O repositório XML do SIGDiC

João Paulo Rodrigues[1], José Correia[1]

INESC Porto, R. Dr. Roberto Frias, 4200-465 Porto
http://www.inescporto.pt/
{jpr,jcorreia}@inescporto.pt

Resumo O projecto SIGDiC pretende implementar um sistema integrado de gestão e difusão da conteúdos textuais (por exemplo, notícias ou SMS), constituindo o repositório o elemento central deste sistema.
A solução escolhida para o armazenamento dos conteúdos procura conciliar as vantagens das bases de dados relacionais, com a estruturação associada ao XML.
Sobre cada conteúdo foi possível definir informação adicional de metadata, utilizando o conjunto de elementos básicos da Dublin Core Metadata initiative.
O ambiente de edição é gerado, dinamicamente, a partir da informação definida no XML Schema que descreve o conteúdo. A partir deste ficheiro é possível inferir a maioria das regras de negócio e gerar, dinamicamente, um ambiente de trabalho Web que, através de formulários, permite a edição da informação.

1 Introdução

O trabalho aqui apresentado foi realizado no âmbito de um projecto financiado pelo PRIME - Sistema Integrado de Gestão e Difusão de Conteúdos (SIGDiC)[8] - cujo objectivo é desenvolver um sistema capaz de integrar diferentes fontes de conteúdos e disponibilizá-los, automaticamente ou semi-automaticamente, em diferentes canais de comunicação (SMS, Teletexto, WAP, Web, etc.).

Tendo em vista a definição de requisitos, o projecto tem como parceiro uma entidade líder em Portugal na área da produção, edição, gestão e difusão de conteúdos - a RTP. Contudo, o SIGDiC foi concebido tendo em mente as necessidades da generalidade das organizações que desenvolvem o seu "core-business"nesta área e, deste modo, são potenciais utilizadores dos resultados do projecto, todas as empresas nacionais e internacionais que produzem, editam e geram conteúdos para os sectores do audiovisual, Internet e telecomunicações.

O SIGDiC irá ter uma arquitectura modular, baseada em sistemas abertos, e está a ser desenvolvido recorrendo, tanto quanto possível, a ferramentas de domínio público. É neste contexto que surge a opção pelo XML [11], como forma de anotar e estruturar conteúdos, armazenados no repositório.

Em termos da arquitectura SIGDiC, o repositório assume uma função centralizadora da informação a disseminar pelos vários módulos. [1]

[1] Por altura da elaboração deste artigo, está desenvolvida apenas a primeira versão.

O objectivo da versão actual do repositório é suportar os serviços mínimos, que permitam o desenvolvimento, em paralelo, da restante arquitectura, bem como, a validação e teste das funcionalidades já implementadas.

2 Noção de Conteúdo

Antes de avançarmos mais, é oportuno definir a noção de conteúdo, tal como é utilizada no âmbito do projecto SIGDiC.

O que é um "conteúdo"para o SIGDiC?

É óbvio que uma notícia de imprensa é um conteúdo, assim como uma imagem, um vídeo, ou um ficheiro de audio poderão ser um conteúdo. Porém, mais importante é perceber a relação que têm entre si, isto é, saber que conteúdos existem relacionados com um determinado acontecimento ou facto.

Esta é a questão de fundo que se pretende tratar com um sistema como o SIGDiC. A noção de conteúdo abrange todos os diferentes conteúdos que se referem a um mesmo evento, acontecimento ou facto que é identificado como noticia ou informação.

O conteúdo relacionado com um acontecimento poderá incluir uma notícia extensa (para ser disponibilizada via Web), uma mensagem SMS (que será enviada a todos os utilizadores que subscreveram o serviço de notícias SMS), o teleponto que será lido pelo pivot durante o telejornal, a referência para um conjunto de ficheiros com imagens ou a referência para um ficheiro de vídeo com a reportagem. Assim, um conteúdo poderá ser constituído por vários tipos de conteúdos que documentam um dado evento, ou facto.

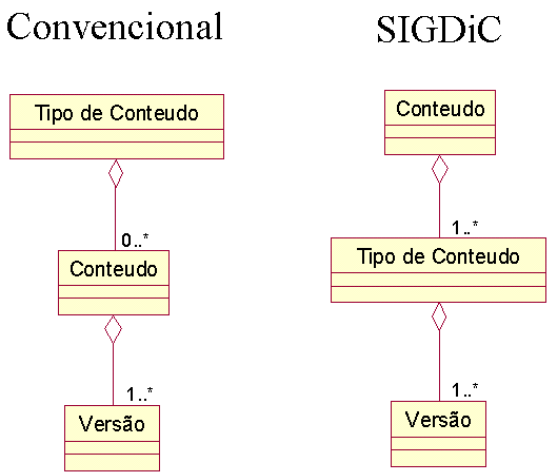

Figura 1. Noção de Conteúdo

No âmbito do projecto SIGDiC, a notícia, a versão WAP, o SMS e a informação do teleponto, são todos tipos de conteúdos que, na sua forma agregada e compilada, constituem o conteúdo SIGDiC.

Os sistemas de gestão de conteúdos convencionais apostam na separação dos vários tipos de conteúdos, de acordo com o meio de difusão.

A noção de conteúdo adoptada no SIGDiC tem por objectivo promover a reutilização de conteúdos, pelo que a informação relativa a uma noticia - o conteúdo - reúne todas as diferentes versões, destinadas a ser difundidas nos vários meios - os tipos de conteúdos.

O objectivo do projecto SIGDiC visa tratar e manter conteúdos baseados em texto. Deste modo, os objectos multimédia, como imagens, vídeo ou áudio, são armazenados mas não são alvo de qualquer tratamento ou processamento, embora seja possível a definição de metadata associada a estes objectos.

3 O modelo de dados

A necessidade de armazenar conteúdos heterogéneos, sem uma estrutura bem definida, requer cuidados especiais na concepção do modelo de dados.

A abordagem inicial perspectivava um modelo de dados algo complexo, onde toda a informação teria que ser armazenada segundo uma estrutura rígida. Ora, a diversidade de conteúdos que o projecto pretende tratar, fazia com que esta abordagem dificultasse sobremaneira o desenvolvimento aplicacional, isto porque, o modelo de dados estaria em constante mutação, de forma a integrar novos tipos de conteúdos, ir de encontro a refinamentos que se mostrassem necessários, ou a englobar novos requisitos.

À medida que fomos percebendo melhor a realidade em questão, tornou-se claro que tínhamos de optar por um modelo de dados muito mais flexível, que permitisse armazenar conteúdos de diferentes características. Veja-se, por exemplo, dois tipos de conteúdos que o sistema poderá tratar e armazenar: a informação de temperaturas nas capitais de distrito e mensagens SMS. Enquanto uma mensagem SMS pode ser vista como uma porção de texto, com uma extensão limitada, a informação das temperaturas é, provavelmente, definida como uma tabela, de uma forma perfeitamente estruturada.

A resposta para esta problemática foi encontrada no XML.

Recorrendo ao XML, foi possível definir uma forma de armazenar informação muito mais flexível que a proporcionada por um modelo relacional. Por outro lado, a estrutura definida pode ser facilmente estendida, indo de encontro a tecnologias emergentes, a novos tipos de conteúdos, ou a necessidades futuras das áreas já abordadas, que o futuro venha a ditar.

Mas, a estruturação da informação em XML apresenta outras vantagens.

Assim, dentro de uma filosofia que privilegia sistemas abertos, a adopção de XML como tecnologia de suporte permite que outras pessoas, ou entidades, a partir dos resultados do projecto, facilmente implementem novas extensões e funcionalidades sobre a plataforma SIGDiC.

Outra vantagem da adopção do XML é a possibilidade de realizar processamento sobre a informação do conteúdo. Recorrendo a transformações XSLT [10] é possível, por exemplo, exportar conteúdos sob a forma de ficheiro Word, ou PostScript/PDF, bem como, gerar páginas Web ou WAP com base em partes do conteúdo original.

A flexibilidade proporcionada pelo XML não deve (não pode!) comprometer a integridade da informação. Deste modo, a informação armazenada é validada por XML Schemas. A utilização de XML Schemas é uma mais valia, quando é considerada a possibilidade de novas funcionalidades serem desenvolvidas directamente por quem delas necessita. Através de XML Schemas, é possível estender, ou refinar, a estrutura e a semântica de uma parte do documento. O próprio XML Schema constitui um documento que, de uma forma clara e inequívoca, documenta as regras que devem ser verificadas, sem necessidade de informação adicional.

Independentemente de estar escolhida uma tecnologia de suporte, resta a necessidade de armazenar a informação.

A escolha da tecnologia de armazenamento tenta conciliar as vantagens das bases de dados relacionais, com a estruturação associada ao XML. Assim, a solução adoptada utiliza uma base de dados relacional, para permitir a optimização do processo de indexação e pesquisa, e o XML para albergar o grosso da informação de conteúdos.

A informação armazenada pela base de dados relacional é, basicamente, informação de metadata associada ao conteúdo e informação acessória para operação da aplicação de gestão. Contudo, é também armazenada alguma informação, de forma redundante com a definida em XML, com o objectivo de acelerar processos de indexação e pesquisa.

No que diz respeito ao armazenamento da informação em XML, foram analisadas duas alternativas: 1) gravar, fisicamente, a informação XML como um documento, mantendo uma referência para o ficheiro; 2) armazenar o documento XML na própria base de dados relacional, num campo específico, do tipo CLOB. A escolha recaiu sobre a segunda opção, por permitir armazenar informação de forma centralizada. Porém, a primeira opção já é considerada uma opção fiável, pelo que, se for verificada uma degradação de performance, é possível, em qualquer altura, a migração de uma solução para a outra, de forma simples e rápida.

4 A estrutura do documento XML

Para que seja possível armazenar em XML toda a informação relacionada com um conteúdo, é necessário que seja definida uma estruturação básica de suporte, que permita o armazenamento dos vários tipos de conteúdos que se pretendem tratar.

Tal como foi referido anteriormente, cada conteúdo poderá incluir vários tipos de conteúdos, com estruturas de dados próprias.

Deste modo, um documento XML que define um conteúdo é constituído por um único nó raiz, designado de "content", e como seus descendentes são definidos os vários tipos de conteúdos, bem como, a informação adicional de metadata.

Como atributo do nó raiz deverá ser obrigatoriamente incluído o código do conteúdo, que funciona como identificador único do mesmo em toda a extensão da arquitectura SIGDiC. O código do conteúdo deve, também, ser representado de forma redundante na informação de metadata.

Sob o nó raiz do documento deverão ser definidos, como filhos, os vários tipos de conteúdos. Cada um poderá ser definido de uma forma totalmente independente dos restantes. Assim, para cada filho deverá ser definida uma estrutura de dados adequada e o corresponde XML Schema, que valide a estrutura e o conteúdo da informação introduzida. É desta forma possível definir tipos de conteúdos novos, bem como, realizar alterações, ou correcções, sobre os já definidos.

O tratamento de conteúdos multimédia, como imagens, vídeo ou áudio, está fora do âmbito do projecto submetido ao PRIME, mas é necessário poder referenciá-los no sistema de informação. Assim sendo, os objectos multimédia são considerados como informação acessória, sendo os ficheiros carregados para o servidor, transparentemente, sem alterações ou processamento.

A referenciação dos objectos multimédia é efectuada no próprio documento XML, sob o elemento "media", sendo definidos vários elementos "assets", que representam e referenciam todos os objectos carregados para o servidor.

O projecto prevê, também, a ligação entre conteúdos que, de alguma forma, estejam relacionados. O estabelecimento de referências cruzadas entre conteúdos é feito através do seu código identificador, sendo suportado pelo elemento "xreference". Estes relacionamentos são bidireccionais, o que implica que se no conteúdo A for referenciado o conteúdo B é, automaticamente, criada em B uma referência para A.

Para além dos conteúdos propriamente ditos, é também necessário incluir informação adicional de metadata, com o objectivo de permitir a classificação, pesquisa e indexação dos conteúdos. Assim, para cada conteúdo e tipo de conteúdo, é incluída informação adicional de metadata, referente à autoria e criação do conteúdo.

A metadata identificada para versão actual do repositório, inclui: utilizador que criou o conteúdo, data de criação, utilizador responsável pela última alteração, data da última alteração, título e resumo.

A opção adoptada para a representação da metadata, recaiu sobre o conjunto de elementos Dublin Core V1.1[3], os quais permitem cobrir as necessidades de metadata actualmente definidas para o projecto. Porém, esta opção não limita que a informação de metadata seja estendida, incorporando, por exemplo, elementos qualificados do Dublin Core. Adicionalmente, por ser XML, a informação de metadata pode ser incluída no conteúdo e não como um conjunto de referências externas.

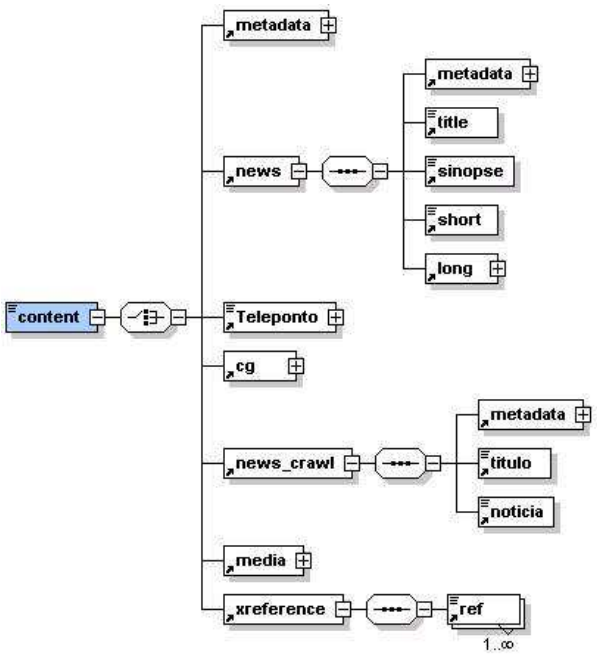

Figura 2. Exemplo de estrutura do documento

A aplicação do conjunto de elementos Dublin Core V1.1, foi realizada segundo as recomendações [4] do Dublin Core Metadata Initiative, para documentos estruturados XML, tendo todas as recomendações referidas sido implementadas.

A metadata incluída no documento XML, deve abranger, quer informação geral de metadata do conteúdo, quer informação específica de metadata directamente relacionada com cada um dos tipos de elementos. Adicionalmente, poderá ainda existir informação de metadata associada a cada referência para objectos multimédia. Desta forma, quer o elemento raiz do documento, quer qualquer dos seus filhos, representando os diversos tipos de conteúdos, deverão incluir o elemento Metadata, com o conjunto básico de elementos Dublin Core V1.1.

Qualquer gestor de conteúdos ficaria incompleto sem um sistema de controlo de versões. Neste domínio, a opção tomada foi a de implementar a funcionalidade de controlo de versões directamente em XML, em que, associado a cada instância do tipo de conteúdo, existe um atributo que identifica a versão.

O processo adoptado é o seguinte: na criação de uma nova instância de cada tipo de conteúdo, o número da versão é inicializado com o valor 1; posteriormente, qualquer alteração de algum tipo de conteúdo origina a criação no documento XML, de um novo elemento correspondente ao tipo de conteúdo alterado, com o número da versão incrementado.

Este processo faz com que seja sempre possível a recuperação de uma versão mais antiga. Por outro lado, é possível manter o controlo de versões ao nível de cada tipo de conteúdo e não ao nível macro do conteúdo.

Uma ferramenta como o SIGDiC deverá possibilitar que a difusão de alguns tipos de conteúdos esteja condicionada por aprovação editorial de um responsável (editor). Esta funcionalidade de aprovação, poderá ser obtida através da adição de um elemento específico "aprovedBy", o qual, sempre que esteja presente, confirma a existência de aprovação. O conteúdo deverá ser preenchido com o código do utilizador que aprovou ou, caso se mostre necessário constituir prova, a assinatura digital, obtida a partir do certificado digital.

5 O ambiente de edição

Com o objectivo de alargar a gama de utilização do SIGDiC a todos os sistemas operativos e, principalmente, permitir o acesso a partir de qualquer ponto, interno ou externo, sem necessidade de software específico, todo o sistema foi desenvolvido de forma a ser suportado por tecnologias Web.

Utilizando a tecnologia disponível pelo pacote DB[6] do PEAR - PHP Extension and Application Repository[5], foi possível criar a camada de isolamento, entre a camada aplicacional desenvolvida sobre PHP[7], ao mesmo tempo que era mantida a performance pela utilização de funções nativas de PHP, para acesso à base de dados.

Adicionalmente, é necessário integrar no PHP o suporte para XML, mais propriamente, o suporte da tecnologia XML DOM[1] e transformações XSLT[12][10].

Para permitir uma melhor organização dos conteúdos, o repositório obedece a uma estrutura em árvore, a qual é suportada por uma base de dados relacional.

O ecrã que representa a organização do repositório tem um layout semelhante ao de muitas outras aplicações de gestão de ficheiros (por exemplo, Windows Explorer), ou seja, do lado esquerdo é representada a árvore que corresponde à hierarquia de pastas, enquanto que do lado direito se pode visualizar os conteúdos disponíveis na pasta activa (título do conteúdo e alguma informação de metadata).

Evidentemente, nesta árvore é possível realizar operações básicas, como criar, alterar ou apagar pastas.

No repositório existe uma pasta que desempenha uma função semelhante ao "Recycle Bin"dos sistemas operativos, isto é, permite que conteúdos e pastas sejam apagados, mas que possam ser recuperados, visto não serem fisicamente apagados.

A questão realmente complexa no que concerne ao repositório é a edição de conteúdos. Dado que o repositório foi modelizado de forma a ser flexível e extensível, e em virtude de não existir uma estrutura predefinida, como se poderá criar uma interface de gestão sobre informação tão mutável e heterogénea?

A solução adoptada passa pela geração dinâmica dos formulários de edição, a partir da informação do XML Schema.

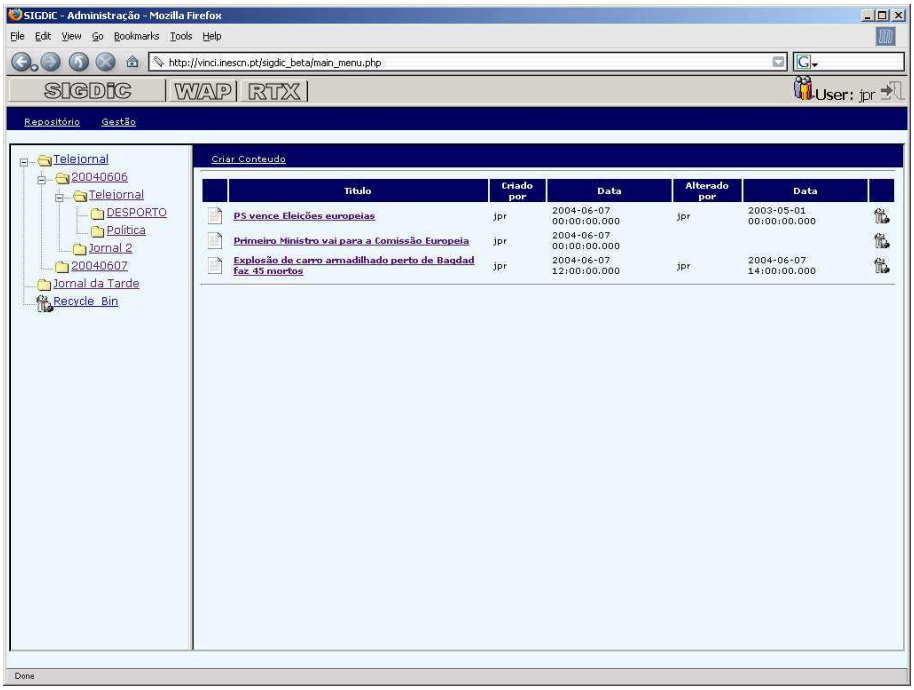

Figura 3. Repositório

Uma das funcionalidades implementadas no SIGDiC é a possibilidade de editar qualquer tipo de conteúdo. Ou seja, além dos tipos de conteúdos já definidos, que constituem um conteúdo, é possível definir novos tipos, de forma a poder incluir novos meios de difusão.

Estruturalmente, qualquer novo tipo de conteúdo tem que ter um ficheiro XML Schema e cumprir três regras:

- O elemento que define o novo tipo de conteúdo, tem que ser representado na árvore do documento XML como filho directo do elemento "content".
- O elemento que define um tipo de conteúdo, tem que ter como filho um elemento do tipo "metadata", para incluir meta-informação de acordo com as recomendações Dubin Core.
- O elemento que define um tipo de conteúdo, tem que ter um atributo "version", para permitir a implementação do sistema de controlo de versões.

A primeira regra é de caracter funcional, pois permite, no XML Schema, identificar todas as partes constituintes de um conteúdo, através da definição dos descendentes do elemento "content".

A segunda e terceira regra são de carácter mais operacional, pois garantem o suporte para as funcionalidades de metadata e controlo de versões do repositório.

A edição de um conteúdo pode implicar editar um, ou mais, tipos de conteúdos. A figura seguinte mostra um ecrã de edição de um conteúdo exemplo, com instâncias definidas para quatro tipos distintos de conteúdos.

O ecrã inclui o título do conteúdo no cabeçalho e informação de metadata, que inclui: o criador do conteúdo e a data de criação, e o responsável pela última alteração em qualquer instância de qualquer tipo de conteúdo.

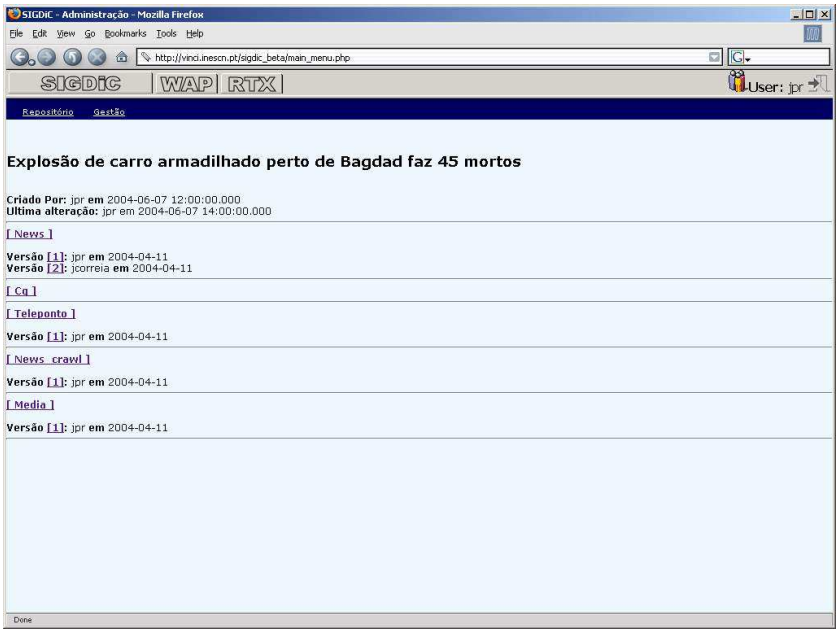

Figura 4. Tipos de conteúdos

No corpo do ecrã surge cada um dos tipos de conteúdos, já definidos, com a informação associada das versões já criadas. Para cada tipo de conteúdo está definida uma área específica.

Adicionalmente, surge também a informação de referências para objectos multimédia e a informação que relaciona, de forma cruzada, o conteúdo com outros existentes.

A acção sobre a designação, ou número da versão, de um qualquer tipo de conteúdo, origina a expansão da área associada, de forma a apresentar o formulário de edição.

Conforme já foi referido, o XML Schema que define o documento XML é usado para, dinamicamente, construir a interface de edição. Ora, dado que o schema não inclui normalmente informação que permita limitar o tamanho de cada elemento, não é possível criar um formulário perfeitamente adaptado à informação a carregar.

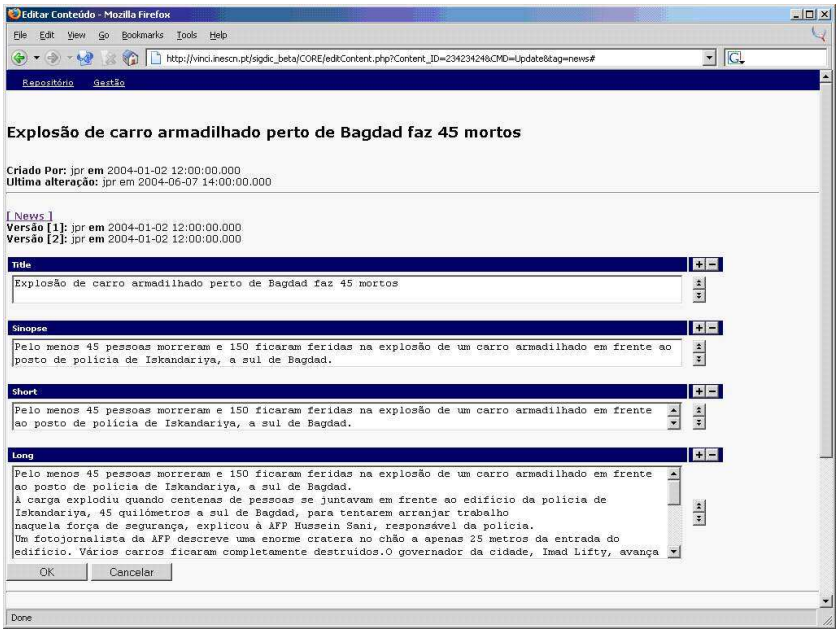

Figura 5. Edição de conteúdo

A solução passa por tratar todos os elementos de um tipo de conteúdo da mesma forma, permitindo situações "esquesitas"como, por exemplo, que o título seja mais extenso que a própria notícia. Não havendo a noção do comprimento máximo, ou pelo menos médio, para a informação de um dado elemento, a solução passou por definir uma dimensão, por defeito, que permita uma utilização confortável para a maioria dos elementos. Nos casos em que a dimensão da caixa de texto não permita uma edição confortável, sem que seja necessário recorrer frequentemente a deslocações verticais do texto, é possível ajustar a dimensão da caixa de edição, através de dois botões dispostos do lado direito. Nos casos de edição de instâncias de tipos de conteúdos já existentes, a dimensão da caixa de texto é automaticamente ajustada de acordo com a informação existente.

Outra questão importante prende-se com a possibilidade de repetição de elementos. Para certos conteúdos é possível a existência repetida de mais do que um elemento. Se permitido pelo XML Schema, para esse elemento será mostrado um conjunto de botões adicionais, que permitem criar mais um elemento no documento.

Para permitir uma melhor adequação da interface de edição, é possível, para casos particulares de conteúdos, definir uma interface de edição específica. Caso esta exista, a interface específica substitui a standard. Esta interface define apenas um front-end de edição, já que todo o processamento de informação é assegurado pelo processo standard.

6 Conclusões

Apesar de se tratar de um projecto em desenvolvimento, é já possível retirar algumas conclusões acerca da adequação de algumas das opções efectuadas.

A estruturação dos conteúdos no repositório, recorrendo a XML, permitiu garantir a adequação futura do sistema, ao permitir adicionar novos tipos de conteúdos, sem a necessidade de desenvolver uma nova versão do repositório.

A opção por uma solução de armazenamento híbrida, composta por uma base de dados relacional e documentos XML, não representou, até ao momento, qualquer problema.

A opção por desenvolver toda a aplicação de gestão numa arquitectura Web de três camadas, tem vantagens operacionais, mas apresenta alguns problemas de performance e implementação. Do ponto de vista operacional, apresenta vantagens no que diz respeito a custos de licenciamento de aplicações e manutenção do parque informático. Porém, implicou um custo e tempo de desenvolvimento superior e uma menor usabilidade, quando comparada com uma aplicação em ambiente desktop equivalente.

Foi possível garantir, através da tecnologia PEAR DB, o isolamento entre o sistema de base de dados e a camada aplicacional, não tendo sido notada qualquer degradação de performance face à utilização de métodos nativos de acesso à base de dados.

A estruturação da informação em XML no repositório, e a possibilidade de definição de novos tipos de conteúdos, impede a criação de um ambiente aplicacional perfeitamente adequado à informação a tratar. Apesar deste aspecto, conseguiu-se uma usabilidade muito próxima da normalmente obtida em aplicações Web, especialmente desenhadas para lidar com informação perfeitamente estruturada.

Foi possível desenhar uma aplicação para lidar com os tipos de conteúdos já identificados, que permite tratar, virtualmente, qualquer novo tipo de conteúdo textual.

Referências

[1] PHP DOMXML. PHP DOMXML Extension . `http://www.php.net/domxml`.

[2] Miguel Mira da Silva Hugo Alhandra. Abordagens para armazenamento de metadata em Data Warehouse usando o XML. Artigo, IST, 2003.

[3] Dublin Core Metadata Initiative. Dublin Core Metadata Initiative, 2004. `http://dublincore.org/`.

[4] Dublin Core Metadata Initiative. Guidelines for implementing Dublin Core in XML, 2004. `http://dublincore.org/documents/dc-xml-guidelines/`.

[5] PEAR. PHP Extension and Application Repository , 1999. `http://pear.php.net`.

[6] PEAR.DB. Database Abstraction Layer , 2002. `http://pear.php.net/package/DB`.

[7] PHP. Andi Gutmans, Zeev Suraski (PHP3), 1997. `http://www.php.net`.

[8] João Paulo Rodrigues. SIGDiC - Sistema Integrado de Gestão e Difusão de Conteúdos. Tese de mestrado, INESC Porto, 2004.

[9] Miguel Mira da Silva Rui Cerveira Nunes. Comparação de várias estratégias para armazenamento de XML. Artigo, IST, 2003.
[10] W3C. XSL Transformations, 1999. http://www.w3.org/TR/xslt.
[11] W3C. Extensible Markup Language (XML), 2003. http://www.w3.org/XML/.
[12] PHP XSLT. PHP XSLT Extension . http://www.php.net/xslt.