

ProcML: Padronização de processos administrativos públicos utilizando XML

JOÃO PAULO FERREIRA¹

ROBERTO SOUTO MAIOR DE BARROS²

(recebido em 17/12/2003; aprovado em 11/11/2004)

PALAVRAS-CHAVE

Processos administrativos digitais – Setor público – Internet – XML

RESUMO

Atualmente, um dos grandes desafios para o setor público é tornar mais ágil a prestação dos seus serviços. Racionalizar os custos e melhorar a qualidade dos serviços oferecidos consiste em desenvolver soluções onde a utilização da Tecnologia da Informação torna-se um dos fatores primordiais. O objetivo principal deste projeto é estabelecer um padrão, ProcML – Processos Markup Language, para a representação de processos administrativos que tramitam em órgãos públicos baseado em XML. XML é uma linguagem de marcação que provê conceitos para descrever, armazenar, intercambiar e manipular dados estruturados e semi-estruturados. Tomando como referência o banco de dados existente na UFPE – Universidade Federal de Pernambuco, propomos a criação de processos administrativos digitais. Tais processos são controlados eletronicamente e podem ser acessados pelos interessados através da Internet. Em síntese, este artigo examina o software já existente e em funcionamento na universidade, propõe um novo modelo baseado em novas tecnologias, ressaltando as vantagens do uso da linguagem XML, além de sugerir novas abordagens.

1. INTRODUÇÃO

O setor público tem no acompanhamento de processos uma de suas atividades de maior importância, permitindo que os interessados consultem os detalhes e as tramitações dos seus processos.

O volume de documentos e processos cadastrados e tramitados nos setores dos órgãos públicos causa sérios problemas quanto ao manuseio, confiabilidade e segurança. Os documentos são guardados em pastas, gerando uma grande quantidade de papel. Isso dificulta a manipulação e deixa vulneráveis dados que deveriam ser confidenciais e confiáveis.

Este artigo apresenta os detalhes do padrão ProcML [Pro04], baseado na linguagem XML – eXtended Markup Language [Ray01, Marc01], proposto para o armazenamento e intercâmbio de processos administrativos. Foi realizado um

¹ jpf@cin.ufpe.br

² roberto@cin.ufpe.br

estudo mais detalhado sobre o assunto, com a finalidade de agilizar o andamento dos processos, fornecer informações mais precisas, claras e confiáveis, tornar o acompanhamento de processos acessível através da Internet e, ainda, possibilitar a integração com outros sistemas.

Para a validação do padrão proposto foi implementado o protótipo SIPROC-Web, um sistema desenvolvido para o acompanhamento de processos da UFPE. O protótipo encontra-se atualmente em fase de testes na universidade.

O restante deste trabalho está dividido em cinco seções. Na seção 2, faz-se uma breve introdução aos padrões do W3C relevantes à compreensão deste artigo. Na seção 3, os conceitos de acompanhamento de processos e o sistema legado serão apresentados. A seção 4 descreverá o padrão ProcML. Os detalhes do estudo de caso serão descritos na seção 5. Por fim, na seção 6, serão apresentadas as conclusões.

2. PADRÕES DO W3C

Esta seção apresenta os padrões do W3C – World Wide Web Consortium que foram utilizados no desenvolvimento do ProcML e da solução proposta.

A base do padrão ProcML é a linguagem XML, no entanto, alguns outros padrões como DOM, XSL e, principalmente, XML Schema, auxiliaram no desenvolvimento do padrão e da solução proposta no estudo de caso.

DOM – Document Object Model [Haro02] é uma interface para manipulação de documentos XML. A sua especificação define uma série de objetos que representa cada um dos componentes de um documento XML. Estes objetos ficam armazenados em uma estrutura de árvore, simplificando o acesso e a manipulação dos documentos XML. A maneira como a navegação DOM flui torna este o padrão mais recomendado para a manipulação de documentos ProcML. A utilização de JDOM [JDOM4], uma API – Application Program Interface que implementa o padrão DOM em Java, tornou mais simples a manipulação (leitura e armazenamento) dos documentos ProcML.

XSL – XML Stylesheet Language [CCD+01] é uma linguagem utilizada para a criação de folhas de estilo em documentos XML e é subdivida em: XSLFO e XSL Formatting Objects e XSLT – XSL Transformations [Tidw01]. A primeira é utilizada para formatação e apresentação de documentos XML. A segunda é uma linguagem para transformação de documentos XML.

As principais utilidades da linguagem XSL são:

- adicionar elementos especificamente para visualização;
- criar novo conteúdo a partir de um já existente;
- apresentar os dados com o nível certo de detalhes para cada tipo de leitor;
- separar a apresentação da estrutura dos dados; e
- transformar documentos XML em HTML, PDF, DOC, XML, etc.

É importante ressaltar que este foi o padrão utilizado na exibição dos processos na solução proposta. Adiante, em “Exibindo um Processo”, serão dados mais detalhes de sua utilização.

2.1 Validação com XML Schema

Os documentos XML devem ser bem-formatados e, opcionalmente, válidos. Documento bem-formatado é aquele que obedece a sintaxe padrão de XML. Documento válido é aquele que segue a validação determinada por um documento à parte.

Apesar de não ser obrigatório, o uso de um padrão de validação aumenta a exatidão e a confiabilidade dos documentos XML. Os dois padrões mais utilizados para este fim são: DTD – Document Type Definition [W3C03] e XML Schema [Vlis02].

DTD é o padrão mais antigo, porém é simples e limitado, além de não ser um documento XML. Como a solução proposta por este artigo utiliza XML Schema, não serão dados mais detalhes sobre as DTDs.

XML Schema é o padrão mais recente para validação e tem como características ser um documento XML, definir tipos simples e complexos e possuir mecanismos de extensibilidade. Por ele são definidos:

- elementos que podem estar contidos em um documento;
- atributos de um elemento;
- aninhamento dos elementos;
- número e a ordem dos elementos filhos; e
- a obrigatoriedade, o tipo e o valor padrão dos elementos.

Os tipos de elementos podem ser básicos e complexos. Os tipos básicos podem ser *string*, *integer*, *boolean*, etc. Os tipos complexos ou *complex type* são aqueles definidos pelo próprio desenvolvedor. O exemplo a seguir mostra um documento XML simples e em seguida o XML Schema que o define.

```
<xml>
  <alunos>
    < Pessoa nome=“José do Nascimento”/>
    < Pessoa nome=“Paula da Silva”/>
  </alunos>
</xml>

<xs:element name=“Alunos”>
  <xs:complexType>
    <xs:sequence>
      <xs:element name=“Pessoa” type=“tpPessoa”/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name=“tpPessoa”>
  <xs:attribute name=“Nome” type=“xs:string”/>
</xs:complexType>
```

Exemplo 1: um documento XML Schema

O exemplo nos mostra que o elemento “alunos” deve ser formado por uma seqüência de elementos “pessoa” do tipo “tpPessoa” enquanto estes devem possuir os atributo “nome” do tipo *string*.

3. ACOMPANHAMENTO DE PROCESSOS

Para um amplo entendimento deste artigo serão esclarecidas nesta seção as referências teóricas utilizadas e sua abordagem prática. Serão detalhados os termos, as atividades e os conceitos utilizados durante o acompanhamento de processo e, em seguida, será feita uma breve análise do sistema legado.

3.1 Conceitos

3.1.1 Processo

Processo é a principal entidade do sistema. É armazenado fisicamente em pastas repletas de documentos (papel). Seu “ciclo de vida” se inicia na sua criação e termina no seu arquivamento. Os processos podem possuir diferentes tipos de documentos de origem, tais como: memorando, ofício, requerimento, etc.

Um processo é originado por diversos tipos de interesse ou assunto, tais como: solicitação de funcionário, solicitação de banca avaliadora, mudança de cargo, pedido de férias, etc. Qualquer atividade ou serviço que a instituição possa oferecer é requisitado através de processos.

3.1.2 Andamento

A transferência de processo entre setores é conhecida como andamento ou tramitação. Em cada setor há uma “caixa de entrada” e uma “caixa de saída” que registram o fluxo de deslocamento físico das pastas dos processos. Um processo só é enviado para outro setor quando a pessoa autorizada, denominada despachante, efetua um parecer. Um andamento pode ser cancelado pelo requisitante, ou seja, após ter chegado ao setor destino ter seu retorno solicitado.

Um exemplo de andamento é a tramitação do processo “pedido de férias”. Após sua solicitação ter sido efetuada, o referido processo tem sua origem no departamento de recursos humanos e seu destino é o setor administrativo.

3.1.3 Despacho

Ao chegar em um determinado setor o processo é avaliado por um funcionário gabaritado (despachante) para fornecer um parecer (despacho) em relação à solicitação feita. Em cada setor que o processo tramita a função do despachante é essencial para que o processo tenha andamento. Um despacho é realizado a cada andamento, arquivamento, anexação e apensação. O despacho só pode ser efetuado por despachantes, os únicos que têm poder decisório para alterar o curso ou *status* de um processo.

3.1.4 Anexação

Um processo está anexado quando faz alguma referência ou serve como complemento de outro processo. Qualquer movimentação direcionada para o processo principal será refletida em todos os seus anexados. A anexação é feita em caráter definitivo e só poderá ser desfeita pelo setor solicitante.

Um exemplo de anexação acontece quando os processos “1” e “2” que estão anexados ao processo “A” receberão o mesmo deslocamento que “A” quando este for tramitado.

3.1.5 Apensação

Um processo é classificado como apensado quando faz alguma referência ou serve como complemento de outro processo, acrescentando dados para aumentar a clareza e agilizar o seu parecer. Na apensação, assim como na anexação, qualquer movimentação recebida pelo processo principal será refletida em cada um de seus apensados. A apensação é feita em caráter temporário e qualquer setor pode dispensar um processo do principal.

3.1.6 Arquivamento

Um processo está arquivado quando este tiver seu ciclo de vida “encerrado”, seja por ter sido cancelado ou ter recebido parecer definitivo. Quando arquivados os processos não podem mais sofrer alterações ou andamentos. Entretanto, o processo pode ser desarquivado, tornando-se novamente um processo ativo, tramitável e passível de alterações.

3.2 Sistema Legado

Para a implementação do protótipo de validação do padrão ProcML foram tomados como referência os processos administrativos que tramitam na UFPE com a utilização do banco de dados existente na universidade gerado pelo SIPROC – Sistema de Acompanhamento de Processos [SIP03].

O banco de dados data do ano de 1990 e contempla cerca de 350.000 processos (arquivados e não arquivados) das suas mais diversas origens. Além de registros de quase 2.000.000 de tramitações até o presente momento.

Abaixo estão relacionadas as principais funcionalidades do sistema legado.

- Incluir, alterar e excluir processos e andamentos;
- consultar processos por interessado, setor e data de cadastro; e
- imprimir relatórios gerenciais (ex: tempo de duração de processos em setores, quantidade de processos por setores, etc.).

Durante a fase de análise do sistema legado foram encontradas algumas falhas, tais como:

- armazenamento restrito a dados de controle de processos e andamentos. O conteúdo completo (despachos e documentos) é armazenado na pasta de cada processo gerando grandes volumes de papel;
- uso de pastas tornando vulneráveis informações que deveriam ser sigilosas;

- carência de dados digitalizados aumentando a dependência da pasta do processo;
- existência de uma única cópia do processo em papel, de forma que qualquer dano ocorrido à pasta causará dano ao processo; e
- inexistência de meios de comunicação com outros sistemas.

O sistema legado utiliza uma distribuição Cliente/Servidor. Dessa forma, a aplicação cliente acessa diretamente um servidor de banco de dados. Toda a lógica de negócio está na aplicação cliente, o SGBD – Sistema de Gerenciamento de Banco de Dados realiza apenas a função de repositório de dados.

O funcionamento do sistema legado é bastante simples. As consultas, atualizações e relatórios são passados em forma de *queries* ou consultas SQL da aplicação cliente para o servidor de banco de dados. Os dados de resposta são, então, retornados para a aplicação e exibidos para o usuário.

Uma das desvantagens dessa distribuição é que qualquer alteração realizada na aplicação demanda uma atualização da versão do sistema legado em cada uma das máquinas cliente. Este problema toma grandes dimensões em uma instituição como a UFPE que possui mais de 100 setores.

4. O PADRÃO PROCML

Esta seção apresenta o ProcML, padrão criado para representação e intercâmbio de processos administrativos. A proposta apresentada neste artigo partiu do estudo do acompanhamento de processos e da análise do sistema legado. O sistema, SIPROC-Web, proposto para validação do padrão ProcML, será apresentado na seção 5.

O padrão ProcML foi modelado para atender às necessidades essenciais do acompanhamento de processos; para ser extensível, permitindo que novas propriedades sejam facilmente adicionadas sem que haja perda de compatibilidade com outros sistemas; e para poder ser utilizado por outras instituições, com diferentes modelos e particularidades no acompanhamento de processos.

Baseado em pesquisas recentes, nota-se que, atualmente, não existe nenhum padrão que utilize a linguagem XML com a finalidade de padronizar processos administrativos do setor público. A escolha da linguagem XML como padrão para definição do ProcML deve-se ao fato da sua simples utilização e de já estar consolidada tanto na área acadêmica como no mercado. O LATTES [LAT04] do CNPq é um exemplo de solução que utiliza XML para padronização e integração de plataformas.

O ProcML pode ser utilizado para o armazenamento e intercâmbio de processos entre sistemas. Todos os dados do processo encontram-se em um documento ProcML. O documento ProcML é iniciado com o cabeçalho XML e os dados básicos do processo. Em seguida, arquivamento, apensação, anexação, andamentos do processo, sendo estes opcionais. Finalmente, eventuais extensões do padrão. A figura abaixo exibe a estrutura do documento ProcML.

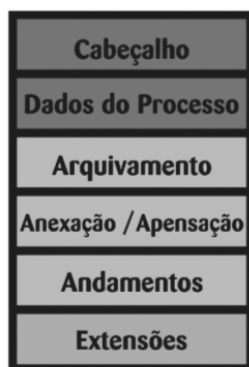


Figura 1– Estrutura do padrão ProcML

A figura mostra primeiramente o cabeçalho e os dados do processo, estes são elementos obrigatórios. Em seguida, “arquivamento”, “anexação”, “apensação” e “andamentos” que são opcionais. Por fim, têm-se extensões que podem estar no final do arquivo ou nos próprios elementos do padrão.

O exemplo a seguir ilustra o trecho de um documento ProcML com seus principais elementos: número de processo, interessado, data de cadastro, assunto, cadastrante e documento. Em seguida, cada um dos elementos será detalhado.

```
<processo>
  <numero-processo>
    99999.999999/9999-99
  </numero-processo>
  <interessado>
    <cpf>789.321.536-89</cpf>
    <nome>Renato Henriques</nome>
  </interessado>
  <data-cadastro>2002-03-09</data-cadastro>
  <assunto>PEDIDO DE BANCA</assunto>
  <cadastrante>
    <cpf>789.321.536-34</cpf>
    <nome>Rafaela Goncalves</nome>
  </cadastrante>
  <documento>
    <codigo>653</codigo>
    <descricao>
      O referido aluno solicita banca examinadora
    </descricao>
    <setor>Centro de Inform.tica</setor>
    <tipo>Ofício</tipo>
    <conteudo>
      Conteúdo do Processo
    </conteudo>
  </documento>
</processo>
```

```

        </conteudo>
    </documento>
    <andamentos>
        <andamento>
            <data-envio>2002-03-09</data-envio>
            <setor-origem>
                Secretaria de Curso
            </setor-origem>
            <usuario-origem>
                <cpf>789.321.536-40</cpf>
                <nome>Fabiana Medeiros</nome>
            </usuario-origem>
            <data-recebimento>
                2002-03-09
            </data-recebimento>
            <setor-destino>
                CoordenaÁ„o de Curso
            </setor-destino>
            <usuario-destino>
                <cpf>789.321.536-40</cpf>
                <nome>Liliana Vieira</nome>
            </usuario-destino>
            <despacho>
                <despachante>
                    <cpf>789.321.536-40</cpf>
                    <nome>Liliana Vieira</nome>
                </despachante>
                <data>2002-03-09</data>
                <conteudo>
                    O setor n„o pode resolver tal problema
                    devido à falta de documentaÁ„o necess·ria.
                </conteudo>
            </despacho>
        </andamento>
    </andamentos>
</processo>

```

Exemplo 2 : Documento ProcML

O cabeçalho do documento processo é formado pelos dados de versão inerentes ao padrão XML e pela indicação do documento XML Schema utilizado para validação.

O número do processo é a identificação do processo e tem um significado para cada instituição. Na UFPE o número do processo é formado por número da instituição, número sequencial, ano do processo e dígito verificador.

O interessado é a pessoa que iniciou o processo. O elemento interessado é do tipo `pessoaType` contendo apenas o CPF e o nome do interessado. Utilizando o mecanismo de extensão pode-se acrescentar dados como data de nascimento e endereço.

A seguir tem-se: data de cadastro, assunto indicando de forma resumida o conteúdo do processo e cadastrante, do tipo `pessoaType`, que representa a pessoa que cadastrou o processo no sistema.

O documento é a origem do processo, pode ser um ofício, carta, solicitação, etc. No padrão ProcML estão contidos: o código (número sequencial), uma breve descrição do documento, o setor de origem, o tipo de documento e, finalmente, o conteúdo do documento.

Os andamentos formam uma lista opcional de elementos andamento, que indica cada tramitação do processo. Cada andamento armazena: data de envio, setor de origem, usuário de origem, data de recebimento, setor destino, usuário destino e despacho.

O elemento despacho é obrigatório no arquivamento, na anexação, na apensação e, ainda, no andamento do processo. Ele contém o despachante, a data e o conteúdo do parecer.

Um típico documento ProcML ainda pode conter elementos como arquivamento, apensação e anexação que não estão representados no exemplo.

O arquivamento é opcional e formado por um elemento despacho. Os elementos apensação e anexação são opcionais e formados por um processo principal que indica a qual processo este está apensado ou anexado e, ainda, por um despacho.

O ProcML é regido por um documento XML Schema que define os principais elementos do processo detalhados anteriormente. O XML Schema completo do padrão ProcML foi incluído como apêndice.

4.1 Estendendo o padrão

O padrão ProcML permite que dados não modelados durante uma primeira análise ou não contemplados na versão atual do padrão possam ser incorporados sem que seja perdida a compatibilidade com outras versões e plataformas.

Para estender o padrão, o desenvolvedor deve adicionar novas *tags* XML ao documento do processo e ao documento XML Schema para que os novos dados possam ser validados.

O exemplo abaixo mostra o XML Schema de um documento ProcML que acrescenta “Data de Nascimento” aos elementos do tipo “Pessoa”.

5. ESTUDO DE CASO — SIPROC-WEB

O estudo de caso aqui descrito serviu para validação do padrão ProcML. No protótipo SIPROC-Web os processos administrativos são totalmente virtuais e padronizados. O sistema é acessado através da Internet e todas as operações são realizadas via Web.

```

<xs:complexType name="pessoaType">
  <xs:sequence>
    <xs:element ref="cpf"/>
    <xs:element ref="nome"/>
    <xs:element name="data-nascimento"
type="xs:date"/>
  </xs:sequence>
</xs:complexType>

```

Exemplo 3: Estendendo o padrão ProcML

A verificação do estudo de caso foi realizada com a implantação de um projeto piloto com os processos reais referentes ao NTI – Núcleo de Tecnologia da Informação e Reitoria da UFPE. Isso permitiu que todas as transações fossem testadas e que parte dos usuários do sistema fizesse uma análise preliminar do novo sistema.

Além das funcionalidades do sistema legado, o sistema permite a integração com outros sistemas e com o sistema legado. A versão atual contempla apenas as funcionalidades de cadastro de processos (inclusão, alteração e consulta) e, parcialmente, integração com outros sistemas (importação e exportação de processos). A produção de relatórios gerenciais foi deixada para um momento posterior, pois não se julgou essencial sua existência para a validação do padrão proposto.

5.1 Arquitetura

O sistema foi desenvolvido utilizando o padrão MVC – Model View Controller [BMR+96]. O padrão MVC prega a separação do modelo, da apresentação e do controlador. O primeiro é a representação do sistema, classes e regras de negócio. O segundo é a interface gráfica, entrada e saída de dados. O último é o controlador que faz a ligação entre o modelo e a interface gráfica.

A arquitetura do SIPROC-Web possui três camadas:

- Camada de Interface Gráfica ou Apresentação (Web);
- Camada de Aplicação ou Sistema (Java); e
- Camada de Dados (XML).

A figura a seguir apresenta a arquitetura do SIPROC-Web.



Figura 2 – Arquitetura do SIPROC – Web

A Camada de Apresentação é responsável pela entrada e exibição dos dados. É o elo de ligação entre o usuário e o sistema. Para o SIPROC-Web optou-se pelo ambiente Web pela facilidade de acesso que este oferece. Dessa forma, os usuários necessitam apenas de um navegador e acesso à Internet. O servidor Web utilizado foi o Apache/Tomcat [Tom03].

A Camada de Aplicação é responsável pelo funcionamento do sistema. Nela estão contidas as regras de negócio. Comunica-se diretamente com a Interface Gráfica e com a Camada de Dados. Foi implementada utilizando-se a linguagem de programação orientada a objetos Java [Sun03].

A Camada de Dados fica encarregada de fazer a persistência dos dados do sistema. Por persistência entende-se o armazenamento dos dados para futuras consultas e atualizações. A Camada de Dados do SIPROC-Web armazena todos os dados em documentos ProcML.

A manipulação dos documentos ProcML foi realizada utilizando-se JDOM. O acesso é feito da seguinte forma: quando a camada de dados necessita recuperar um documento XML, o JDOM segue lendo o documento e montando a árvore de objetos. Esta serve para a consulta, manipulação e a alteração dos processos.

5.2 Exibindo um processo

Na subseção anterior foi discutido que a exibição dos dados do processo é realizada na camada de apresentação. A linguagem XSL foi utilizada para exibir o conteúdo de um documento ProcML.

A utilização de XSL possibilita que, de forma simplificada, um mesmo processo seja exibido de diferentes formas de acordo com o perfil do usuário do sistema e, ainda, possa ser convertido para PDF, facilitando a impressão. A figura a seguir mostra a visualização de um processo pelo usuário.

SIPROC WEB					
Sistema de Acompanhamento de Processos					
Processo					
Número:	12345	Assunto:	Pedido de Férias 2004		
Data:	2004-08-13	Interessado:	Lilia Vieira		
Documento					
Código:	4096-4096-4096	Tipo:	Ofício		
Setor:	Reitoria	Descrição:	Pedido de Férias		
Conteúdo: Nonononon nononono nonono nonon nononononon, nonoon nononono, nononon.					
Andamentos					
Setor Origem:	Reitoria	Data Envio:	2004-08-13	Setor Destino:	Recursos Humanos
					Data Recebimento: 2004-08-14
Setor Origem:	Recursos Humanos	Data Envio:	2004-08-15	Setor Destino:	Recursos Humanos
					Data Recebimento: 2004-08-15

Figura 3 – Exibição de um processo utilizando de XSL

A exibição é feita da seguinte forma: o sistema retorna para o navegador o documento ProcML do processo e uma referência para o documento XSL que deverá transformar os dados do processo em uma página HTML. O código completo do XSL utilizado para fazer esta transformação está disponível em [Pro04].

5.3 Distribuição

A distribuição do SIPROC-Web foi estruturada da seguinte forma: os clientes acessam o servidor de aplicação. Os dados ficam armazenados em um repositório ProcML, contendo todos os documentos ProcML. A figura 4 ilustra a distribuição do SIPROC-Web.

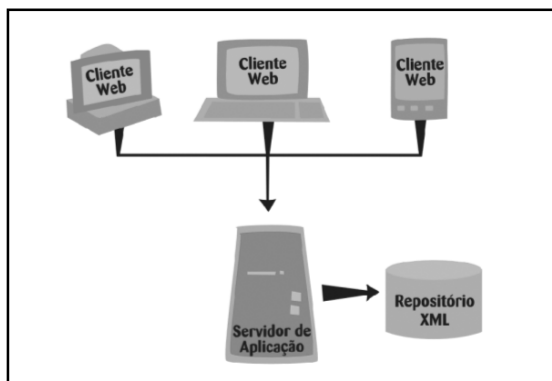


Figura 4 – Distribuição do SIPROC-Web

5.4 Funcionamento

Toda vez que um usuário solicita uma lista de processos, o sistema repassa a consulta para o controlador que, por sua vez, consulta apenas a base de dados. Assim, o usuário recebe uma lista de processos com apenas alguns dados sobre os processos.

Quando o usuário decide detalhar um processo, o controlador consulta o repositório e retorna o documento ProcML. Este pode ser mostrado diretamente para o usuário, ser convertido através da utilização de um documento XSLT, ou ainda criar um objeto a ser utilizado pelo sistema.

Quando a Interface Gráfica recebe um pedido do usuário para atualizar um processo (inserir, remover ou alterar), o pedido é repassado para o sistema e também para o controlador de dados. São feitas então as atualizações no documento ProcML referente ao processo.

5.5 Resultados obtidos

O estudo de caso foi avaliado por usuários da reitoria da universidade e do NTI. Os usuários mostraram bastante entusiasmo e segundo eles os pontos mais positivos da solução proposta foram:

- o armazenamento de todo o processo no sistema, o que facilita a visualização dos documentos de um processo e elimina a necessidade de consultas à pasta do processo; e
- todas as transações estão disponíveis na Web. No sistema legado, os usuários precisavam instalar uma cópia da aplicação, o que dificultava o lançamento de novas versões do sistema e reduz o número de usuários.

Os usuários ainda apontaram a necessidade de consultas e relatórios que estavam contemplados no sistema legado, porém estes não fizeram parte deste estudo de caso.

6. CONCLUSÕES

Este artigo apresentou a linguagem ProcML proposta para padronizar processos administrativos. O padrão foi validado através de um protótipo, SIPROC-Web, e encontra-se pronto para ser utilizado pela UFPE.

Sem um método efetivo para gerenciar os processos administrativos que tramitam entre os setores dos órgãos públicos, os usuários vivenciam muitas dificuldades para automatizar os processos envolvidos na manipulação dos documentos que são armazenados em papel.

A existência de um padrão de armazenamento e intercâmbio de processos evita que o próprio usuário se torne um integrador de processos, tendo que lidar com várias representações diferentes para os dados em um mesmo ambiente. Com XML foi possível estabelecer tal padrão, que é independente de plataforma de hardware e software.

Devido ao retorno positivo dos usuários em relação à solução proposta, a universidade estuda, agora, a migração do sistema legado para o novo SIPROC-Web. Tal migração necessitaria apenas da implementação dos relatórios contemplados no sistema legado e de um controle de acesso. No entanto, a universidade já definiu o ProcML como padrão de intercâmbio de processos administrativos entre seus diversos sistemas.

Novas técnicas em programação e processamento de dados não afetarão a estrutura dos documentos ProcML. Se for necessário acrescentar mais detalhes à estrutura dos documentos, bastará modificar os modelos e acrescentar novas *tags* onde forem requeridas.

Durante o desenvolvimento teve-se a preocupação de deixar a modelagem “aberta” de forma que novos estudos possam ser implementados sem que haja perdas na modelagem proposta. As próximas implementações serão para prover a comunicação com outros sistemas. Um dos caminhos para obter tal integração é a utilização de SOAP [STK01].

SOAP – Simple Object Access Protocol é um protocolo para a troca de dados em um ambiente descentralizado e distribuído. É totalmente baseado em XML e utiliza o protocolo HTTP para realizar o intercâmbio de dados. A simplicidade do protocolo o torna acessível a qualquer desenvolvedor. SOAP define a troca de mensagens, no formato XML, entre cliente e servidor.

SOAP permite a implementação de um *webservice* [Cera02], ou seja, um serviço disponível através da Internet que pode ser acessado por qualquer sistema, sem que seja necessário um protocolo padrão ou conhecer o seu funcionamento. Atualmente, o SIPROC-Web permite a importação e exportação *off-line* de documentos ProcML na forma de arquivos. Com a utilização do SOAP, o SIPROC-Web poderia comunicar-se com qualquer outro sistema de qualquer instituição, permitindo o intercâmbio *on-line* e a tramitação de processos, pesquisas em sistemas de recursos humanos, etc.

Com a apresentação deste trabalho, espera-se padronizar definitivamente os processos administrativos para que estes possam ser armazenados, consultados e, ainda, tramitados por diversas instituições de forma simples, ágil e segura.

ProcML: A Public Processes Standard Using XML

KEYWORDS

Digital Prozesse – Public processes – XML

ABSTRACT

One of the challenges in the public sector is to make services more efficient. Lowering costs and improving the quality of the services offered involves the development of solutions that uses Information Technology as one of its most important pillars. The main objective of this project is to establish a standard (ProcML) to represent public processes using the XML language. XML is a meta markup language used to describe, store, exchange and update structured and semi-structured data. Based on the existing processes stored in UFPE – Ferederal University of Pernambuco, we propose the development of a digital base of processes. This paper examines the legacy system used by the university and proposes a new model, based on recent technologies, emphasizing the advantages of using XML.

REFERÊNCIAS BIBLIOGRÁFICAS

- [BMR+96] BUSCHMANN, F., MEUNIER, R., ROHNERT, H., SOMMERLAD, P., STAL, M. A *System of Patterns: Pattern-Oriented Software Architecture*, John Wiley and Sons, 1996.
- [CCD+01] CAGLE, K., CORNING, M., DIAMOND, J., DUYNSTEE, T., GUDMUNDSSON, O., JIRAT, J., MASON, M., PINNOCK, J., SPENCER, P., TANG, J., TCHISTOPOLSKII, P., TENNISON J., WATT, A. *Professional XSL*, Wrox Press Inc, 2001.
- [Cera02] CERAMI, E. *Web Services Essentials*, O'Reilly & Associates, 2002.
- [LAT04] LATTES.lattes.cnpq.br, Última visita em 23 de Novembro de 2004.
- [CJD+96] COLBY, M., JACKSON, D. S., DEROSE, S. J., DUCHARME, B., DURAND, D., MYLONAS, E. *Using SGML*, Que, 1996.
- [Haro02] HAROLD, E. R. *Processing XML with Java: A Guide to SAX, DOM, JDOM, JAXP, and TrAX*, Pearson Education, 2002.
- [JDOM4] JDOM. www.jdom.org, Última visita em 23 de Novembro de 2004.
- [Marc01] MARCHAL, B. *XML By Example*. Prentice Hall PTR, 2001.
- [PK00] PFAFFENBERGER, B., KARROW, B. *HTML 4 Bible*, John Wiley and Sons, 2000.
- [Pro04] ProcML. www.joaoz.com/procml/. Última visita em 11 de Novembro de 2004.

- [Ray01] RAY, E. T. *Learning XML*, O'Reilly & Associates, 2001.
- [SIP03] SIPROC. www.processo.ufpe.br, Última visita em 11 de Outubro de 2003.
- [STK01] SNELL, J., TIDWELL, D., KULCHENKO, P. *Programming Web Services with SOAP*, O'Reilly & Associates, 2001.
- [Sun03] Java Sun. java.sun.com. Última visita em 11 de Outubro de 2003.
- [Tidw01] TIDWELL, D. *XSLT*, O'Reilly & Associates, 2001.
- [Tom03] Tomcat. jakarta.apache.org. Última visita em 11 de Outubro de 2003.
- [Vlis02] VLIST, E. V. D. *XML Schema*, O'Reilly & Associates, 2002.
- [W3C03] DTD W3C. www.w3.org/xml/1998/06/xmlspec-report. Última visita em 11 de Outubro de 2003.

SOBRE OS AUTORES

JOÃO PAULO FERREIRA

Engenheiro de Software do SIG@UFPE

Bacharel em Ciência da Computação – UNICAP

Mestrando em Ciência da Computação – UFPE.

ROBERTO SOUTO MAIOR DE BARROS

Professor Adjunto do Centro de Informática da UFPE

Bacharel em Ciência da Computação – UFPE

Mestre em Ciência da Computação – UFPE

Ph.D. in Computing Science – The University of Glasgow, Reino Unido.