

# **GestInf - Um Repositório de Meta-Dados**

Mário J. Silva, João P. Campos, Nuno Maria e Pedro Gaspar

Faculdade de Ciências da Universidade de Lisboa

Departamento de Informática

## **Resumo**

O GestInf é um sistema para recolha, armazenamento e consulta de meta-informação, ou seja, um sistema de gestão de dados sobre outros sistemas de informação. Os dados que modela representam modelos de dados de outros sistemas. Constitui um exemplo de uma ferramenta de apoio à gestão do conhecimento. Este artigo descreve a arquitectura do repositório do GestInf, os programas de recolha desenvolvidos e a organização do editor de modelos que desenvolvemos para visualizar e modificar os modelos definidos no repositório.

**Palavras chave:** meta-dados, gestão do conhecimento, engenharia de modelos.

## **1 Introdução**

Nas organizações de média e grande dimensão, o grande número de colaboradores e a variedade de áreas de conhecimento necessárias à condução do negócio impõem a informatização dos sistemas de gestão do conhecimento. Ao centralizar a informação dispersa permite-se o acesso fácil e uniforme a essa informação. Após um tempo mais ou menos longo, é inevitável que, por muitas e diversas razões, a informação de negócio se apresente dispersa por vários sistemas de armazenamento. Estes possuem modelos de dados desenvolvidos independentemente segundo as conveniências de cada uma das aplicações utilizadas nas áreas de trabalho necessárias à condução do negócio. Na maioria dos casos, ao analisarmos o conjunto dos modelos como um só, detectamos uma enorme quantidade de dados duplicados, dependências funcionais e todas as anomalias que os analistas procuraram eliminar no desenho lógico individual de cada um dos modelos em separado.

O projecto GestInf visou a criação de um repositório para integração dos meta-dados com a informação de negócio dispersa por várias bases de dados dentro de uma organização. Permite

capturar e relacionar toda a informação associada ao ciclo de vida das aplicações, desde a análise até à sua exploração. O seu desenvolvimento decorreu no âmbito de um projecto conjunto entre a Faculdade de Ciências da Universidade de Lisboa e o Banco de Portugal.

O GestInf teve como objectivo principal encontrar formas de obter reduções do tempo e esforço de desenvolvimento e manutenção das aplicações de um sistema de informação. O GestInf é um exemplo de uma ferramenta de gestão de conhecimento [Meta Data Coalition OIM KMM 1999]. Realiza as seguintes funções:

- Identificação de componentes de software já desenvolvidos com vista a estudar a possibilidade de os reutilizar em novas aplicações.
- Avaliação do impacto de alterações efectuadas aos modelos de dados nas aplicações em produção.

O GestInf constitui portanto um sistema de informação para recolha e organização da informação existente sobre os sistemas de informação de uma organização. Os componentes de software principais deste sistema são:

- repositório de meta-dados, ou seja uma base de dados com a informação relativa à estrutura e atributos de modelos e estruturas de dados de ferramentas de software, aplicações e sistemas de informação;
- programas de extracção dos meta-dados de outros sistemas de informação ou das ferramentas CASE usados no seu desenvolvimento e seu carregamento no repositório;
- ferramentas de navegação e modificação interactiva dos dados residentes no repositório;
- ferramentas de geração de relatórios sobre os dados carregados no repositório e consulta dos registos dos processos de carregamento realizados.

O GestInf adopta a forma mais moderna de conceptualização da meta-informação, seguindo um modelo de referência proposto pelo Object Management Group (OMG) [Object Management Group 2002]. Este organiza a informação em quatro camadas:

**dados:** a informação propriamente dita, manipulada pelos sistemas modelados;

**modelos:** a informação que representa as estruturas de dados dos sistemas de dados;

**meta-modelos:** as representações dos modelos;

**meta-meta-modelo:** o modelo de representação dos meta-modelos.

Neste artigo apresentamos o meta-meta-modelo e meta-modelos que definimos para representar os meta-dados geridos pelo GestInf, as ferramentas de desenvolvidas para o seu carregamento e,

finalmente, a interface concebida para navegação e interrogação sobre esses dados. Com o sistema construído é possível carregar:

- modelos de vários sistemas de bases de dados relacionais, a partir os seus catálogos;
- estruturas de dados de aplicações definidas em ferramentas CASE;
- meta-dados descritos pelos utilizadores; definimos um formato que permite a sua introdução numa folha de cálculo, para posterior carregamento no repositório;
- quaisquer outros meta-dados, por introdução manual directamente sobre o repositório com recurso a um editor.

O resto desta comunicação está organizado da seguinte forma: na Secção 2 descrevemos os requisitos que presidiram ao desenvolvimento do repositório GestInf, os seus objectivos e o modelo lógico adoptado; na Secção 3, as aplicações construídas; na Secção 4, a implementação do GestInf; na Secção 5, comparamos o GestInf com outros repositórios e modelos de meta-informação propostos; finalmente, na secção 6, sumarizamos as conclusões sobre os resultados alcançados e tecemos considerações sobre direcções possíveis de evolução do sistema GestInf.

## **2 GestInf**

A gestão de conhecimento depende da existência de processos sistemáticos para recolha, organização e utilização da informação das organizações, de forma a dotá-las de uma visão integrada do seu negócio que as leve a adquirir vantagens competitivas.

Considere-se, por exemplo, uma organização que utiliza para diversos fins bases de dados suportadas por SGBD's de vários fabricantes; a organização desejaria ver as bases de dados de forma integrada, como se de uma só se tratasse, para assim saber de que dados dispõe, e, possivelmente, relacionar os dados disponíveis nas BD dos diferentes fabricantes.

Para permitir a obtenção de uma visão integrada de todos os dados e dependências, o GestInf carrega todos os modelos de informação num repositório. A informação sobre estes modelos é guardada num meta-modelo, ou seja, uma estrutura de dados capaz de representar, por exemplo, qualquer modelo de dados de qualquer base de dados. Na organização para a qual o GestInf foi concebido foram inicialmente criados dois meta-modelos:

**modelo físico de dados:** permite armazenar e relacionar informação dos modelos das bases de dados;

**modelo lógico de dados:** permite armazenar e relacionar informação originária das ferramentas CASE utilizadas.

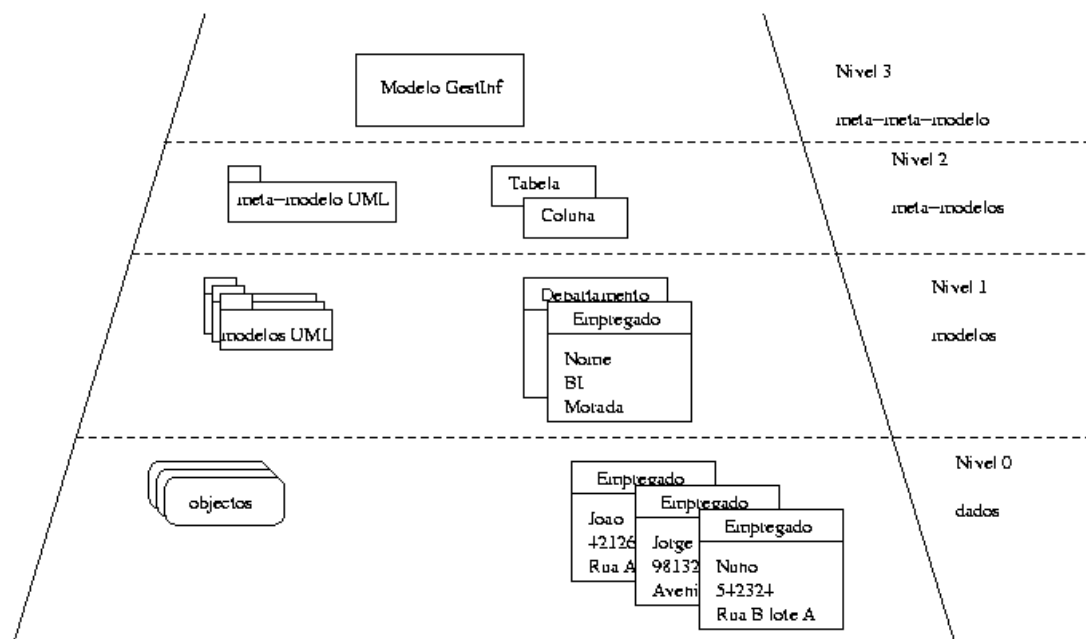


Figura 1: Modelo de meta-informação em quatro camadas da Meta Object Facility (OMG)

Ao operar sobre meta-modelos, o GestInf permite representar os modelos de dados de forma comum e relacionar informação entre dados de áreas de negócio distintas. Contudo, apenas com meta-modelos, o GestInf, apesar de permitir relacionar objectos provenientes de duas fontes de dados diferentes, não permitiria representar, por exemplo, a materialização de determinada entidade lógica numa tabela. Surge assim a necessidade de criar um novo nível de abstracção, o de um meta-meta-modelo, ou seja, uma estrutura de dados que permita representar os meta-modelos criados, relacionando objectos de meta-modelos diferentes. Este modo de organização da informação em modelos, meta-modelos e meta-meta-modelos é central na arquitectura de informação do GestInf, pois determina na sua essência a forma como toda a informação é estruturada.

### Modelo de Meta-Dados

No repositório do GestInf os meta-dados são geridos em quatro níveis (ver Figura 1), segundo um modelo de organização proposto pelo OMG na sua MOF — Meta Object Facility [Object Management Group MOF, 1999]:

- No nível inferior (nível 0, ou nível de dados) encontram-se os dados, que não são objecto de processamento pelo GestInf;
- No nível seguinte (nível 1, ou nível de modelo) consideram-se os modelos de dados;

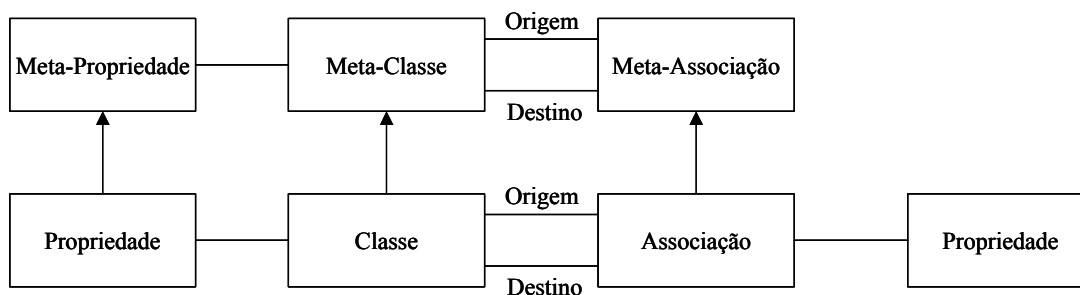


Figura 2 Meta-meta-modelo resumido do GestInf

- No nível acima (nível 2 ou nível de meta-modelo) consideram-se os meta-modelos de dados, sobre os quais se definem os modelos do nível 1;
- No último nível (nível 3 ou meta-meta-modelo) encontra-se o modelo lógico, que define a forma como os dados são armazenados. Este modelo suporta todo o repositório e sobre ele é possível definir modelos dos níveis 2 e 1.

O meta-meta-modelo representa a estrutura de dados em que assenta o GestInf e será doravante designado simplesmente por *modelo do GestInf*.

### Modelo do GestInf

O modelo do GestInf, representado sumariamente na Figura 2, tem quatro entidades centrais: classe (ou categoria), meta-classe, meta-associação e associação. Para ilustrar a utilização do modelo, consideremos, como exemplo, a modelação pelo GestInf de um modelo relacional com uma única tabela, *Empregado*:

- **meta-classe** representa uma definição de um tipo de classes do GestInf, pelo que representaria no exemplo os conceitos de *Tabela* e *Coluna*;
- **classe** representa uma instância de uma determinada meta-classe, pelo que representaria no exemplo a tabela *Empregado* e a coluna *Número de Segurança Social* nessa tabela;
- **meta-propriedade** representa uma característica de uma meta-classe, como por exemplo o *Nº de ocorrências na Tabela*. As classes podem instanciar as meta-propriedades da meta-classe a que pertencem, por exemplo *o Nº de ocorrências na Tabela Empregado é 2750*;
- **meta-associação** representa um tipo de associação que se define entre duas meta-classes; por exemplo, *Tabela tem Coluna*;

- **associação** representa uma instância de uma meta-associação aplicada a duas classes, por exemplo *Empregado tem Número de Segurança Social*;
- **propriedade** representa informação estruturada (por exemplo: datas, números, expressões curtas) ou referências para informação não estruturada (por exemplo: URLs de páginas HTML, imagens, sons). As associações, tal como as classes, podem ter propriedades. No entanto, estas são independentes das meta-associações.

### 3 Aplicações do GestInf

Para carregar os modelos de dados no GestInf foram desenvolvidos quatro programas:

**carregador Sybase:** projecta o catálogo de uma base de dados Sybase no modelo físico de dados;

**carregador DB2:** projecta o catálogo de uma ou mais bases de dados DB2 (IBM) no modelo físico de dados;

**carregador Key:** projecta os modelos definidos nas ferramentas Key Enterprise [Sterling Software 1996] no modelo lógico de dados;

**carregador por lotes:** permite carregar para o repositório informação sobre modelos de informação em folhas de cálculo Microsoft Excel; os dados nas folhas de cálculo têm de estar organizados num formato específico definido para este carregador.

Os carregadores Sybase, DB2 e Key são dependentes dos meta-modelos dos sistemas a partir dos quais importam os dados. Se estes forem alterados, os carregadores terão também de ser alterados.

Para consultar a informação carregada no GestInf foram construídas três ferramentas com funções distintas:

**Extracção de relatórios:** permite formatar relatórios com conteúdo seleccionado do repositório do GestInf;

**Exploração interactiva:** permite navegar interactivamente no conteúdo do GestInf, atravessando as relações entre os objectos. Esta aplicação apenas permite visualizar os dados carregados, não sendo possível a alteração dos modelos;

**Editor de modelos:** também designado por carregador discreto de informação, ou carregador manual, permite navegar no conteúdo do GestInf, criar novos objectos e

novos relacionamento entre estes, assim como apagar objectos existentes ou seus relacionamentos;

As ferramentas de exploração interactiva e o carregador discreto são completamente independentes dos meta-modelos definidos no repositório de informação, pelo que podem continuar a ser utilizadas, mesmo se os meta-modelos forem alterados ou forem criados novos meta-modelos para outros domínios de informação.

No resto desta secção discutimos em maior detalhe o editor de modelos desenvolvido para o GestInf, por a sua descrição permitir ilustrar a forma característica de manipular a informação presente em sistemas de gestão de dados baseados em meta-modelos de informação, como é o caso do presente. Finalizamos a secção com uma discussão do mecanismo de controlo de acessos desenhado para o repositório.

### **Editor de Modelos**

A aplicação de carregamento manual de dados (também chamada de carregamento discreto, ou de edição) permite ver e editar os modelos de dados (nível 1) disponíveis no GestInf. O editor foi desenvolvido em três fases:

- na primeira fase, fez-se a análise dos domínios da informação dispersa pela organização, em conjunto com a análise do GestInf. Desta análise resultou o modelo do GestInf e os meta-modelos aqui apresentados;
- numa segunda fase, realizou-se a análise navegacional, tendo-se projectado os vários elementos disponíveis no GestInf em classes modeladas na linguagem UML [Fowler e Kendall 1997];
- finalmente, procedeu-se ao desenho da interface utilizador, onde os objectos abstractos definidos na análise navegacional foram mapeados em objectos gráficos interactivos.

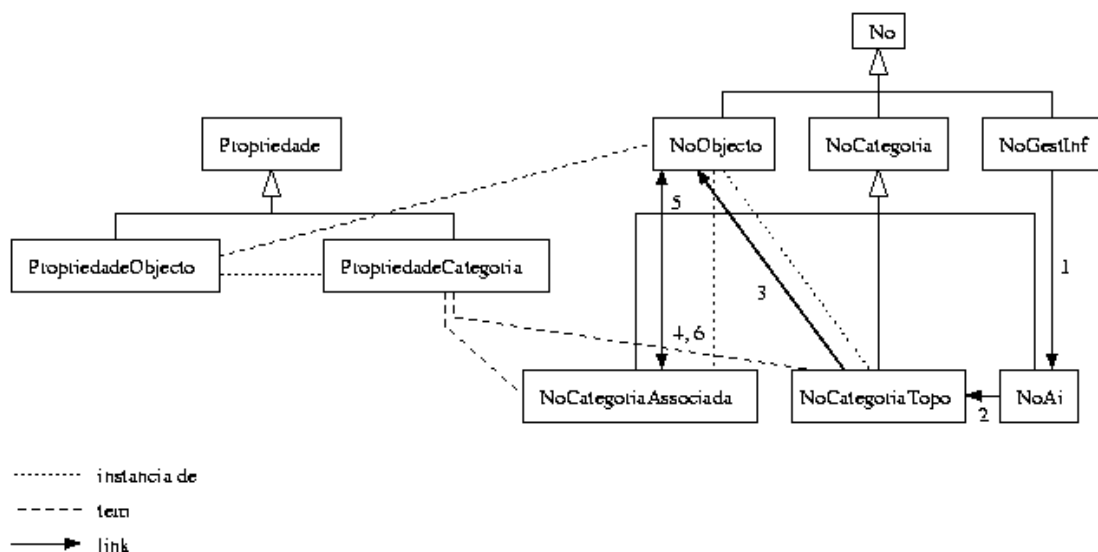


Figura 3 Modelo da aplicação de edição de modelos

O modelo de objectos do editor encontra-se representado na Figura 3. No grafo que representa o modelo, as categorias e objectos do GestInf são representados como nós. Para permitir o acesso à informação, foi criado um nó especial, o *NoGestInf*, que não representa quaisquer dados.

Existem vários tipos de categorias representadas:

- As áreas de informação representadas como *NoAi* representam áreas de negócio da organização; não existem objectos que instanciem áreas de informação; não têm propriedades;
- As categorias de topo são meta-classes de objectos. Podem existir sem que as suas instâncias estejam obrigatoriamente associadas a outros objectos;
- As categorias associadas representam categorias de objectos que não podem existir sem estarem associados a outros objectos.

Os *NoObjecto* representam classes do GestInf. São instâncias de uma categoria associada ou de uma categoria de topo.

As categorias e os objectos têm propriedades; as propriedades de objecto são sempre instâncias de propriedades de categoria. Como os modelos e meta-modelos armazenados no GestInf podem ser bastante complexos, existe um número muito elevado de formas possíveis de exploração destes dados. Para podermos construir um modelo de navegação houve, por essa razão, que seleccionar um número limitado de caminhos possíveis de exploração dos dados (representados na Figura 3 por setas mais espessas).

1. Entrando por *NoGestInf*, o utilizador obtém a lista das áreas de informação;



2. Seleccionando uma área de informação, obtém as categorias de topo dessa área de informação;
3. Seleccionando uma das categorias de topo, obtém todos os objectos dessa categoria;
4. Seleccionando um dos objectos, obtém as categorias de objectos passíveis de serem associados com este;
5. Seleccionando uma das categorias associadas, obtém a lista de objectos desta categoria associados com o objecto seleccionado em 4;
6. Seleccionando um dos objectos volta para 4, com esse objecto pré-seleccionado.

A metáfora de navegação adoptada corresponde a considerar o grafo dos dados do GestInf como uma rede projectada num espaço bi-dimensional (2D); pegando nesta rede e *puxando* pelas áreas de informação, cria-se uma nova terceira dimensão, transformando o grafo que é a base do nível 1 numa árvore. Há, contudo, uma excepção que quebra esta metáfora: os ciclos contidos nos meta-modelos.

É de notar ainda que as classes obtidas não espelham directamente as relações entre os vários níveis do GestInf (por exemplo, *NoObjecto* não é filho de *NoCategoria*).

Esta aplicação é usada como editor de modelos de dados (nível 1); no entanto, todos os modelos de dados armazenados no GestInf estão aplicados a algum meta-modelo. No editor, usamos os meta-modelos para formatar a interface onde aparecerão os modelos de dados, sendo os dois níveis tratados de forma diferente. Por esta razão, há portanto discrepâncias entre o modelo adoptado para esta ferramenta e o do GestInf.

A aplicação foi desenvolvida em Java [Sun 2002]. Adopta um *Look&Feel* em que o grafo do GestInf aparece representado como uma árvore, com recurso aos componentes gráficos do package *Java Swing*, nomeadamente o componente *JTree*. O mapeamento do grafo na árvore processa-se da seguinte forma:

- Os nós são nós da árvore;
- O *NoGestInf* é a raiz da árvore;
- A listagem dos links possíveis de serem seguidos corresponde à expansão de um nível de um ramo da árvore; a expansão está limitada a um só nível de cada vez (e não mais, porque os ciclos dos meta-modelos causariam ciclos infinitos na aplicação);
- Seguir um link corresponde a seleccionar um nó.

Quando na ferramenta de edição o utilizador segue um link ancorado num nó do grafo rotulado com o nome de um objecto, a interface de navegação apresenta-lhe automaticamente um painel com as propriedades do objecto. Para editar os objectos, há que recorrer ao botão direito da interface de utilizador. Ao pressioná-lo, pode-se de seguida optar por uma das opções de criar / alterar / apagar o objecto. De forma similar, ao seleccionar com o mesmo botão um ramo do grafo, pode editar os objectos do tipo associação do modelo.

Os resultados das acções sobre os objectos variam em função da forma como o utilizador chegou a eles; isto é, os objectos criados a partir do editor *herdam* automaticamente as propriedades daqueles que os antecedem no caminho percorrido ao navegar até eles. Por exemplo, ao criarmos um objecto debaixo de uma categoria de topo criamos um objecto dessa categoria; se criarmos um objecto debaixo de uma categoria associada (mesmo que a categoria representada seja a mesma), criamos um objecto dessa categoria e uma associação com o objecto associado à categoria.

### **Controlo de Acesso**

No GestInf o controlo de acessos ao repositório é definido ao nível da classe. Os modos de acesso suportados são: leitura, escrita, remoção e criação. Os utilizadores vêm a informação consoante as permissões que têm sobre os objectos, e cada objecto reage diferenciadamente, consoante as permissões que o utilizador detém sobre a categoria do objecto.

Estando os dados dos modelos armazenados numa base de dados relacional, torna-se impossível estabelecer mecanismos de controlo de acesso a quem, tendo recebido privilégios de consulta a uma parte dos dados do repositório, não deva ter possibilidades de leitura da sua totalidade. Não existem, no modelo relacional, formas de restrição da leitura de um modelo (ou partes destes) através regras de integridade especificadas sob forma declarativa ou triggers.

O controlo de acesso é assim feito pelas aplicações cliente do GestInf. Por exemplo, na aplicação de edição dos modelos implementámos um mecanismo de lista de controlo de acessos, mediante o qual a navegação pelas classes dos modelos é restrita aos utilizadores autorizados na lista de controlo de acessos.

## **4 Implementação**

O GestInf foi desenvolvido por fases e por módulos, de acordo com uma estratégia de desenvolvimento por protótipos sucessivos [Pressman 2001]. O nosso trabalho de desenvolvimento esteve ligado aos três protótipos iniciais. Em cada um dos protótipos o modelo do repositório e os meta-modelos evoluíram e sofreram modificações. A cada protótipo foram

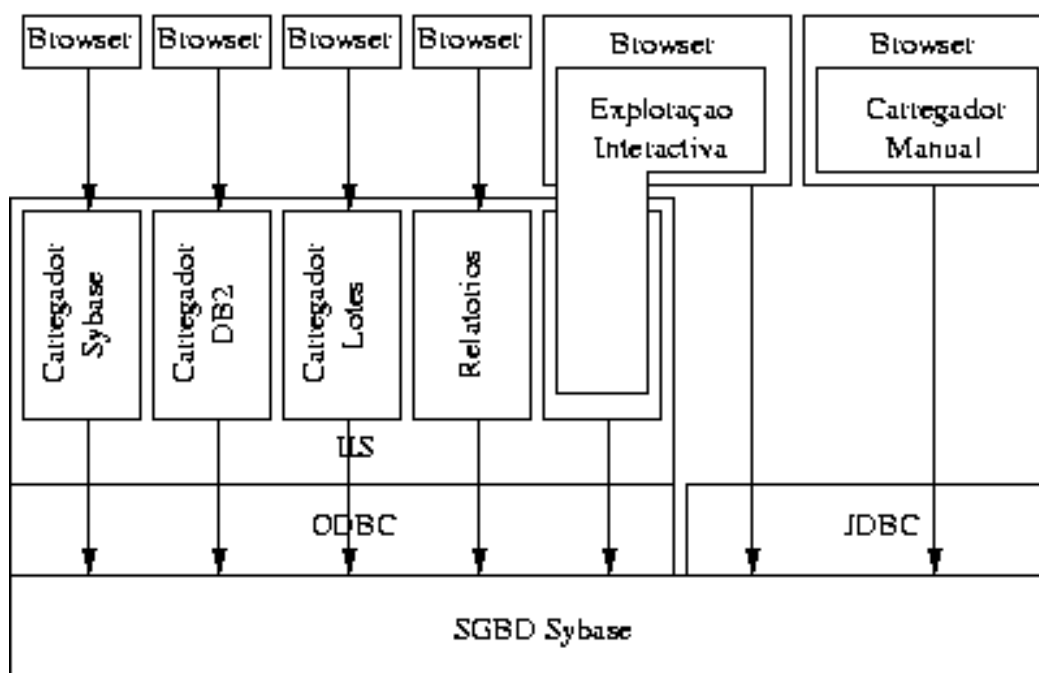


Figura 4: Arquitectura do Software da Implementação do GestInf

acrescentados novos módulos que permitiram aumentar as funcionalidades disponíveis sobre o sistema.

A abordagem de desenvolvimento por protótipos foi escolhida porque, apesar de haver um conjunto geral de objectivos a atingir, não era possível definir antecipadamente as interfaces dos vários módulos, o processamento a dar aos dados ou a forma de apresentação dos resultados a adoptar. Ao longo do processo de desenvolvimento foram construídas várias ferramentas que permitem carregar, manipular e extrair informação do GestInf.

O GestInf foi construído sobre uma arquitectura cliente-servidor, representada na Figura 4. Os dados estão armazenados de duas formas diferentes: a informação estruturada está armazenada num sistema de gestão de base de dados relacional Sybase; a informação não estruturada está armazenada em ficheiros HTML.

Todos os utilizadores acedem ao sistema através de *browsers Web*. Para a utilização das ferramentas de exploração interactiva e de carregamento discreto de dados é necessário que o browser disponha de uma máquina virtual Java versão 1.2 (actualmente conseguido através da instalação do Java Plug-in).

As ferramentas de geração de relatórios e carregamento automático e por lotes foram construídas em *VB Script Active Server Pages (ASP's)* e correm no servidor sobre *Microsoft*

*Internet Information Server*. As ASP acedem à base de dados que suporta o GestInf através do protocolo *ODBC*.

A ferramenta de carregamento discreto de informação foi construída como applet Java 1.2 e corre numa máquina virtual Java no browser dos utilizadores. Estes acedem à base de dados que suporta o GestInf através de *JDBC*.

A ferramenta de exploração interactiva (editor de modelos) é uma applet Java 1.2 que invoca também algumas páginas HTML geradas com ASP.

O esforço total dedicado ao projecto pela equipa de desenvolvimento foi de aproximadamente 30 pessoas\*mês, distribuídos ao longo de 15 meses durante o ano de 2000 e início de 2001. A equipa contou com 7 elementos, todos envolvidos parcialmente no projecto.

O esforço de desenvolvimento do editor de meta-modelos foi de aproximadamente 10 pessoas\*mês. A maior parte do tempo foi dispendido nos testes e depuração do código, justificável pela complexidade das estruturas recursivas que manipula. Por outro lado, o tempo dispendido no desenvolvimento da totalidade dos carregadores e geração de relatórios não terá excedido 8 pessoas\*mês.

## **5 Outros Repositórios de Meta-modelos**

Existem várias propostas de standardização de modelos de meta-dados. O Open Information Model (OIM) é uma norma de um modelo para a representação de meta-dados [Meta Data Coalition OIM 1999]. Possui meta-modelos para representar, por exemplo, esquemas de bases de dados relacionais, armazéns de dados, e informação proveniente de ferramentas de desenvolvimento de software. O OIM foi desenvolvido pela Meta Data Coalition (MDC) [Meta Data Coalition 2001].

Um dos produtos de software que implementa o OIM é o Microsoft Repository (MR) [Bernstein et al. 1997]. Trata-se de um repositório orientado a objectos com suporte para versões, configurações e espaços de trabalho (*workspaces*). É acedido pelas aplicações que o usam através de um conjunto de objectos COM e armazena os dados numa base de dados relacional *SQLServer*. O MR é suportado tecnicamente pela Microsoft, embora seja um componente interno de alguns dos seus produtos de software.

O Common Warehouse Meta-model (CWM) é um modelo proposto para permitir a troca de modelos de negócio e modelos de armazéns de dados entre ferramentas, repositórios e plataformas disponíveis em ambientes heterogéneos [Object Management Group CWM 2001]. Foi construído sobre três outras normas:

- Unified Modeling Language (UML), uma linguagem para especificar, visualizar, construir e documentar sistemas de software [Booch et al. 1999];
- Meta Object Facility(MOF), uma norma para repositórios de meta-dados e meta-modelos [Object Management Group MOF 1999];
- XML Meta-data Interchange (XMI), uma aplicação de XML para o intercâmbio de meta-dados [Object Management Group XMI 2000].

A MDC terminou entretanto a sua actividade e uniu-se ao Object Management Group (OMG), doando o OIM como um contributo para o desenvolvimento do CWM. O OIM foi abandonado e não será desenvolvido ou suportado. O CWM é presentemente a única iniciativa de normalização da modelação de meta-dados suportada pela indústria de software.

## 6 Conclusão e Trabalho Futuro

As ferramentas de software para gestão do conhecimento das organizações (e o próprio conhecimento sobre a melhor forma de as concretizar) têm vindo a amadurecer, sendo prova disso a cada vez maior clarificação dos standards sobre os quais se desenvolverá o mercado destas ferramentas. Atendendo à evolução recente, o desenvolvimento da próxima versão do GestInf deverá necessariamente implementar o modelo de dados do CWM. Com a consolidação observada no domínio das ferramentas de gestão do conhecimento, não basta hoje apenas oferecer uma arquitectura estruturada nos mesmos princípios. Ao adoptar um modelo de dados aberto e partilhado com outras ferramentas poderá ser acedido por todas as aplicações de gestão de conhecimento que a indústria não irá deixar de desenvolver sobre este modelo.

O GestInf foi construído de forma iterativa até atingir a forma de protótipo operacional. Tem ferramentas para o carregamento e interrogação que permitem satisfazer a grande maioria das necessidades identificadas pelos utilizadores do sistema para o qual foi desenvolvido. No entanto, existem bastantes aspectos que podem ser melhorados com vista a aumentar as funcionalidades disponíveis. De entre estes salientamos:

**Sistema de controlo de versões:** à medida que o número de dados aumenta, tornar-se-á imperativo aumentar a quantidade e duração dos processos de recolha. Isto gerará automaticamente erros irrecuperáveis e contenção no acesso ao repositório pelos programas de edição e carregamento automático. Será, portanto, essencial contemplar as versões dos objectos como entidades fundamentais do modelo numa próxima versão.

**Suporte para gestão de informação pouco estruturada:** As aplicações desenvolvidas para o GestInf apenas integram modelos de informação estruturada. Num sistema deste tipo pode ser necessário gerir também meta-dados não estruturados ou apenas semi-estruturados (XML).

Sendo a estrutura de dados do repositório muito complexa, com vários níveis de abstracção interligados e muitas dimensões, muito ficou por fazer no que respeita à usabilidade das aplicações construídas para interacção dos utilizadores com o repositório, um trabalho tão necessário como de difícil abordagem. Podemos salientar, de entre os que trariam benefícios mais imediatos:

**Pesquisa por palavras chave:** No GestInf, deveria haver uma ferramenta para localização de objectos por palavras-chave, segundo um modelo de recuperação de informação [Baeza-Yates e Ribeiro-Neto 1999].

**Edição interactiva de meta-modelos:** Está prevista uma ferramenta que permita a edição interactiva de meta-modelos.

Os repositórios de meta-informação são um instrumento cada vez mais essencial nas grandes organizações com sistemas de informação onde reinam a heterogeneidade, a replicação e a redundância de informação. Embora o GestInf tenha sido desenvolvido num contexto de aplicação restrito ao suporte ao desenvolvimento e operação de aplicações informáticas em ambientes multi-base de dados, o seu âmbito de interesse é muito mais geral. A gestão dos meta-dados é apontada como um dos principais factores críticos de sucesso dos projectos de data-warehousing [Kimball 1998]. Sem contexto, a informação num data-warehouse deixa de ter valor. Um repositório como o GestInf, integra numa estrutura de dados extensível e uniforme toda a meta-informação, complementada por um conjunto de ferramentas de visualização e validação da mesma. Ao mesmo tempo, constitui-se como uma fonte centralizada de meta-dados que oferece a outras ferramentas de gestão do conhecimento a localização exacta da informação nas fontes. Possibilita assim a obtenção imediata de informação sem o atraso inerente à organização dos processos de transformação em estágios sucessivos, como preconizado nas abordagens clássicas de data-warehousing.

## **7 Agradecimentos**

A concepção do GestInf esteve a cargo de uma equipa que, além dos autores, integrou, pelo Banco de Portugal, Fernando Veiga da Costa, Cristina Guedes e Conceição Portas.

## 8 Referências

- Baeza-Yates, R. e Ribeiro-Neto, B., *Modern Information Retrieval*. ACM Press Series, Addison Wesley, Nova Iorque, 1999.
- Booch, G., Rumbaugh, J., Jacobson, I., *The Unified Modeling Language User Guide*, Addison-Wesley, 1999.
- Bernstein, P., Harry, B., Sanders, P., Shutt, D., Zander, J, “The Microsoft Repository”, *VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases*, Morgan Kaufmann, 1997.
- Fowler, M., Kendall, S., *UML Distilled: Applying the Standard Object Modeling Language*, Addison-Wesley, 1997.
- Kimball, R., *The Data Warehouse Toolkit*, John Wiley & Sons, Inc., 1996.
- Meta Data Coalition, *Open Information Model Knowledge Management Model, Knowledge Descriptions*, Review Draft, 15 de Julho de 1999.
- Meta Data Coalition, *Open Information Model*, proposta da Microsoft Croperation, Version 1.10 (Proposal), Agosto de 1999.
- Meta Data Coalition, *Driving the evolution of metadata interchange format standards*, Janeiro de 2001.
- Object Management Group. <http://www.omg.org>, 2002
- Object Management Group, *Meta Object Facility (MOF) Specification*, proposta conjunta de vários fabricantes, Version 1.3 RTF, 27 de Setembro de 1999.
- Object Management Group, *OMG XML Metadata Interchange (XMI) Specification*, Version 1.1, 2000.
- Object Management Group. *Common Warehouse Metamodel (CWM) Specification*, Version 1.0, 2001.
- Pressman, Roger S., *Software Engineering: A Practitioner's Approach*, Fifth Edition, McGraw-Hill, 2001.
- Sterling Software, *Key:Enterprise Development: Environment for Building Large-scale Client/Server Applications*, 1996.
- Sun Microsystems, *The Source for Java Technology*, <http://java.sun.com>, 2002.