automação de transferência de bancos de dados para enredos, busca de padrões em seqüências de ações, em valores de argumentos de funções e em regras, geração de arquivos a serem utilizados por programas desenvolvidos em Prolog e programas de KDD.

Palavras-chave: Enredo, Narrativa, Geração Interativa de Enredos, PlotDBase, Banco de Dados Temporal, Descoberta de Conhecimento, Mineração de Dados, Programação em Lógica, Prolog.

¹ Este trabalho é financiado pela Eletrobrás com uma bolsa através do FDT - Fundo de Desenvolvimento Tecnológico e pela PUC-RIO através de bolsa parcial da VRAC - Vice-Reitoria Acadêmica.

1 - Introdução

Acontecimentos da vida real são descritos por discursos em linguagem natural gerando narrativas. Após transcrição ou digitalização, os textos podem ser armazenados em documentos e guardados em meios magnéticos ou óticos. Imagens e sons podem ser referenciados nos textos através de ligações (links) para arquivos especializados para tais fins.

A pesquisa de conhecimento em conjuntos de documentos selecionados, pode ser uma tarefa árdua, demorada e com resultados imprecisos. O usuário deverá saber o que procurar, fornecendo conjuntos de palavras chaves e conectivos lógicos entre elas (and, or e not) a uma ferramenta de busca (browser), que encontramos implementada, bem ou mal, em diversos produtos de software.

Neste trabalho, apresentamos uma proposta para armazenamento de informações na forma de enredos possibilitando recuperação posterior da informação e descoberta de conhecimento através de softwares específicos. As informações são principalmente oriundas de bancos de dados de sistemas aplicativos e narrativas de discursos em linguagem natural.

Propomos também a construção de uma ferramenta que auxilie um usuário a implementar e manter os dados referentes ao gênero de sua especialidade.

Entre as funções da ferramenta de armazenamento e busca, apresentamos um método para comparar enredos, possibilitando descobrir semelhanças e diferenças entre eles. Outra função permite a construção de estruturas de dados específicas para descoberta de conhecimento de regras de inferência de objetivos a partir das pré e pós condições de um conjunto de operações executadas em ordem parcial.

É possível aplicar estes conceitos em planejamentos e acompanhamentos de empreendimentos, registros de informações de comportamentos pessoais e verificação de conseqüências de atitudes, questionários de pesquisas qualitativas, narrativas em geral tais como obras literárias, obras científicas, notícias, documentos de processos judiciais, diários de bordo, agendas, atas de reunião, entre muitos outros.

Em qualquer dos gêneros utilizados, fica a critério do usuário especialista, responsável pela definição, implementação, implantação e operação do problema, escolher que informações armazenar, qual o nível de detalhe que lhe interessa e que tipo de processamento para recuperação de informações irá utilizar e com qual finalidade.

A quantidade de informação e nível de detalhamento pode ser o maior possível, mesmo que no momento o usuário não saiba exatamente o que quer recuperar. Pode acarretar apenas problemas físicos de armazenamento (com valor por unidade cada dia mais barato e capacidade cada dia maior) e da velocidade de processamento (que dobra a cada ano e meio). No entanto o valor da informação é cada dia maior, pois não armazenando-a no momento que acontece, estará perdida para sempre, ou custará muito mais caro posteriormente. Estando armazenada, poderemos ou não precisar recuperá-la. Não estando, só podemos não precisar ...

Utilizamos os conceitos teóricos de representação de discurso e da programação em lógica, as idéias do estruturalismo, sua evolução para um gerador interativo de enredo, as técnicas de descoberta de conhecimento e mineração de dados da inteligência artificial, as tecnologias de bancos de dados e conceitos de Engenharia de Software.

Cada frase da narrativa é convertida em um predicado lógico, afirmativo ou negativo, composto basicamente de um verbo ou uma função que relacione objetos. A análise sintática é a divisão de um período em suas orações, com a devida classificação, e de cada oração em seus elementos constituintes, com sua classificação e a das funções dos mesmos, tais como sujeito, predicado, objeto etc. . Desta forma teremos cada narrativa sendo descrita como um conjunto de predicados, podendo ser ordenados cronológica e espacialmente, conforme o texto descritivo do acontecimento real. É importante que o enredo gerado transmita todos os conceitos básicos que o leitor teria a partir da narrativa, sendo no entanto um resumo desta.

O conjunto dos enredos relativos a um gênero, relacionado ao conjunto das narrativas, é armazenado em um banco de dados de enredos específicos, que chamaremos de PlotDBase, como composição dos nomes PLOT (enredo) e DataBase (banco de dados ou base de dados). Sobre estes dados, programas em lógica podem ser programados com o intuito de descobrir conhecimento.

Este trabalho apresenta no capítulo 2 a Motivação para o seu desenvolvimento; no capítulo 3 a descrição sucinta dos conceitos básicos utilizados; no capítulo 4 as propostas do nosso trabalho para a criação de um PlotDBase; o capítulo 5 a proposta de desenvolvimento de uma ferramenta para especialistas. As conclusões e propostas de futuros trabalhos apresentamos no capítulo 6.

2 - Motivação

Hardware/Software

A evolução tecnológica computacional e de comunicações nos últimos anos proporcionou facilidades e capacidade para armazenamento de grandes quantidades de dados em meios magnéticos ou óticos, o intercâmbio através de redes, principalmente na Internet e variadas possibilidades de programação de métodos que processem as informações, salientando a capacidade de multiprocessamento e compressão de dados.

Em 15 anos vimos a evolução dos PC's:

<i>recurso disponível</i>	<i>em 1987</i>	<i>em 2002</i>	<i>crescimento no. vezes</i>
processador	(PC-XT) 4,77 MHz	(Pentium 4) 2,53 GHz	543
memória	640 KB	2 GB	3.276
armazenamento disco (HD)	20 MB	181,6 GB	9.297
armazenamento flexível	(FD) 360 KB	(DVD) 9,6 GB	27.962
rede local	4 Kbps	100 Mbps	25.600
rede comunicação	(telefone) 300 bps	(TV a cabo) 2 Mbps	6.990

fontes: www.intel.com ; www.seagate.com ; www.philips.com ;

Quantidade de Informações

A quantidade de informações disponível sobre um assunto específico pode ser de tal grandeza que sua absorção se torne impossível através de qualquer método conhecido, seja manual ou automatizado.

Como exemplo, imagine-se pesquisar um determinado assunto científico no qual milhares de pesquisadores estejam trabalhando atualmente em todo o mundo. A quantidade de documentos a serem lidos e conceitos a serem absorvidos pode exigir um tempo e esforço que supere qualquer capacidade humana. No entanto, sabemos que muitos documentos referenciam outros e na realidade o número de idéias é bem menor.

Modificar Necessidades

Muitos sistemas aplicativos para empresas continuam sendo desenvolvidos baseados em levantamentos iniciais de dados, nos requisitos de sistemas, funcionais e não funcionais e na análise feita. Uma modificação de interesses ou necessidades de novas informações pode acarretar um trabalho de redesenho, que mesmo com as modernas técnicas da Engenharia de Software, principalmente de reuso de módulos e programação orientada a objeto, envolvem um custo e tempo de implementação por vezes inviável. Muitas vezes, o sistema se torna obsoleto por não atender a novos requisitos funcionais solicitados.

Modificar bancos de dados

O projeto de banco de dados de qualquer sistema aplicativo envolve um trabalho detalhado de modelagem de dados, estruturação, implementação e implantação. Os SGBDs e ferramentas específicas para modelagem, disponíveis comercialmente, auxiliam em muito este trabalho. No entanto, redefinições de dados com o sistema em operação podem ser extremamente trabalhosas ou não apresentarem resultados efetivos em custo/benefício para o usuário.

Aplicativos tornam-se obsoletos

Mesmo que sistemas aplicativos forneçam todas as informações pré programadas, os sistemas ficam falhos quando precisamos armazenar informações e métodos não previstos.

Do passado vem a modificação para o futuro

É importante que informações diretas e indiretas a uma aplicação sejam fornecidas com o propósito de encontrar no futuro, causas de sucessos e fracassos de empreendimentos. Sabemos que todo empreendimento tem chances de sua execução não sair exatamente como planejado. Mas quando se investiga os fatos ocorridos, pode-se aprimorar os procedimentos para evitar os mesmos erros em novos empreendimentos.

Estruturalismo de Propp

A partir de um estudo detalhado de uma centena de contos de fadas russos, Propp [Propp-68] encontrou apenas 31 funções bem definidas. Na época, primeira metade do século XX, cada frase de cada conto foi avaliada e construídas tabelas para compará-las, descobrindo-se ou induzindo situações semelhantes. Com este estudo, mostrou que analisando-se textos

literários, frase por frase, separando-se os personagens, tipos de atuação no conto e funções que executam, pode-se comparar textos literários de um mesmo gênero, e encontrar padrões neles.

Gerador Interativo de Enredos

Tomando-se esta idéia e utilizando-se os recursos dos bancos de dados dedutivos ativos e temporais e lógica temporal, Ciarlini [Ciarlini-99] construiu um Gerador Interativo de Enredos em linguagem Prolog, nomeando as funções proppianas como operações e estabelecendo pré e pós condições para cada uma delas. Complementando o modelo com regras de inferência de objetivos, restrições (constraints) e observações, criou um sistema semi-automatizado para gerar enredos a partir de uma base de dados iniciais, que fornece os personagens e seus estados (condições). Para cada gênero deve ser construída uma biblioteca de planos típicos. O gerador pode reconhecer se um enredo, representado por seus predicados ou cláusulas, pertence ou não ao gênero considerado. O gerador interage com o usuário, sugerindo operações para a construção de um enredo, permitindo que ele aceite-as ou não, e que forneça novas informações ao sistema

Facilidades em SGBDs

Os atuais SGBDs disponíveis comercialmente, oferecem facilidades para implementar bancos de dados ativos, através de triggers - stored procedures, e bancos de dados temporais.

A possibilidade de intercâmbio de informações e de ativação de rotinas implementadas em diferentes linguagens de programação, facilita a construção dos procedimentos de carga de bancos de dados em uma linguagem e a execução em outra.

Discourse Representation Theory

Em [Saurer-93] há uma demonstração de um sistema de dedução natural para a teoria de representação de discurso (DRT). Usa as DRSs - estrutura da representação de um discurso – para representar exatamente o que está na mente do narrador que entende o discurso. No entanto um ouvinte competente do discurso não só entende o que o discurso diz literalmente e explicitamente, mas também o que ele “diz” implicitamente. Assim certas informações podem ser inferidas do discurso.

Planejamento

[Wilensky-83] mostrou que os requisitos para se construir um planejamento são diferentes dos requisitos de entendê-lo. A união de dois planejamentos pode ser uma tarefa bem mais complexa que simplesmente construir um novo planejamento com a união das tarefas dos dois planejamentos originais. Criar um planejamento otimizado envolve modificação de objetivos, cooperação ou competição.

Agentes de Software - KDD – Data Warehouse

Construir ferramentas usando agentes de software para interagir com aplicativos que disponibilizem dados no momento de suas atualizações, podem agilizar o processo de verificação de resultados, baseados na investigação de modelos baseados em Data

Warehouses e processos de pesquisa de Knowledge Discovery and Data Mining. Serão bastante úteis caso possam gerar alarmes em tempo real, possibilitando tomadas de decisão imediatas em casos extremos.

Tendo em vista os recursos disponíveis apresentados acima, viemos através deste trabalho propor soluções que os utilizem, mesclando suas características, tentando obter vantagens no aprimoramento de sistemas de planejamento e acompanhamento de empreendimentos, sejam eles empresariais, pessoais, de pesquisas ou culturais.

Nossa maior motivação é o fato de estarmos em condição de propor uma alternativa para usuários especialistas cujo principal treinamento será construção de modelos ER, entidades e relacionamentos, para definição das estruturas de dados e em programação em lógica, usando-se lógica de primeira ordem e aplicando em programação por cláusulas definidas – utilizada em Prolog.

A busca de conhecimento pode ser encarada como uma área de pesquisa de novos negócios em uma empresa, visando obter possibilidades de respostas para muitos problemas encontrados no dia a dia.

Exemplo 1

Analisando um empreendimento no setor elétrico (construção de uma usina hidroelétrica ou uma linha de transmissão), podemos considerar basicamente que a obra será feita exatamente conforme o projeto aprovado técnica, física, orçamentária e financeiramente. Ou seja, a obra é iniciada se o projeto está aprovado tecnicamente, a obra é viável e todos os recursos físicos podem ser adquiridos, existe disponibilidade orçamentária para a execução da obra no tempo planejado e os recursos financeiros poderão ser disponibilizados a cada etapa concluída.

Para o projeto, planejamento e acompanhamento da obra, diversos sistemas aplicativos podem ser usados, sendo que muitos são pacotes de software disponíveis no mercado.

Todo o planejamento e acompanhamento não isenta a ocorrência de falhas e atrasos no empreendimento, cujas causas podem ser desconhecidas ou apenas imaginadas.

O interesse em descobrir causas de falhas é para evitá-las em planejamentos futuros. Em auditorias podem encontrar os culpados.

Certamente uma das dificuldades na determinação de causas de falhas ou sucessos é a falta de informações. Quando elas estão disponíveis, geralmente faltam métodos para manipular corretamente as informações e obter resultados que possam ser considerados decisivos.

As causas podem ser mais facilmente encontradas caso sejam informados todo o planejamento e projeto da obra, informações físicas, financeiras e orçamentárias de acompanhamento de cada etapa, informações sobre o corpo de trabalhadores da obra, dados meteorológicos diários no local, informações constantes sobre idoneidade das empresas executoras (só estão executando porque foram verificados estes dados na concorrência - mas e durante a obra, a empresa continua idônea?).

Entre os dados de pessoal por exemplo podem ser informados as funções e salários de cada trabalhador (dados de um aplicativo de folha de pagamento), setor em que trabalha, dados cadastrais e de movimentação de pessoal (tais como admissões, demissões, acidentes de trabalho, greves, dias úteis trabalhados), condições ambientais e de higiene do local da obra, atendimento médicos, etc.

Exemplo 2

Um exemplo esclarecedor é o de uma notícia de interesse mundial. Simultaneamente, ela é divulgada por todos os meios de comunicação, em plantões de noticiários, através de emissoras de televisão, sites de notícias na Internet, emissoras de rádio, edições extras de jornais impressos, tornando-se praticamente impossível um só ser humano sobre a face da Terra não ter conhecimento do assunto após poucos dias do feito. Imaginamos aqui que estes seres humanos tenham pelo menos um contato com a civilização (alguém de uma cidade que por sua vez tem ligação com a capital de uma província e por sua vez tenha contato com uma capital de país). Os casos extremos seriam as pessoas que deliberadamente se isolem em lugares sem o conhecimento de qualquer outra pessoa e que não possuam nenhum "aparelho" para comunicação (ondas de rádio, sinais de satélite, telefonia, sinalizador, etc.).

Em poucos minutos, uma quantidade enorme de informações será divulgada.

Um usuário que deseje ter todas as informações a respeito e que disponha de dispositivos para receber as informações, receberá estas informações de forma repetida, ou seja, informações provenientes da mesma fonte ou interpretadas do mesmo modo. Por outro lado, deixará de conhecer diversos detalhes do acontecimento, devido à impossibilidade de rastrear todos os meios de comunicação, da não divulgação intencional de detalhes do acontecimento (como informações filtradas por governos defendendo interesses próprios e que controla o local do acontecimento), de interpretações variadas da notícia por divulgadores, entre outras.

O tempo para absorção das descrições do fato, a memória humana necessária para armazenar todos os detalhes e a capacidade lógica de interpretação, podem tornar uma pequena quantidade de informações em apenas uma noção do que aconteceu, criando dúvidas sobre um ouvinte que precise tomar alguma decisão baseado nos fatos.

Considerando que um acontecimento esteja ligado a um gênero, é de suma importância que acontecimentos semelhantes possam ser localizados rapidamente e que haja possibilidade de encontrar-se as ligações lógicas entre eles de modo a prevenir-se os eventos futuros, baseados em registros de fatos.

Do ponto de vista de um usuário, podemos considerar que acontecimentos da vida real podem ser gerados de diversas formas e estarem ligados a diversos objetivos.

Exemplo 3

Uma notícia do meio político, modificando diretrizes econômicas, pode afetar o planejamento de ampliação de uma indústria, tendo em vista que seu objetivo de aumento de produção para exportação estará comprometido.

3 – Fundamentos Básicos Utilizados

Enredo é definido na segunda acepção no dicionário Houaiss [Houaiss-01] como *sucessão de acontecimentos que constituem a ação, em uma produção literária (história, novela, conto etc.); entrecho, trama.*

A conversão de narrativas em enredos nos permite estabelecer comparações entre as narrativas, através da comparação entre os predicados escolhidos, modificando-se os valores dos argumentos.

Como exemplo,

na narrativa 1: “João casou-se com Maria.”

enredo 1: homem(João). => variável: homem
mulher(Maria). => variável: mulher
casar(João, Maria). => forma geral: casar(homem, mulher).

na narrativa 2: “Finalmente aconteceu o casamento de José e Rute.”

enredo 2: homem(José).
mulher(Rute).
casar(José, Rute).

narrativa 3: “Mr. Smith married Sara”.

enredo 3: man(Mr. Smith). => variável: man
woman(Sara). => variável: woman
marry(man, woman). => forma geral: marry(man, woman).

Comparando-se os enredos 1 e 2 encontramos uma ação semelhante nas duas narrativas, acontecendo com argumentos (variáveis) idênticos e personagens diferentes.

Os enredos 1 ou 2 podem ser comparados ao enredo 3 com o auxílio de uma tabela de correspondência ou tradução de predicados e termos.

O trabalho de transformação de uma narrativa para um enredo pode ser auxiliada por um programa agente de software que aprenda com o usuário termos não previstos e próprios do gênero ao qual pertença a narrativa. Distinguir em frases o que é ação do que é condição (fato), separar os objetos (personagens, recursos, coisas) de suas qualificações.

Para desenvolvimento de nossa proposta e sugestões de projetos especialistas que poderão ser implementados, tomamos inicialmente os conceitos da Morfologia de Contos Maravilhosos [Propp-68].

Propp nos demonstra que a pesquisa minuciosa sobre as funções executadas pelos personagens em uma centena de contos de fadas russos, selecionados entre os de Afanas'ev, podem analogamente à morfologia nas ciências exatas, descobrir que existem seqüências padrões. Assim ele conseguiu sintetizar todas as ações acontecendo em todos os contos com

apenas 31 funções. Como resultado prático, sem acreditar que ele poderia necessariamente querer isto, variando os personagens e os subconjuntos de funções, todos os contos analisados poderiam ser escritos. Houveram críticas ao seu trabalho assim como ao estruturalismo de modo geral. Mas quando foi traduzido o seu livro na Inglaterra em 1968, houve uma divulgação ampla. O seu método pode ser utilizado em conjuntos de obras de outros gêneros.

Na verdade o termo gênero literário encontra até hoje divergências em suas definições. Em [Houaiss-01] encontramos para gênero, na acepção 5 *Rubrica: literatura*.

em teoria literária, cada uma das divisões que englobam obras literárias de características similares [Inicialmente tripartite e já objeto de estudo de Platão e Aristóteles, é com o Romantismo que os estudos sobre os gêneros alcançam maior divulgação, sendo tb. divididos em três: lírico, épico e dramático; no entanto, o problema da classificação dos gêneros permanece com o aparecimento, p.ex., da narrativa, atualmente considerada como um gênero proveniente, segundo alguns, do desenvolvimento do gênero épico.]

Utilizaremos gênero com a acepção 1 *conceito geral que engloba todas as propriedades comuns que caracterizam um dado grupo ou classe de seres ou de objetos* e deixaremos para o usuário definir o grupo e a extensão das propriedades comuns que possam caracterizar.

A criação de enredos de um mesmo gênero irá proporcionando ao próprio usuário especialista uma experiência que resultará cada vez mais num trabalho mais consistente.

Quanto à automação do processo e reconhecimento de uma instância como fazendo parte ou não do gênero, encontramos o Gerador Interativo de Enredos [Ciarlini-99]. Definindo-se previamente uma biblioteca de planos típicos com um conjunto de operações (funções) com pré e pós condições, um conjunto de regras de inferência de objetivos, e outras estruturas, pode ser construído um enredo (ou plano) como uma seqüência de operações em ordem parcial.

4 – Criação de um PlotDBase

Neste capítulo apresentamos uma proposta para construir enredos a partir de narrativas em documentos textos e informações em bancos de dados pertencentes a sistemas aplicativos. Os dados transformados são armazenados nas estruturas do PlotDBase - bancos de dados de enredos. A partir deste, podem-se compor Data Warehouses, para serem processados por ferramentas de KDD - knowledge discovery and Data Mining – e bancos de dados específicos para uso por programas em Prolog, entre eles o IPG – gerador interativo de enredos.

Os dados são armazenados segundo uma identificação do documento original ou do sistema aplicativo e respectivo banco de dados. Cada documento é inicialmente classificado como pertencente a um gênero ou como “sem classe” ou “genérico” se o gênero for desconhecido.

Não estaremos considerando aqui formas gráficas de representação do discurso. No caso de discurso falado, pode-se transformá-lo em uma narrativa em arquivo texto manualmente ou com o uso de um programa de reconhecimento de voz. Alguns existentes são genéricos e exigem que sejam treinados por cada pessoa (cada voz) que o usará. Outros são específicos

para cada área de conhecimento, devido principalmente ao volume, qualidade e tipo de informações de cada área e ao jargão dos discursos.

A escolha do que armazenar é uma decisão do usuário especialista, que será treinado para definir o modelo de dados de enredo. Consideramos que o fato de estarmos transformando dados em enredos, estamos proporcionando ao usuário uma sumarização de suas informações.

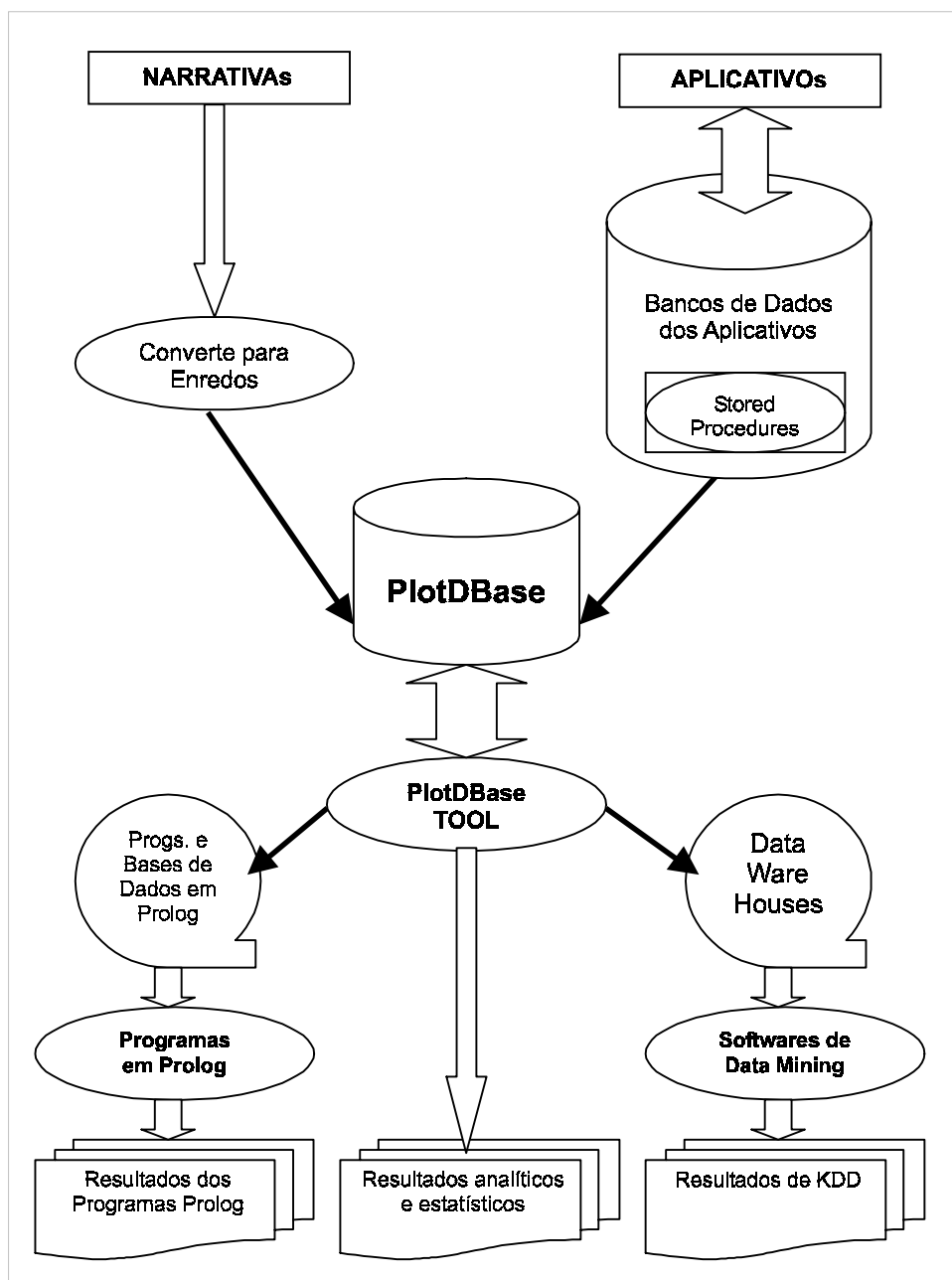


Figura 1 - Fluxo Geral de Processos na Criação do PlotDBase e uso do PlotDBase Tool

Assim fica a seu critério escolher o nível de detalhamento desejado.

A estrutura do PlotDBase permite o armazenamento de:

- gêneros (identifica conjuntos de obras literárias, científicas, casos reais ou quaisquer outros a critério do usuário)
- modelos de enredos, ligado a um gênero, podendo conter estruturas de dados do IPG
- instâncias de enredos definidas conforme o modelo de enredo
- cópias de segurança de enredos
- cópias de enredos, conforme seleção do usuário, para simulações
- enredos resultados de operações lógicas sobre dois ou mais enredos como conjuntos (união, interseção, subtração)
- stored procedures a serem implementadas nos bancos de dados de aplicações
- seleções do usuário

Consideramos importante nesta proposta que o usuário especialista seja o responsável pela definição do modelo e coordenador do trabalho de carga dos dados.

O usuário especialista será treinado para definir o modelo do enredo de acordo com a lógica primeira ordem, conforme cláusulas definidas, como utilizado pelo Prolog e de acordo com as estruturas exigidas pelas ferramentas. Assim, a interpretação do conteúdo será dado pelo próprio usuário que é a pessoa que entende da informação.

Elimina-se deste processo a figura do projetista de banco de dados, que entende a ferramenta – SGBD, aprende sobre o assunto a ser armazenado e define as estruturas. Aqui, o usuário aprende como construir a estrutura.

Quando a origem das informações é um banco de dados de aplicações, espera-se que o projeto de banco de dados relacional tenha sido implementado em um SGBD comercial, conforme modelo entidade relacionamento. A norma SQL-92 apresenta as formas de definição de triggers e stored procedures que já estão implementadas e disponíveis na maioria dos SGBDs.

O ABD - administrador de bancos de dados - disponibilizará ao usuário especialista as estruturas dos bancos de dados de aplicações.

O usuário especialista escolherá as tabelas e campos que deseja copiar do BD e definirá triggers e stored procedures para realizar esta tarefa. Normalmente os triggers serão acionados no momento de uma transação - inclusão, exclusão ou modificação - na base de dados. O trigger aciona uma stored procedure na qual estará especificada a composição e geração de registros do PlotDBase conforme o modelo de enredo escolhido.

Um modelo de enredo permite instanciar os enredos possíveis no mundo fechado que se quer representar. Um modelo contém todos os elementos que poderão ser utilizados nas instâncias.

Ao se criar aplicações convencionais, constrói-se inicialmente um modelo ER do banco de dados. Este modelo depois de validado, implementado e testado, torna-se a base do sistema e não deve sofrer modificações. Após carga inicial do banco de dados, as funções do aplicativo passam a modificá-lo sob operação do usuário. Caso se constate necessidade de modificar a estrutura do banco de dados após carga de dados sabe-se que uma nova carga dos dados será necessária.

A estrutura de dados do PlotDBase permite que se definam vários modelos de enredos e vários enredos por modelo, proporcionando maior flexibilidade ao usuário.

No PlotDBase será permitida a modificação do modelo do enredo, desde que nenhuma instância de enredo a ele subordinada, seja invalidada. Cabe ao usuário escolher: se modificar o modelo deverá transformar dados que se tornarão inválidos.

Utilizaremos a linguagem de primeira ordem para definir o modelo do enredo. O alfabeto da linguagem de primeira ordem consiste de:

- símbolos lógicos:
 - variáveis
 - conectivos lógicos (and, or, not)
 - quantificadores (para todo x tal que ...)
- símbolos não lógicos:
 - constantes – objetos específicos
 - símbolos predicativos – relacionamento entre objetos
 - símbolos funcionais – funções sobre objetos

As sentenças de uma linguagem de primeira ordem são:

- termo – é uma variável, uma constante ou uma expressão da forma $f(t_1, \dots, t_m)$ onde f é um símbolo funcional admitindo m argumentos e t_1, \dots, t_m são termos.
- fórmula – é ou uma fórmula atômica da forma $p(t_1, \dots, t_n)$ onde p é símbolo predicativo admitindo n argumentos e t_1, \dots, t_n são termos ou é uma expressão obtida compondo-se fórmulas através dos conectivos lógicos ou prefixando-se fórmulas com quantificadores.

Os termos permitem denotar objetos do domínio do problema diretamente através de seus nomes ou de variáveis ou indiretamente através de funções.

As fórmulas atômicas permitem expressar relações existentes entre objetos do domínio do problema.

As fórmulas não atômicas permitem expressar propriedades das relações sobre o domínio do problema.

Um programa em linguagem de primeira ordem é qualquer coleção finita de fórmulas de 1ª ordem e uma consulta é qualquer fórmula de 1ª ordem exprimindo as condições a serem satisfeitas por uma resposta correta.

Na notação Prolog, os símbolos lógicos a seguir são representados como:

<i>Notação Lógica</i>	<i>Prolog</i>
conjunção (and)	“,” (vírgula)
disjunção (or)	“;” (ponto e vírgula)
negação (not)	“not” (functor not)
cadeias de caracteres (átomos)	começam com letras (a-z) minúsculos ou entre ‘

literais)	(aspas simples) quando possuírem caracteres especiais ou começarem por maiúscula.
variáveis anônimas	“_” (subinhado)
variáveis	começam com letras maiúsculas (A-Z)

O IPG definido em [Ciarlini-99] consta das seguintes estruturas de dados para definir um enredo:

- **termos** – predicados que descrevem os objetos (o que não é ação ou operação) tais personagens, recursos, coisas, localizações, etiquetas de tempo, etc.
- **funções** – qualifica os objetos.
- **operation** – descreve uma operação válida que pode ocorrer no enredo:
operation (identificador, nome, lista de pré condições, lista de pós condições, custo, lista de efeitos primários, lista de operações componentes, lista de pares de ordem dos componentes).
- **hierarquia de generalização** – na forma:
is_a (operation_i(arg₁, ..., arg_m), operation_j(arg₁, ..., arg_n)).
- **especificação de restrição de integridade** – na forma:
integrity_constraint (lista antecedente de fórmulas básicas de regras, lista conseqüente de fórmulas básicas de regras).
Cada uma das listas é uma conjunção de regras relacionadas contendo especificações de condições no tempo. São utilizadas os seguintes tipos de especificações, através dos metapredicados **h**, **p**, **o**, **e** que significam respectivamente hold, possibly, occurred, established:
 - h (t, cond) - a condição cond é necessariamente válida no tempo t
 - h (cond) - restrição obrigatoriamente verdadeira
 - p (t, cond) - a condição cond é possivelmente válida no tempo t
 - p (cond) - restrição possivelmente verdadeira
 - o (t, cond) - a condição cond ocorreu no tempo t
 - e (t, cond) - a condição cond tornou-se válida no tempo t
- **lista de prioridades de pré condições** – define uma lista de condições agrupadas por prioridades:
critical_list (lista de listas do tipo [prioridade, condição, ..., condição])
- **regra de inferência de objetivos** – na forma:
rule (antecedente da regra, conseqüente da regra, tipo do objetivo)

o antecedente é uma lista conjunção de fórmulas básicas de regras, cada uma relacionadas a um tempo conforme lista mostrada anteriormente.

o conseqüente da regra é uma conjunção de fórmulas atômicas positivas do tipo “h” de um ou dois argumentos, representada por uma lista.

tipo do objetivo inferido pela regra. Pode ser *limited*(LIMIT), *condicional*(COND) ou *normal*.

- **observação** – usada para utilização do reconhecedor e do simulador do IPG representada em listas de objetivos condicionais e limitados.

Outras estruturas são utilizadas internamente pelo IPG e não vamos mencioná-las aqui.

A instanciação de enredos só é possível escolhendo um modelo de enredo já definido. No entanto, o usuário poderá optar por definir ou completar o modelo do enredo no momento da instanciação dos casos.

Desta forma, permite-se que o modelo de enredos evolua independentemente dos dados já informados. Ao se modificar um modelo, a ferramenta irá validar os dados já informados, evitando-se inconsistência. Caso o usuário queira poderá atualizar os dados já instanciados conforme o novo modelo ou aplicá-lo somente aos novos dados. Da mesma forma, os dados anteriores podem ser mantidos caso sejam mantidas definições do modelo anterior.

Formas de definição de enredos

Dependendo do detalhamento desejado e das informações que se desejam obter posteriormente, são usadas formas distintas de definir os modelos de enredos e instanciá-los.

A **forma completa** é feita analisando-se todo o documento original e selecionando os objetos, os fatos e operações, definindo completamente o modelo. Pode-se até criar as regras de inferência de objetivos e demais elementos do modelo. Somente com o modelo completo inicia-se a instanciação do enredo.

A **forma direta** é utilizar a ferramenta para ir definindo o modelo de enredo no momento de sua instanciação. Neste caso podem ser criados os objetos, os fatos e as operações no momento que aparecem, sem se preocupar com definições de pré e pós condições. A definição da operação no modelo apenas com sua identificação e argumentos permite que seja utilizada para instanciar:

operation (id, nome_operação (Var₁, ..., Var_n), [], [], 0, [], [], []).

A ferramenta pode auxiliar o usuário a completar as definições do modelo tais como pré e pós condições de operações, ordem das operações, regras, restrições e prioridades.

A **forma indireta** é permitir a instanciação de fatos e operações sem ter definido nenhum modelo. Após a entrada do enredo, a ferramenta oferece ferramentas ao usuário para listar todos os seus componentes e auxiliar na definição do modelo. Um programa desenvolvido em Prolog pode generalizar a partir dos valores dos argumentos de operações e de fatos com mesmo nome – no SICStus Prolog, a instrução utilizada é *subsumes/2*: (*subsumes*(?General, ?Specific)). Com estes resultados o usuário define o modelo e submete a instanciação a uma verificação completa de integridade.

Fatos e Operações executadas no tempo e espaço

Um fato pode ser bem definido no tempo e no espaço, se ele é armazenado juntamente com o tempo inicial, tempo final e localização.

O tempo inicial é o instante de tempo em que o fato começa a ser válido. O tempo final é quando o fato deixa de ser válido.

A localização designa a posição física onde o fato é válido, por exemplo, uma cidade.

Uma operação é uma ação que é iniciada num tempo inicial, tem uma duração intermitente e termina num tempo final. É executada em uma localização que designa uma região.

Uma localização pode representar um conjunto de localizações, por exemplo, uma região geográfica composta por mais de um estado da federação ou por várias cidades de diferentes estados.

Um fato que ocorra ao mesmo tempo em duas localizações diferentes e que não formem uma identidade geográfica, pode ser representado por dois predicados, com os mesmos tempos e cada um com sua localização, ou com um predicado e uma lista de localizações válidas.

Ao definir um enredo com condições (fatos) e operações já executadas ou em execução, o usuário pode querer manter a ordem cronológica que está no documento original ou banco de dados. Uma outra informação importante podem ser as localizações onde aconteceu ou podem acontecer o fato ou a operação. Isto possibilita a definição de regras que valham em determinado local e não valham em outro.

Um predicado que mantém a cronologia e localização pode ser da forma:

insert operation (T_i , T_f , Loc, $op(t_1, \dots, t_m)$)

insert fact (T_i , T_f , Loc, $c(t_1, \dots, t_m)$)

onde

- T_i é uma etiqueta de tempo (p.ex. data/hora) ou um tempo relativo 'tx-2d' = tempo definido em tx menos 2 dias, para indicar o momento de início da validade da condição ou do início da execução da operação.
- T_f é uma etiqueta de tempo (p.ex. data/hora) ou um tempo relativo 'tx-2d' = tempo definido em tx menos 2 dias, para indicar o momento de término da validade da condição ou do término da execução da operação.
- Loc é um literal ou uma lista de literais que define localizações onde é válida ou aconteceu a condição ou operação.
- $op(t_1, \dots, t_m)$ é o predicado definido para uma operação com os seus m valores de argumentos t_1, \dots, t_m .
- $c(t_1, \dots, t_m)$ é o predicado definido para uma condição com os seus m valores de argumentos t_1, \dots, t_m .

Há também a possibilidade de não definir uma das variáveis tempo ou localização. Neste caso, o argumento é substituído por 0 (zero) para tempo ou [] (lista vazia) para localização:

insert operation (T_i , T_f , [], $op(t_1, \dots, t_m)$) - válida em qualquer localização

insert fact (T_i , 0, Loc, $c(t_1, \dots, t_m)$) - válida desde um tempo inicial

insert operation (0, 0, [], $op(t_1, \dots, t_m)$) - operação executada onde tempo inicial, tempo final e localização não interessam.

insert fact (0, 0, [], $c(t_1, \dots, t_m)$) - o fato é válido em qualquer tempo e lugar.

Funções de apoio na definição de dados

A ferramenta de criação de enredos disponibiliza para o usuário algumas funções úteis durante as entradas de dados:

- listar os elementos de um enredo em ordem (por etiqueta de tempo, localização, predicados, etc.), permitindo escolher um elemento já definido;
- pesquisar seqüências de operações e/ou condições padrões em um enredo, possibilitando definição de uma operação complexa (formada por outras operações);
- criar um modelo de enredo a partir de uma cópia de um modelo de enredo já definido
- comparar dois enredos do mesmo gênero. Encontrar seqüências padrões e permitir a inserção do conjunto como uma operação complexa em um modelo;
- listar todos os valores dos parâmetros utilizados nos predicados de um enredo;
- preparar um arquivo com valores de parâmetros existentes em um enredo; calcular estatísticas de ocorrências de valores individuais e em tuplas – poderá ser utilizado por programas de KDD;
- selecionar predicados (condições ou operações) e valores de parâmetros para criar um sub-enredo;

Busca de conhecimento

Na busca de conhecimento, alguns algoritmos são não polinomiais, o que pode ser impossível chegar a um resultado. Normalmente isto se deve ao número de variáveis de pesquisa.

A ferramenta a ser desenvolvida irá permitir a seleção de valores de variáveis para que seja possível obter determinados resultados, quando não se pode obtê-los completamente. Qualquer seleção feita fornecerá a priori o tempo estimado de processamento para que o usuário decida pela execução ou não do procedimento. Há também possibilidade de interrupção do processamento exibindo-se os resultados até o momento.

A comparação entre enredos longos pode não ser viável em função do tempo de processamento exigido. Para permitir que qualquer par de enredos possa ser comparável, pode-se atribuir valores (graus) a predicados e solicitar a ferramenta que produza um enredo simplificado (uma síntese do detalhado), com os predicados de grau igual ou superior a um valor informado.

Outra possibilidade é compor operações complexas a partir das operações básicas, criando uma hierarquização. Uma pesquisa inicial sobre cada enredo no PlotDBase, medindo a frequência de seqüências de 2, 3, ..., n operações, onde n será também limitado pela capacidade de processamento. Baseado na ordem das seqüências mais freqüentes o usuário escolhe o que deve hierarquizar, criando as operações complexas. Pode ser gerado uma nova versão considerando a v.1 como a primeira instância informada.

Obtenção de Informações em Bancos de Dados de Aplicações

Normalmente as aplicações são pouco suscetíveis de sofrer modificações para exteriorizar informações.

No caso de uma aplicação ser suportada por um SGBD que atenda aos requisitos do SQL-92, há possibilidade de implementar triggers e stored procedures que serão ativadas no momento de uma transação. Cabe ao especialista buscar o conhecimento das estruturas do banco de dados e das transações que são atualmente executadas e definir o procedimento de geração de informações para serem armazenadas no PlotDBase. Um arquivo intermediário pode ser utilizado, neste caso como um Log de cada banco de dados.

Como exemplo, um sistema de recursos humanos registra todas as transações referentes aos empregados da empresa. Ao incluir as informações sobre o novo empregado no banco de dados, uma stored procedure cria um registro no Log da operação realizada, na forma de um predicado:

hire (jorge, 20, carpinteiro, 1)

como instanciação do predicado definido no modelo como:

hire (employee, salary, function, level)

Para registrar a execução da ação, a operação é inserida com um tempo inicial t_i , 0 (zero) como tempo final, loc1 como uma lista de localizações e o predicado hire e valores dos argumentos:

```
is_a(ti, time).
ti= '19/01/2002-08:00'.
is_a(loc1, localization).
loc1 = ['Rio de Janeiro'].
```

insert_operation (t_i , 0, loc1, hire (jorge, 20, carpinteiro, 1))

A operação hire foi definida com as seguintes pré condições:

```
not (is_a ( person, employee))
has_a ( person, function )
```


comparados e então encontraremos condições comuns aos planos originais. Desta comparação podemos deduzir semelhança de condições em dois planos distintos. Isto implica em encontrar objetivos semelhantes em enredos distintos, o que permite determinar regras de inferência de objetivos.

Informações importantes podem ser obtidas na junção de dois enredos, estruturados e modelados conforme apresentado. Entre elas, fatos que aconteceram em dois lugares e na mesma época podem se repetir, mostrando que alguns padrões podem ser comuns.

Uma aplicação interessante é a junção da ficção com a realidade. Um conto pode ser enriquecido com dados reais, tornando-o mais atrativo e educativo.

5 – Funções da Ferramenta de Apoio aos Usuários Especialistas

Neste item vamos apresentar funções de uma ferramenta de uso geral para a construção dos enredos, a partir de dados armazenados em bancos de dados de aplicações ou a partir de documentos textos que descrevem narrativas.

O PlotDBase armazenará os modelos e instâncias dos enredos. Estas definem:

- **termos** – predicados que descrevem os objetos utilizados nos enredos:
 - Q** conjunto dos literais qualificadores usados nos modelos que definem os tipos de argumento dos predicados das condições e das operações. As funções serão usadas para relacionar cada literal com o qualificador. Entre eles: personagem, recurso, tempo, intervalo_de_tempo, localização.
 - P** conjunto de literais usados para instanciar os enredos indicando os personagens
 - R** conjunto de literais usados para instanciar os enredos indicando os recursos e coisas
 - AT** conjunto das etiquetas de tempo usados para instanciar os enredos indicando tempo absoluto (data / hora)
 - RT** conjunto das etiquetas de tempo usados para instanciar os enredos indicando tempo relativo que referencia tempo absoluto ($rt=f(at)$). Operações de soma ou subtração de um valor tempo absoluto representado por um literal.
 - DT** conjunto das etiquetas de tempo usados para instanciar os enredos indicando intervalo de tempo.
 - L** conjunto das localizações usadas para instanciar os enredos.

- **funções** - qualificam os valores dos argumentos de acordo com os qualificadores e estabelecem os fatos ou condições do enredo:

qualificador (literal). - designa que o literal é do tipo **qualificador**

is_a (literal, qualificador). - corresponde a **qualificador** (literal).

is_on (literal, localização). - indica que literal está em localização.

COND conjunto das funções que representam as condições (também chamados fatos ou estados) dos objetos do enredo.

- **operation** - descreve uma operação válida que pode ocorrer no enredo na forma:

operation (identificador, nome, lista de pré condições, lista de pós condições, custo, lista de efeitos primários, lista de operações componentes, lista de pares de ordem dos componentes).

OP conjunto dos predicados que representa operações definidas

(OP x S) descrição de cada operação com suas pré e pós condições

(OP x OP) conjunto das operações complexas, formadas por mais de uma operação

- Definição de estruturas para uso pelo IPG gerador interativo de enredos:

HG conjunto das hierarquia de generalização na forma:
is_a (operation_i(arg₁, ..., arg_m), operation_j(arg₁, ..., arg_n)).

RO conjunto das regras de inferência de objetivos na forma:
rule (antecedente da regra, conseqüente da regra, tipo do objetivo)

RI conjunto das restrições de integridade na forma:
integrity_constraint (lista antecedente de fórmulas básicas de regras, lista conseqüente de fórmulas básicas de regras).

LP conjunto das listas de prioridade
critical_list (lista de listas do tipo [prioridade, condição, ..., condição])

OB conjunto das observações (listas de objetivos condicionais e limitados)

Pode ser adotada uma nomenclatura única de qualificadores para qualquer enredo, o que facilitará as operações sobre conjuntos de enredos. Uma recomendação é utilizar nomes em língua inglesa, por ser de uso universal, evitando-se tabelas de correspondência

Funções da ferramenta:

- definir Gêneros
- definir Modelo de Enredo em cláusulas definidas (termos, funções , etc.) e segundo estruturas utilizadas por programas tais com o IPG
- instanciar enredos conforme o modelo definido pelo usuário
- permitir modificação do modelo de enredo durante a entrada de dados de instâncias, complementando-o.
- verificar consistência dos enredos já armazenados após modificação de modelo, de modo a manter a integridade dos dados
- construir modelos e enredos a partir de dois existentes, usando conceitos de união, interseção e subtração de conjuntos
- pesquisar e exibir para o usuário os conjuntos de valores dos argumentos utilizados pelos predicados nos enredos
- definir seleções de critérios considerando os símbolos definidos no modelo e os valores de argumentos de um conjunto de enredos
- construir programa em lógica e banco de dados específico a ser utilizado por um IPG – gerador interativo de enredos – escrito em Prolog
- traduzir modelos e instâncias de enredos, considerando tabela de correspondência (por exemplo para permitir comparar enredos de narrativas escritas em línguas diferentes)
- construir arquivos e Data Warehouses específicos para ferramentas de Data Mining a partir de seleções de critérios definidos
- permitir a construção e atualização de bibliotecas de planos típicos (utilizadas pelo IPG)
- criar triggers e stored procedures para serem exportadas para os SGBDs das aplicações. As stored procedures gravarão no Log predicados nos formato definidos pelas estruturas de dados, no momento da atualização do banco de dados da aplicação.
- gerar relação de condições a partir das definições das operações (pré e pós condições) e ordem cronológica
- analisar um enredo e verificar conjuntos compartilhados de parâmetros, apresentando os conjuntos ordenados de parâmetros. Possibilita a escolha do conjunto para análise de frequência estatística e descobrimento de planos típicos.

- calcular estatisticamente a frequência em vários exemplos no mesmo gênero, selecionando as possibilidades mais frequentes de inferência de objetivos
- agentes de software são ativados no momento de atualização de bancos de dados de aplicações para executar procedimentos de estatística, seleção de dados frequentes e verificação de seqüências padrões em vários gêneros

6 - Conclusões e Futuros Trabalhos

As respostas que precisamos podem vir das informações que armazenamos no passado e dos procedimentos de descobrimento de conhecimento sobre elas.

A escolha do que armazenar pode vir da experiência, o que pode acarretar perdas de informações enquanto aprendemos esta lição.

No entanto, olhando para nós mesmos, quais informações coletamos em nossa vida e para que serviram? Como determinar todas as causas dos fatos que hoje nos ocorre, baseado nas informações que coletamos durante o tempo que já vivemos? Quanto de informação temos armazenado?

Algumas destas perguntas ainda continuam sem respostas. Outras não conseguimos responder com toda certeza.

Em sistemas de informação, armazenar “tudo” pode ser uma idéia trabalhosa e cara, mas talvez a ideal.

Processar “todas” as informações para encontrar conhecimento certamente terá mais trabalho perdido que resultados palpáveis.

Portanto, melhorar os métodos pode gerar menos desperdício e mais resultados. Dos usuários espera-se que adquiram intimidade com os dados referentes à sua especialidade, tenham a intuição de coletar outras informações que poderão ser pertinentes e conheçam bem os métodos para processá-los e descobrir o conhecimento que possa ser útil ao seu negócio.

Apresentamos neste trabalho uma proposta para armazenar informações de várias fontes, bancos de dados de aplicações e narrativas, nas estruturas propostas para enredos, em um PlotDBase, banco dados de enredos, usando notações da programação em lógica, e obtendo conhecimento através da interação e iteração do usuário com ferramentas inteligentes.

Sabemos da dificuldade que será o treinamento de usuários e dos programas. Mas acreditamos na facilidade que trará para a descoberta de conhecimento.

Entre os trabalhos a serem desenvolvidos futuramente está a implementação de um protótipo da ferramenta, criação de um PlotDBase com dados de uma empresa onde haja potencial para uso de KDD, selecionar e treinar com software de KDD para uso com os dados selecionados, e definição e aplicação de treinamento a usuários responsáveis pelos sistemas.

Agentes de software podem ser construídos para aprendizado e geração automática de enredos a partir de narrativas, com técnicas de análise de textos especialistas.

Um estudo de compressão de dados armazenados no PlotDBase é aconselhável.

REFERÊNCIAS

- [Casanova-87] Casanova, M.A., Giorno, F.A. e Furtado, A.L. – Programação em Lógica e a Linguagem Prolog – Ed. Edgard Blücher Ltda – São Paulo, 1987.
- [Ciarlini-99] Ciarlini, A.E.M. – Geração Interativa de Enredos – Tese de Doutorado – Departamento de Informática – PUC-RIO – abril 1999.
- [Fayyad-96] Fayyad, U. M.; Piatetsky-Shapiro, G.; Smyth, P. and Uthurusamy, R. – Advances in Knowledge Discovery and Data Mining – AAAI Press / The MIT Press – California, 1996.
- [Furtado-97] Furtado, A.L. and Ciarlini A.E.M. – Plots of Narratives over Temporal Databases – Departamento de Informática – PUC-RIO - PUCRioInfMCC 6/97
- [Heuser-99] Heuser, C.A. – Projeto de Banco de Dados – Ed. Sagra Luzzato – Porto Alegre, 1999.
- [Houaiss-01] Dicionário Eletrônico da Língua Portuguesa – versão 1.0 – Dezembro 2001 – Instituto Antônio Houaiss – Editora Objetiva Ltda.
- [Macedo-01] Macedo, L.M. – Uso das Técnicas de Análise Morfológica Literária para Descrições de Situações no Mundo Real – Trabalho final apresentado disciplina Tecnologia e Subjetividade – Departamento de Psicologia – PUC-RIO, dez 2001.
- [Passos-00] Passos, E.P.Lopes e Macedo, L.M. – Teoria e Prática de Data Mining para Criação de Regras de Inferência de Objetivos – Technical Notes in Computational Intelligence, ICA, Departamento Engenharia Elétrica – PUC-RIO, dez 2000.
- [Propp-68] Propp, V. I. – Morphology of the Folktale – translation by Laurence Scott. Austin: University of Texas Press, 1968.
- [Propp-84] Propp, V. I. – Morfologia do Conto Maravilhoso – tradução do russo por Jasna Paravich Sarhan – Forense Universitária, Rio de Janeiro, 1984.
- [Ramalho-99] Ramalho, J.A. – SQL Server 7 – Iniciação e Referência – Makron Books do Brasil Ed. Ltda. – São Paulo, 1999.

- [Russel-95] Russel, S. and Norvig P. – Artificial Intelligence A Modern Approach – Prentice Hall – New Jersey, 1995.
- [Saurer-93] Saurer, W. – A Natural Deduction System for Discourse Representation Theory. in Journal of Philosophical Logic 22: 249-302, 1993 –Kluwer Academic Publishers, Netherlands.
- [Weiss-98] Weiss, S. M. and Indurkha, N. – Predictive Data Mining A practical guide – Morgan Kaufmann Publishers, Inc. – San Francisco, 1998.
- [Wilensky-83] Wilensky, R. – Planning and Understanding. Addison-Wesley Publishing Company, 1983.