Uma Abordagem para a Descoberta de Conhecimento em Processos de *Workflow* em Oracle

Rafael S. Garcia, Duncan D. Ruiz

Pontifícia Universidade Católica do Rio Grande do Sul – Faculdade de Informática

Caixa Postal 90619-900 Porto Alegre – RS – Brasil

rafael.sgarcia@terra.com.br, duncan@inf.pucrs.br

Resumo. *Este trabalho apresenta uma nova abordagem para implementação de uma ferramenta Web, plataforma Java-Tomcat, que define uma arquitetura para execução de processos de descoberta de conhecimento sobre base de dados de execuções de Workflow. Nessa arquitetura, algoritmos de mineração e métodos de visualização podem ser dinamicamente integrados, através da redefinição de classes e respectivos métodos. A contribuição está em permitir realizar análises do comportamento de execuções de processos, empregando diferentes configurações, de acordo com a conveniência e disponibilidade dos artefatos (algoritmos e métodos de visualização) disponíveis.*

Abstract. *This work introduces a new approach for the implementation of a KDD (knowledge-discovery in databases) web tool on Java-Tomcat platform. It is defined an architecture for the execution of knowledge discovery processes on workflow execution databases. In this architecture, mining algorithms and visualization methods can be dynamically plugged by proper redefinition of classes and their methods. The main contribution is permitting the analysis of process execution behaviors, considering different configurations, according to the availability and adequacy of artifacts (algorithms and visualizing methods).*

1. Introdução

Atualmente, com a necessidade cada vez maior da informatização e automação das grandes organizações, a automação de processos de negócio através de alguma ferramenta de *workflow* vem se tornando uma prática bastante comum. Diversos processos presentes no dia-a-dia de uma grande empresa podem ser automatizados como, por exemplo, a solicitação e aprovação de férias, viagens e compra de produtos, a definição das atividades que um colaborador deve desenvolver dentro de um determinado projeto ou a identificação de um erro em um aplicativo (*bug*) e as etapas necessárias para a sua correção.

Alguns desses processos podem ser de grande importância estratégica dentro das organizações e os gestores necessitam de ferramentas adequadas para que possam realizar determinadas análises sobre os mesmos. Dentre as dúvidas mais freqüentes dos gestores estão perguntas como: "Qual o processo que mais atrasa?", "Porque este processo tem um índice muito grande de atrasos?" ou "Qual atividade que está levando mais tempo para ser executada do que foi planejado?".

A maioria das ferramentas fornecidas juntamente com os Sistemas de Gerência de Workflow (SGWf), como o *Oracle Workflow Monitor*, não respondem estas perguntas de maneira adequada. Nestas ferramentas, o máximo de interação permitido é fazer um acompanhamento gráfico ou textual do processo, verificando qual recurso o

processo ocupa atualmente, qual o resultado obtido em uma determinada atividade, qual o tempo total de execução do processo, entre outros. Nas organizações, o número de instâncias de processos pode assumir grandes proporções rapidamente. Nestes casos, se um gestor necessitar de uma análise sobre os dados de execuções de processos, terá que fazer uma contagem manual dos processos ou fazer consultas SQL complexas sobre os *logs* da aplicação, ambas as opções de difícil operacionalização.

Nesse sentido, as tecnologias de *Workflow* [Ley00] e de processo de Descoberta de Conhecimento em Banco de Dados (KDD) [Han01] vêm crescendo em visibilidade, projeção e destaque tanto nos meios acadêmicos quanto nos corporativos devido às vantagens e facilidades que elas fornecem. O entusiasmo em torno da união destas duas áreas deve-se, principalmente, ao potencial em agilizar e qualificar a maneira como os gestores gerenciam seus processos de negócio [Bon01].

Este trabalho apresenta uma nova abordagem para a definição de uma arquitetura para a execução de KDD sobre base de dados de *Workflow*, modelados e executados sobre plataforma *Oracle-Oracle Workflow* (OWF) [Cha02]. A idéia é possibilitar aos gestores uma forma visual de análise sobre seus processos de negócio. O ponto positivo desta arquitetura é a facilidade de inclusão de novos algoritmos de mineração de dados e de modos de visualização de resultados no sistema, via a extensão, em Java, de uma classe especial, utilizando a interface proposta. Este trabalho é resultado da pesquisa para a especificação e implementação completa deste ambiente computacional desenvolvido na PUCRS-FACIN ([Gar05, Gar05a]).

2. A arquitetura proposta

Para a especificação da arquitetura desta ferramenta de apoio a análise analítica sobre dados de execução de processos de *Workflow*, iniciou-se identificando todas as etapas necessárias para a realização do processo de KDD, conforme proposto em [Han01]. O SGWf escolhido foi o implementado pela *Oracle*: OWF. A sua escolha baseia-se, na sua grande aceitação no mercado e na sua arquitetura desenvolvida ter certa conformidade com os padrões propostos pela WfMC [Hol95].

A arquitetura proposta é composta por três módulos, conforme ilustrados na Figura 1: Módulo de Importação dos Dados (MID), Módulo de Análise dos Dados (MAD) e Módulo de Visualização (MV). Um quarto módulo é exibido e identificado como *Oracle Database 9i* e representa o SGBD *Oracle*, com uma instalação do *Oracle Workflow 2.6.2* contendo todos os logs de execução de instâncias de processos. No MID é feito o pré-processamento de dados, e é apresentado na seção seguinte. O MAD é tratado na seção 4 e, na seção 5, é tratado o MV.

3. Pré-Processamento de dados

O MID é responsável pelo pré-processamento (extração, transformação e carga) dos dados que representam a execução dos fluxos de *Workflow*. Segundo [Han01] esta é uma das fases mais importantes do processo de KDD, consistindo em mais de 50% do tempo utilizado. Nesta etapa os dados a serem minerados são extraídos da base original e passam por processos de avaliação da necessidade de limpeza de ruídos, padronização, integração e transformação. Exemplos de operações desta etapa são a definição de estratégias para tratar atributos com valores nulos ou com valores que representam a mesma informação, escritos de maneiras diferentes, i.e., Porto Alegre e Poa. Um trabalho realizado de forma incorreta nesta fase pode resultar em padrões inconsistentes ou inválidos que só serão descobertos após a análise e interpretação dos resultados.

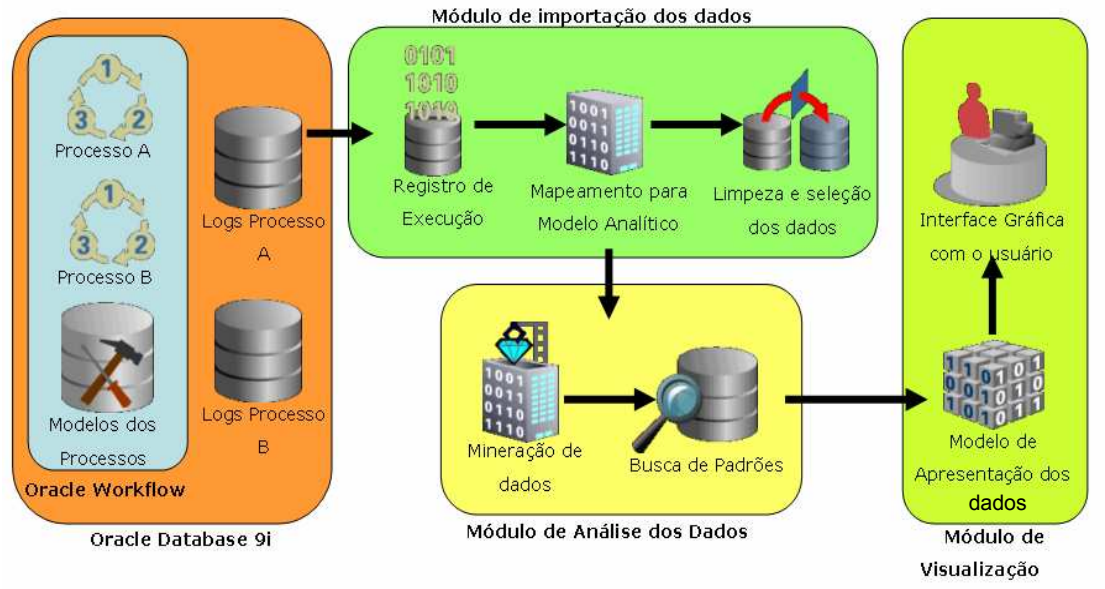

Figura 1 - Arquitetura proposta

No modelo de implementação do OWF tem-se a descrição e o registro de execução (ou Trilha de Auditoria) dos processos de negócios modelados e executados nesta ferramenta. Uma das maneiras para que os dados contidos nele possam ser analisados por processos de descoberta de conhecimento, é organizá-los, de forma analítica, utilizando algum Modelo Multidimensional. Para a arquitetura proposta neste trabalho, foi adotado o modelo desenvolvido e implementado pela HP [Gri04], mostrado na Figura 2.

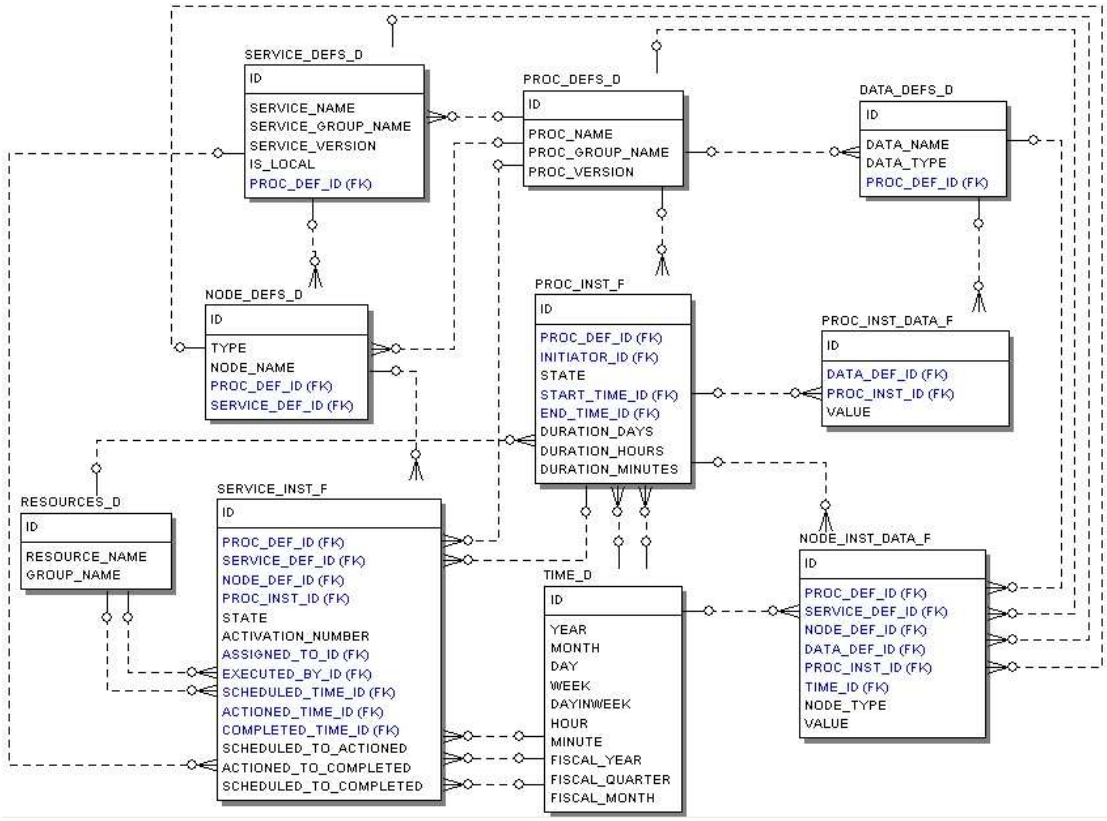

Figura 2 – Modelo Analítico adotado

A escolha deste modelo baseia-se no fato de já ter sido criado focado no armazenamento de modelos e dados de execução de processos de *Workflow*, e dele melhor se adequar ao modelo de implementação do OWF do que o outro modelo testado, proposto em [Ede02]. O ponto positivo em se utilizar um modelo analítico é permitir que a arquitetura possa ser livre quanto ao SGWf utilizado, desde que existam módulos de carga apropriados para tanto.

Para cada tabela que compõe o modelo analítico, foi adotada uma determinada política de importação baseada na procedência dos dados recuperados a partir do modelo do OWF. No modelo do OWF não existe uma distinção entre definições de nodos e serviços, como proposto pelo modelo multidimensional. Logo, as duas dimensões definidas (*Service_Defs_D* e *Node_Defs_D*) armazenam informações sobre as mesmas coisas, ou seja, as definições de atividades. A Tabela 1 descreve as dimensões e fatos do modelo analítico adotado. A Figura 3 mostra a tela de execução da etapa de pré-processamento de dados, denominada "*Execução do Processo de Importação*" por fazer analogia a importar dados de uma base de dados para outra. Na mesma são mostrados, gradativamente, os passos do processo sendo executado.

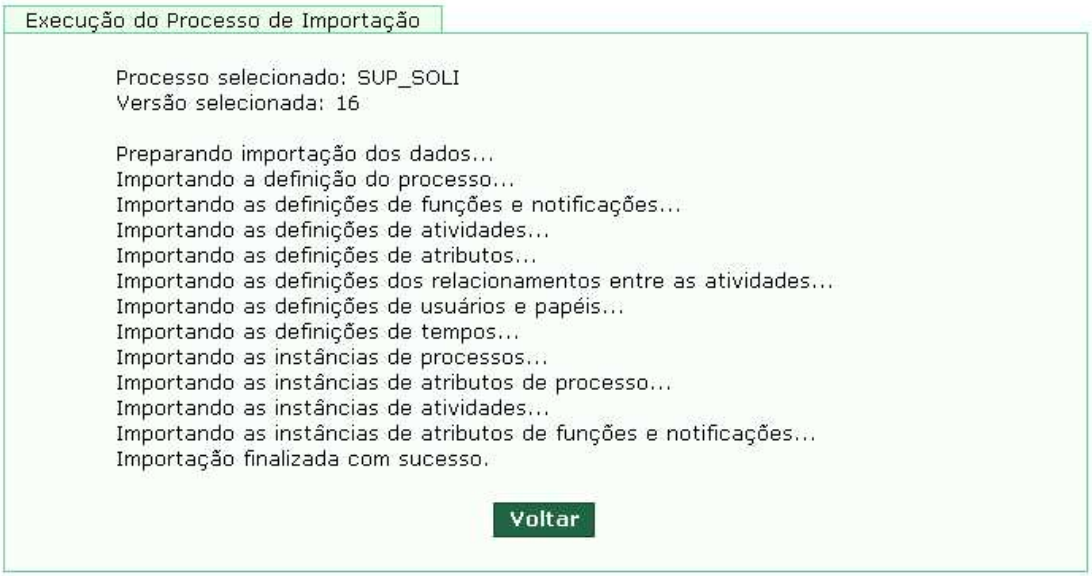

Figura 3 - Execução do processo de importação de dados

4. Mineração de dados

O MAD é responsável pela execução dos algoritmos de mineração de dados e busca por padrões nos processos, ou seja, pela análise e classificação dos resultados da execução destes algoritmos. Ele foi projetado de forma a permitir uma boa infra-estrutura para a incorporação de ferramentas e algoritmos já existentes ou para desenvolvimento de novos algoritmos de mineração de dados especificados pela literatura.

A incorporação de novos algoritmos está desenvolvida para ser genérica, de forma que novos algoritmos possam ser naturalmente adicionados. Esta incorporação é feita através da carga de um arquivo contendo a sua implementação em Java (arquivo *jar*). Para tanto, foi especificada uma interface padrão e cada implementação de algoritmo deverá ter uma classe principal que implemente essa interface. A partir dela, as operações podem ocorrer de maneira automática na ferramenta, bastando apenas um cadastro do novo algoritmo e importação da sua implementação. A interface proposta é chamada *AlgorithmInterface*, com um único método: *executeAlgorithm*. Sempre que um

algoritmo é selecionado para ser executado na ferramenta, o módulo correspondente é carregado automaticamente através do mecanismo de importação de classes Java e uma conversão dinâmica para a interface implementada é realizada. O sistema, então, executa o método implementado para a execução do algoritmo.

Tabela 1 – Descrição das dimensões e fatos do modelo analítico adotado

Tipo	Nome	Descrição
Dimensão	*Proc_Defs_D*	Definição de modelos de processos. São importadas as definições de todos os processos definidos pelo OWF que possuem, pelo menos, uma instância executada. Para a versão de processo, o mesmo conceito foi mantido durante a importação do modelo do Oracle, ou seja, foi usada a versão do processo principal.
Dimensão	*Resources_D*	Definição de usuários e grupos. São importados todos os usuários e papéis que foram definidos como inicializadores de instâncias de processos, ou responsáveis por alguma notificação. Também é criado um usuário padrão que representa o Workflow Engine utilizado na execução de atividades que representam funções automatizadas.
Dimensão	*Time_D*	Definição de datas. Armazena todas as datas utilizadas na base do Workflow em diferentes modos (i.e. dia, mês, ano, hora e mês fiscal). As datas importadas são todas as datas de início e fim de execução de instâncias de processos e as datas de início, limite e fim de execução de atividades.
Dimensão	*Service_Defs_D*	Definição das atividades do tipo função e notificação.
Dimensão	*Node_Defs_D*	Definição das mesmas atividades de Service_Defs_D, incluindo também os processos. No OWF, pode-se definir uma atividade em um Item Type (definição de modelos de processos) e utilizá-la em outro. Basicamente, a diferença entre as duas dimensões é o fato da dimensão dos serviços armazenar o grupo ao qual o serviço pertence. Como esta informação não está definida no modelo da Oracle, a informação importada é o nome de visualização da atividade, meramente documentacional.
Dimensão	*Arc_Defs*	Definição de arcos entre os nodos, ou transições entre atividades. Armazena as informações extraídas da tabela correspondente no modelo do ***Oracle***.
Dimensão	*Data_Defs_D*	Definição de atributos do processo. O OWF possui três tipos distintos de atributos: de processos, atividades e mensagens (notificações). Eles são armazenados em tabelas específicas. Como o modelo da ***Oracle*** permite existirem atributos com mesmo nome para processo e atividades, um tratamento especial foi necessário para que se possa fazer a distinção entre os atributos no modelo multidimensional: o nome do atributo é a concatenação do identificador da origem do dado (***Process***, ***Activity*** ou ***Notification***), do nome do processo, atividade ou notificação e do nome original do atributo.
Fato	*Proc_Inst_F*	Definição das instâncias dos processos. São importadas todas as definições das instâncias executadas e já finalizadas.
Fato	*Service_Inst_F*	Definição das instâncias de atividades. Registra informações de todas as atividades executadas, definidas em ***Service_Defs_D***.
Fato	*Proc_Inst_Data_F*	Armazena todos os valores dos atributos definidos para os modelos de processos e para as instâncias já finalizadas, na forma textual.
Fato	*Node_Inst_Data_F*	Armazena os valores dos atributos definidos para as atividades e notificações, que foram executadas nas instâncias importadas. Os valores são armazenados, também, na forma textual.

Cada desenvolvedor, no momento em que for programar a respectiva classe de interface para o novo algoritmo de mineração, deve se preocupar em como buscar os dados, no modelo analítico usado pela arquitetura, para passar para a sua implementação

do algoritmo. Para a busca de dados na base analítica, uma consulta SQL deve ser definida em um arquivo XML e um nome único no sistema deve ser atribuído a ela. O parâmetro passado pelo sistema para o método *executeAlgorithm* é uma instância de uma classe de serviço que possui a conexão com o banco de dados que se está utilizando (atualmente *Oracle*). Porém, com o uso de uma classe de serviços é possível realizar pequenas alterações para se utilizar qualquer SGBD. Com esta instância, o desenvolvedor pode executar a sua consulta na base de dados e preparar os dados retornados para a execução no seu algoritmo. A única restrição imposta, no momento, é que o algoritmo só pode ser executado sobre os *logs* de execução de um único modelo de processo e versão previamente selecionados pelo usuário (Item_Type e Version, na nomenclatura OWF).

Os dados resultantes da execução do algoritmo são salvos em tabelas. Isto é realizado pela mesma instância de serviço de banco de dados passada por parâmetro do método *executeAlgorithm*. Para cada classe de algoritmo de mineração, é definida uma tabela específica para armazenar os resultados da sua execução. A Figura 4 ilustra a tela de cadastro de um novo algoritmo no sistema, para posterior execução. Na mesma, pode-se perceber a definição da classe que implementa o algoritmo de mineração.

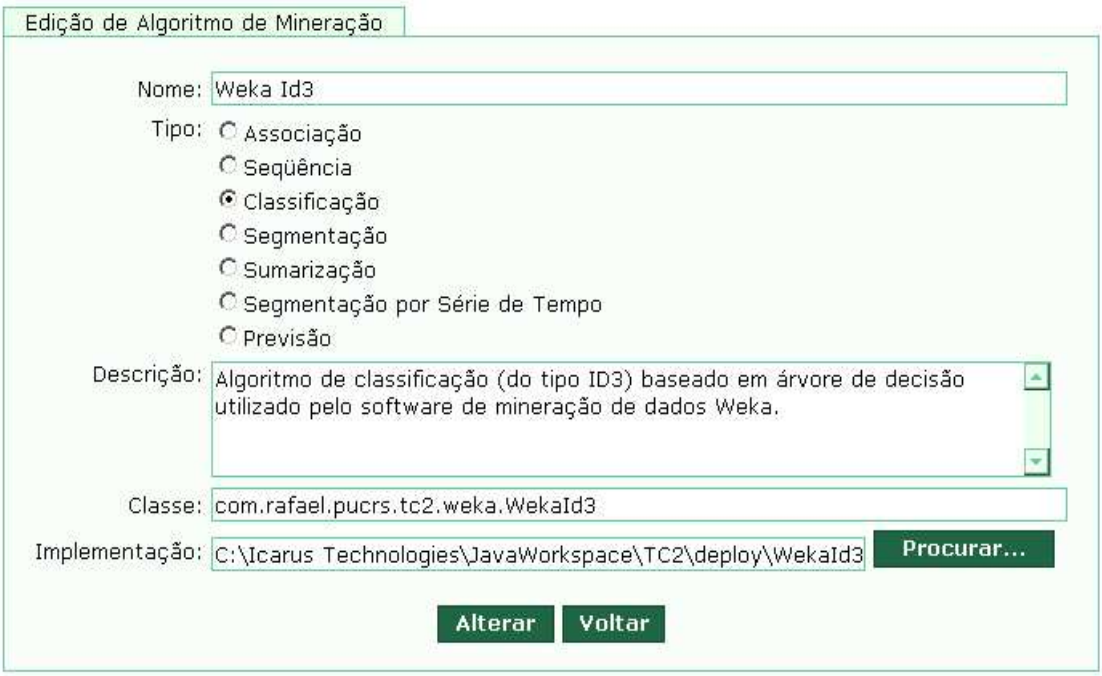

Figura 4 - Cadastro de um novo algoritmo no sistema

5. Visualização de resultados

Identificadas as regras de execução dos processos, os dados consolidados são apresentados ao usuário, utilizando alguma maneira gráfica, pelo MV. Dentre outras, gráficos, fichas texto e recursos de visualização 3D podem ser utilizados. Para os modos de visualização, foi utilizada a mesma idéia de incorporação de implementações, utilizada para os algoritmos de mineração de dados.

Desta forma, a inclusão de novos modos de visualização é feita através da criação de uma classe que implemente uma segunda interface proposta e que seja encapsulada juntamente com o arquivo de implementação (*jar*). Esta interface é

denominada *VisualizationInterface* e define um único método chamado *generateVisualization*. Este método também recebe a instância da classe de serviço com o banco de dados (utilizada para buscar o resultado da execução do algoritmo de mineração) e o descritor da página (instância de *JspWriter*). Com o descritor da página, qualquer tipo de visualização pode ser gerada, deste acompanhamento textual até visualizações gráficas baseadas em *Java Applets* ou componentes do tipo *ActiveX*. A Figura 5 mostra a tela de alteração de um modo de visutalização, com a referência à classe que a implementa, e a Figura 6 mostra a visualização no modo textual.

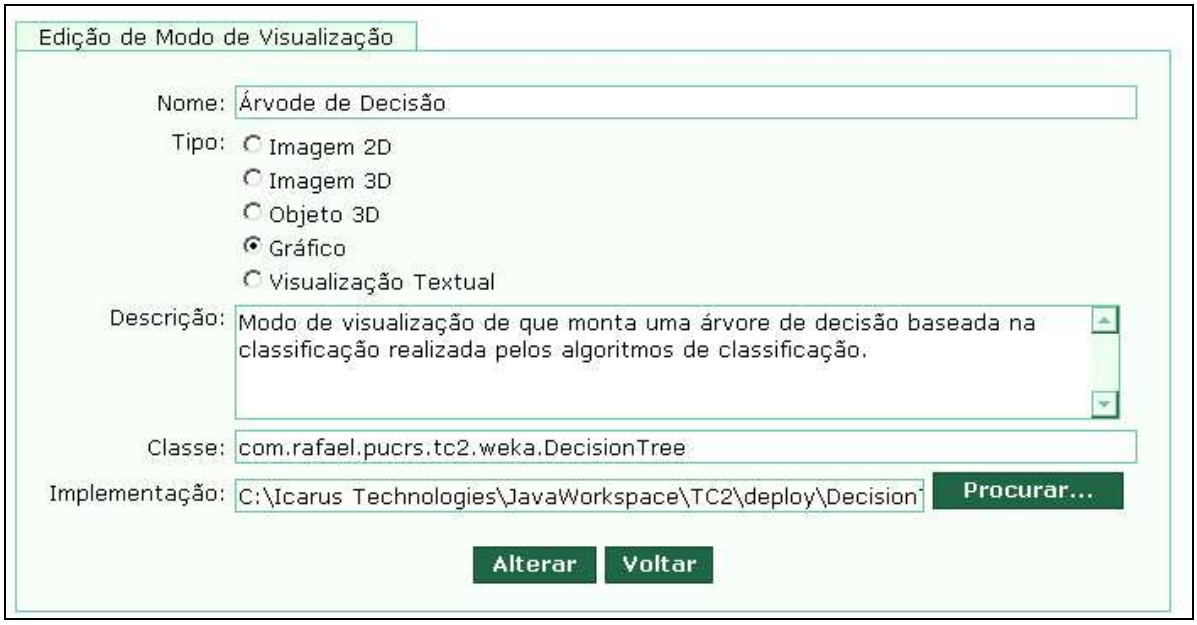

Figura 5 – Alteração de Modo de Visualização.

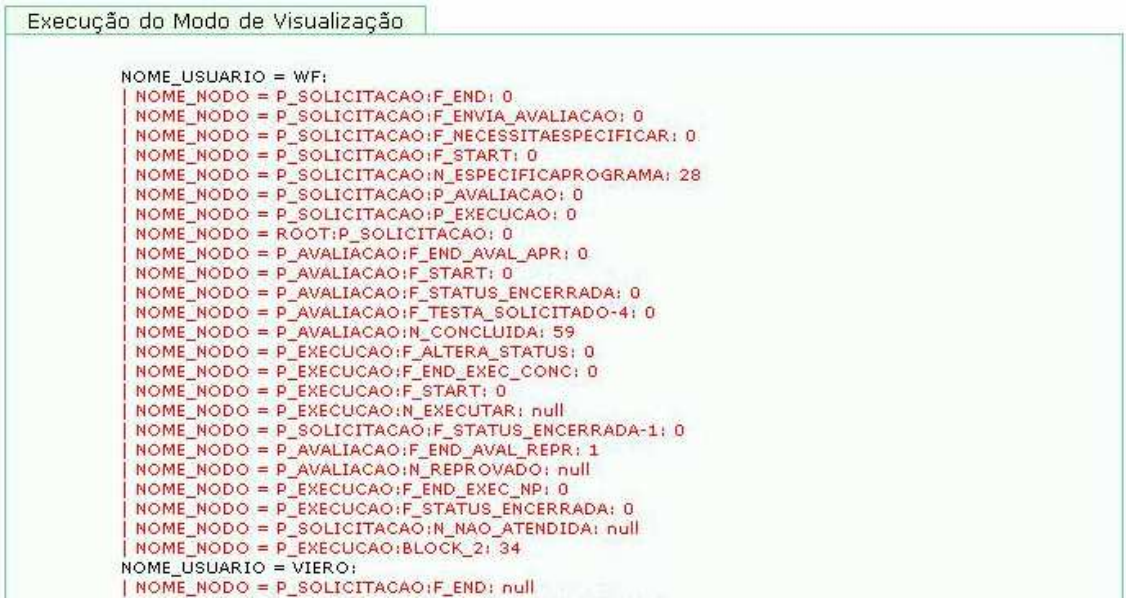

Figura 6 – Modo Texto de Visualização.

6. Considerações finais e próximos passos

Neste trabalho foi apresentada uma arquitetura de análise de *logs* de execução de processos de *Workflow*. A idéia desta arquitetura é a de permitir que os gestores das organizações, possam realizar uma análise sobre o comportamento de seus processos, baseada em técnicas de descoberta de conhecimento. Para tanto, foi desenvolvida uma ferramenta Web, plataforma Java-Tomcat, tendo Oracle Workflow como SGWf. Foram realizadas experimentações sobre 2 distintos processos de negócio reais, contento ao todo 7 versões de processos, e 2368 instâncias de processos. Os resultados obtidos utilizando-se esta ferramenta foram considerados satisfatórios, visto que a ferramenta atingiu seus objetivos em definir uma arquitetura adequada para a realização de processos de descoberta de conhecimento por completo, como apresentado em [Han01].

Como trabalhos futuros, pretende-se: otimizar os scripts de pré-processamento de dados para aumentar a qualidade dos dados importados e conseguir fazer uma melhor distinção entre as versões de sub-processos; criar classes de serviços e arquivos de propriedades para a incorporação automática de outros SGWfs para o processo, além da incorporação de *frameworks* de persistência de dados; estudar e desenvolver tabelas de armazenamento de resultados para outras classes de algoritmos, visto que atualmente apenas algoritmos de classificação são suportados.

Referências bibliográficas

Bonifati, A.; Casati, F.; Dayal, U.; Shan, M. (2001). "Warehousing Workflow Data: Challenges and Opportunities". In: *VLDB' 2001,* p. 649-652, Roma.

Chang, S.; Jaeckel, C. (2002). "Oracle Workflow Guide". Release 2.6.2, Volume 1, Oracle Corporation.

Eder, J.; Olivotto, G. E.; Gruber, W. (2002). "A Data Warehouse for Workflow Logs". *Data & Knowledge Engineering*, v.47 n.2, p.237-267 Austria.

Garcia, R. S.; Ruiz, D. D. (2005). "Pré-Processamento de Dados para Descoberta de Conhecimento em Processos de Workflow Modelados sobre Plataforma Oracle". In: *Escola Regional de Banco de Dados – SBC*. Porto Alegre.

Garcia, R. S. (2005a). "Descoberta de conhecimento em processos de Workflow modelados e executados sobre plataforma Oracle". FACIN-PUCRS (Trabalho de Conclusão de Curso).

Grigori, D.; Casati, F.; Castellanos, M.; Dayal, U.; Sayal, M.; Shan, M. (2004). "Business Process Intelligence"**.** *Computer In Industry.* V.53, p.321-343. Elsevier.

Han, J.; Kamber, M. (2001). "Data Mining: Concepts and Techniques". Morgan Kaufmann.

Hollingsworth, D. (1995). "Workflow Management Coalition: The Workflow Reference Model". http://www.wfmc.org/standards/docs/tc003v11.pdf

Leymann,F.; Roller,D. (2000). "Production Workflow: Concepts and Techniques". Prentice-Hall.