

Geração de Formulários Web com base em *Templates InfoPath*

César Baptista, Nelson Branco, Luís Falcão, and Pedro Félix

Instituto Superior de Engenharia de Lisboa
Rua Conselheiro Emídio Navarro, 1, 1959-007 Lisboa, Portugal

Resumo A aplicação *InfoPath*, pertencente ao conjunto de ferramentas de escritório *Microsoft Office*, tem por função o desenho e preenchimento de formulários associados a documentos XML. Contudo, os cenários de utilização destes formulários são limitados porque a fase de preenchimento requer o uso da aplicação *InfoPath* e o acesso directo às fontes de dados.

O presente artigo descreve aspectos do desenho e implementação do sistema *WebInfoPath* (WIP), para a geração e suporte à execução de formulários web com base em *templates* gerados pela aplicação *InfoPath*. Define-se uma arquitectura extensível para este objectivo e descrevem-se aspectos da sua implementação, nomeadamente: gerador de páginas e controlos ASP.NET; biblioteca de classes para suporte à execução; e forma de integração em aplicações web.

1 Introdução

A aplicação *InfoPath*, pertencente ao conjunto de ferramentas de escritório *Microsoft Office* [7], tem por função o desenho e preenchimento de formulários associados a documentos XML [18]. Nesse contexto, um formulário é definido por: o esquema de dados da informação, um conjunto de vistas gráficas que definem as formas de visualização e inserção da informação, e parâmetros de ligações a fontes de dados.

Na fase de desenho a aplicação *InfoPath* é usada na produção do *template* contendo a informação de definição do formulário. O *template* é constituído por um conjunto de documentos XML, donde se salientam o esquema de dados (documento XSD [16]) e as vistas (documentos XSL [14]). O desenho das vistas, e conseqüente produção dos documentos XSL, é realizado através duma ferramenta de edição gráfica. A definição do esquema de dados do formulário é realizada através dum editor gráfico, ou importada duma definição externa ou inferida duma fonte de dados.

Na fase de preenchimento, a aplicação *InfoPath* recebe um *template* e apresenta a interface gráfica, definida nas vistas, para a visualização e recolha da informação. Esta informação é posteriormente armazenada em ficheiros XML com o esquema do *template* ou submetida a fontes de dados externas.

A aplicação *InfoPath* simplifica o desenho de formulários associados a documentos XML, contudo apresenta um conjunto de limitações importantes:

- O preenchimento e visualização dos formulários requer o uso da aplicação *InfoPath*.
- As fontes de dados têm de estar acessíveis directamente ao computador onde o formulário é manipulado.
- O acesso a estas fontes de dados usa as credenciais do utilizador que edita o formulário.

Estes requisitos, plausíveis no contexto *intranet*, não se verificam em cenários de aplicação na *internet*, limitando o âmbito de utilização dos formulários *InfoPath*.

O sistema WIP tem por objectivo eliminar esta limitação, gerando de forma automática páginas ASP.NET [5] com base nos *templates* criados pelo *InfoPath*. Estas páginas são assim integráveis em aplicações web desenvolvidas sob esta plataforma. O sistema WIP usa o facto da informação do *template* estar definida em idiomas XML, nomeadamente em linguagens normalizadas como o XSD e o XSL.

Este artigo apresenta a arquitectura do sistema WIP, composta por: gerador de páginas e controlos ASP.NET; biblioteca de classes para suporte à execução; e forma de integração em aplicações web. Descreve-se o processo de geração de páginas e controlos gráficos, baseado na alteração dos documentos XSL com a definição das vistas. Apresenta-se a implementação da biblioteca de apoio à execução, enfatizando-se a componente de ligação às fontes de dados suportadas pelo *InfoPath* (sistemas de gestão de bases de dados, *web services* e ficheiros XML). Descreve-se a forma de integração dos formulários gerados em aplicações web.

Este artigo está organizado da seguinte forma: na secção 2 apresentam-se os aspectos da aplicação *InfoPath* necessários à compreensão do sistema WIP; na secção 3 define-se a arquitectura deste sistema; na secção 4 descreve-se o processo de geração de páginas ASP.NET; na secção 5 caracteriza-se a biblioteca de classes com os serviços de ligação de dados e de apresentação; na secção 6 apresenta-se a forma de integração dos *templates InfoPath* em aplicações web. Finalmente, na secção 7, estabelecem-se as conclusões deste trabalho.

2 InfoPath

A aplicação *InfoPath* é uma ferramenta de escritório, incluída no *Microsoft Office 2003*, para o desenho e preenchimento de formulários. A sua operação está dividida nas fases de *desenho* e de *preenchimento*. O *InfoPath* é usado na fase de desenho do formulário para: definir o esquema de dados; desenhar as páginas do formulário, designadas de vistas; e parametrizar as ligações às fontes de dados.

O esquema de dados é representado na linguagem XSD, existindo três formas de definição: importar o documento com o esquema definido na linguagem XSD (ficheiro XSD ou documento WSDL [15]); inferir o esquema dum base de dados ou instância XML; desenhar o esquema, usando o editor gráfico do *InfoPath*.

Cada vista representa uma página do formulário e é constituída por um conjunto de controlos gráficos. A definição da vista inclui a ligação entre estes controlos e elementos do esquema de dados. Cada vista é representada através dum

documento XSL. Estas transformações recebem como entrada o documento XML com os dados da instância do formulário e produzem o documento HTML [12] com a representação gráfica do formulário. Os elementos do documento HTML estão anotados, através de atributos, com informação adicional relacionada com o *InfoPath*. São exemplo destas anotações o atributo `binding` com a expressão *XPATH* [13] que define o valor do elemento e o atributo `xctname` com o tipo de controlo.

Na fase de preenchimento, o *InfoPath* suporta a escrita e leitura dos dados em ficheiros XML, usando o esquema de dados do formulário. Suporta também a ligação dos dados do formulários (obtenção e submissão) a outras fontes de dados: sistemas de gestão de bases de dados e *web services*.

O resultado da fase de desenho é um conjunto de ficheiros, designado por *template* com [6]:

- o manifesto contendo a constituição do *template*, representado num idioma proprietário;
- os esquemas de dados, representados em XSD;
- as vistas, definidas em XSL;
- os dados por omissão e de exemplo do esquema de dados definido;
- os parâmetros de ligação às fontes de dados.

Na fase de preenchimento, o *InfoPath* é usado para:

- criar, abrir e gravar instâncias de formulários;
- inserir, visualizar e editar a informação associada à instância do formulário;
- obter e submeter informação de e para a fonte de dados.

3 Arquitectura

A arquitectura do sistema WIP, ilustrada na figura 1, é constituída por:

- gerador de páginas ASP.NET com as vistas do formulário e controlos auxiliares;
- biblioteca de classes para suporte à execução das páginas;
- *handler* ASP.NET para o processamento de pedidos a *templates*.

O gerador de páginas recebe um *template* e produz um conjunto de páginas e *user controls* ASP.NET. Cada página gerada representa uma vista do formulário e é constituída por controlos ASP.NET, presentes na biblioteca de suporte, e código gerado à *medida*. Os *user controls* representam as linhas dos controlos de repetição.

A biblioteca de suporte à execução contém:

- controlos usados nas páginas geradas;
- a classe base para as páginas geradas;
- as classes com os serviços para o armazenamento persistente dos dados dos formulários;

- as classes para a ligação (obtenção e submissão) às fontes de dados.

A integração dos formulários *InfoPath* em aplicações Web é realizada através dum *handler* específico para este tipo de conteúdos. Os pedidos HTTP [11] para o *template* são direccionados para este *handler*, que realiza as seguintes acções:

- criação das páginas com as vistas do formulário requisitado;
- criação dos objectos com os serviços de suporte à execução;
- redireccionamento do pedido HTTP para as páginas geradas.

Desta forma, o *deployment* de formulários *InfoPath* em aplicações web resume-se à cópia do *template* para a pasta da aplicação.

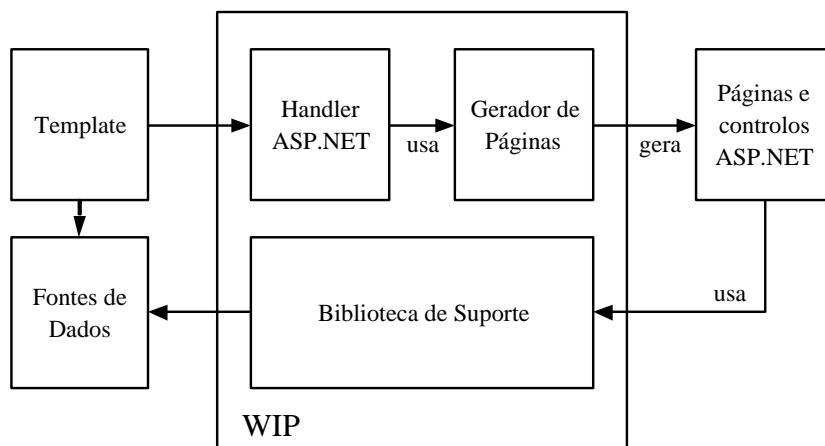


Figura 1. Arquitectura do sistema WIP.

4 Geração de formulários

Cada vista dum formulário *InfoPath* é representada por um documento XSL. Este documento descreve a transformação que produz a visualização HTML a partir do documento XML com os dados do formulário. É constituído por *templates* XSL associados a elementos do esquema de dados do formulário. Cada *template* define a visualização do elemento associado, sendo constituído por:

- Elementos literais HTML representando partes da vista, adicionados de atributos específicos do *InfoPath*.
- Elementos XSL com a selecção da informação do formulário a colocar nos elementos HTML.

O sistema WIP usa este documento XSL para gerar a página ASP.NET com a vista, através do processo descrito em seguida e ilustrado na figura 2.

Seja `view.xml` o documento com a transformação associada à vista `view` e seja `view.aspx` a página que se pretende gerar. O processo de geração de `view.aspx` está dividido em duas fases.

Na primeira fase é gerado o documento XSL `view.aspx.xsl` com a transformação para produzir a página `view.aspx` a partir dum documento XML com os dados do formulário. Esta nova transformação é baseada no documento `view.xml`.

Na segunda fase, é usada o documento `view.aspx.xsl` para gerar o ficheiro `view.aspx` com base nos dados por omissão do formulário (`sampledata.xml`).

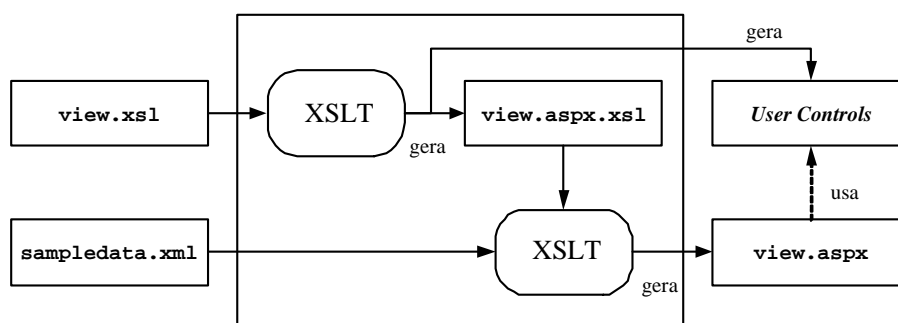


Figura 2. Geração de páginas e *user controls*.

O documento `view.aspx.xsl` é criado através da cópia de `view.xml` com as seguintes alterações:

- É inserido texto literal com o cabeçalho da página ASP.NET (`<@Page ...>`).
- É inserido no elemento `<head>` o texto literal com o código C# específico para esta vista. Este código contém a redefinição de métodos abstractos da classe base das páginas e estabelece a ligação entre os controlos da página e o serviço de ligação às fontes de dados.
- É inserido o elemento `<form>` no elemento `<body>`.
- Transformam-se todos os elementos HTML correspondentes a controlos *InfoPath* em elementos representando controlos ASP.NET. A relação entre os elementos HTML e os controlos ASP.NET é definida por um ficheiro de configuração. A biblioteca de suporte do WIP possui controlos para um subconjunto dos controlos *InfoPath*.

Os elementos com os controlos ASP.NET incluem os atributos de estilo dos elementos HTML, usados no HTML produzido na acção de *render*, e atributos adicionais usados na execução do controlo.

Neste processo é também criado um *user control* ASP.NET por cada elemento dum controlo de repetição. Este processo é idêntico ao descrito anteriormente,

com a diferença que o resultado é o próprio *user control* e não um documento XSL. Estes *user controls* são usados na criação, em tempo de execução, de elementos dos controlos de repetição.

Salienta-se que a transformação de `view.xsl` em `view.aspx.xsl` é também realizada através duma transformação XSLT.

5 Biblioteca de suporte

O WIP inclui uma biblioteca de suporte à execução das páginas ASP.NET geradas, que está dividida em dois subsistemas: **Ligação de Dados** e **WIP Controls**. Cada subsistema disponibiliza um conjunto de serviços.

O subsistema de **Ligação de Dados** inclui o Serviço de Persistência de Instâncias (SPI) e o Serviço de Ligação de Dados (SLD). O subsistema de **WIP Controls** inclui dois tipos de componentes gráficos (controlos ASP.NET): classes base das páginas e dos *user controls* criados na fase de geração; e uma hierarquia extensível de controlos gráficos, designados por Controlos WIP, que constituem as entidades existentes nos formulários web.

A biblioteca inclui implementações de referência para todos os aspectos dos subsistemas. No entanto, a sua arquitectura foi desenhada de modo a que seja extensível. Cada subsistema apresenta mecanismos próprios de extensibilidade, que serão referidos nas secções seguintes.

5.1 Ligação de Dados

No WIP, tal como no *InfoPath*, os controlos gráficos estão ligados a dados da instância corrente de formulário. Uma instância de formulário é um documento XML, instância do esquema de dados associado ao formulário.

A manipulação de formulários envolve quatro operações:

1. carregamento de dados (por omissão ou de uma instância seleccionada);
2. gravação de uma instância com os dados presentes no formulário;
3. pesquisa na fonte de dados;
4. submissão para a fonte de dados;

As operações 1 e 2 manipulam instâncias locais (tipicamente residentes no sistema de ficheiros). As operações 3 e 4 manipulam instâncias da fonte de dados principal.

Para suportar estas operações, o subsistema de **Ligação de Dados** inclui dois tipos de funcionalidades: acesso a dados locais, que suporta o carregamento e persistência de instâncias de formulários; acesso à fonte de dados principal (e secundárias, caso existam), que inclui mecanismos de interrogação e submissão de dados. Ambas as funcionalidades são disponibilizadas pelo SLD.

O carregamento e persistência de instâncias num repositório local é implementado num serviço autónomo, o SPI. Este é incluído pelo SLD que lhe delega a implementação das respectivas funcionalidades. Esta separação foi adoptada de modo a criar mais um ponto de extensibilidade do subsistema. Desta forma,

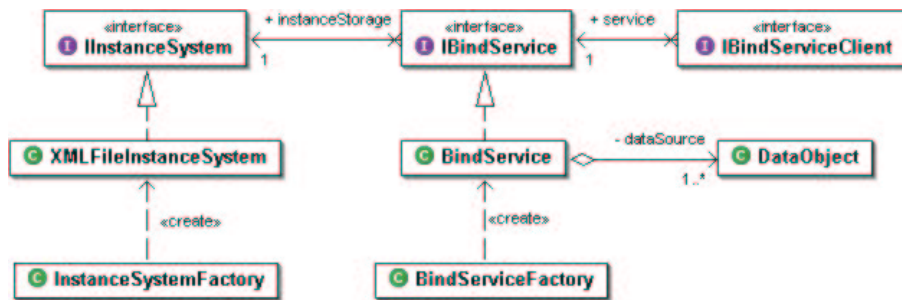


Figura 3. Diagrama de classes do subsistema de **Ligação de Dados**.

é possível mudar a implementação da persistência de instâncias sem alterar todo o mecanismo de ligação de dados.

Cada um dos serviços é representado por uma interface, `IBindService` para o SLD e `IInstanceSystem` para o SPI, conforme se ilustra na figura 3. A biblioteca inclui implementações de referência para cada um dos serviços, nas classes `BindService` e `XMLFileInstanceSystem`, respectivamente.

As instâncias dos serviços deste subsistema são obtidas através das fábricas: `InstanceSystemFactory` para instâncias de `IInstanceSystem` e `BindServiceFactory` para instâncias de `IBindService`. Os tipos concretos criados pelas fábricas podem ser definidos no ficheiro de configuração da aplicação web. Em caso de omissão são criadas instâncias dos tipos referência.

Para que os controlos WIP possam ser associados aos dados de uma instância de formulário têm que implementar a interface `IBindServiceClient`. Através desta interface, o SLD sincroniza (em ambos os sentidos) os dados com os controlos. Os dados são transferidos entre os controlos e o serviço através do documento XML que contém os dados da instância corrente do formulário.

Serviço de Persistência de Instâncias - SPI

A classe `XMLFileInstanceSystem` é a implementação de referência do SPI. Esta carrega e guarda instâncias de formulários em ficheiros XML no sistema de ficheiros local.

Serviço de Ligação de Dados - SLD

O SLD é a entidade pública deste subsistema. Realiza a ligação entre os controlos gráficos WIP e a instância corrente de formulário. Suporta as quatro operações de manipulação dos formulários referidas anteriormente.

A implementação de referência deste serviço é realizada na classe `BindService`. Delega ao SPI a manipulação local de instâncias de formulários. Accede às fontes de dados principal e secundárias (caso existam) através de objectos de

dados (`DataObject`). Estes ligam-se através de adaptadores específicos para cada tipo de fonte de dados. Um `DataObject` suporta as funcionalidades de pesquisa, submissão, ou apenas uma destas, dependendo do tipo de adaptadores com que é construído. Um adaptador tem um papel semelhante aos adaptadores existentes no ADO.NET [10].

Este subsistema inclui implementações referência de adaptadores para as seguintes fontes de dados: ficheiros XML; sistemas gestores de bases de dados com *drivers* ADO; e *web wervices*.

Os objectos associados a este serviço são criados através de fábricas. A biblioteca inclui uma fábrica de referência (`BindServiceFactory`) que cria instâncias de `BindService` para a implementação do SLD e cria instâncias do SPI através da respectiva fábrica. Por omissão, cria instâncias dos adaptadores de referência para cada um dos tipos de fonte de dados suportados. No entanto, os tipos que implementam os adaptadores podem ser definidos no ficheiro de configuração da aplicação web.

De modo a suportar outro tipo de fontes de dados e/ou outra implementação deste serviço, a fábrica de referência pode ser substituída, definindo o tipo que a implementa no ficheiro de configuração da aplicação web.

5.2 WIP Controls

O subsistema de **WIP Controls** inclui controlos ASP.NET que dão suporte à execução das páginas e *user controls* criados na fase de geração.

Páginas

A biblioteca inclui a classe `WIPPage` que estende `System.Web.UI.Page` e é a base de todas as páginas geradas no WIP. Inclui funcionalidades e recursos comuns a todas as páginas no WIP, nomeadamente:

- botões para realizar as operações de manipulação de formulários (carregar, gravar, pesquisar e submeter);
- disponibilização da instância do Serviço de Ligação de Dados (5.1);
- iniciação e libertação de recursos;

Linhas de Controlos de Repetição

A biblioteca inclui controlos ASP.NET que representam os controlos de repetição do *InfoPath*. Estes, recebem como parâmetros de construção o SLD e os controlos que representam uma linha, na forma de um *user control*, criado na fase de geração de páginas. A geração de uma nova linha corresponde à criação duma instância desse *user control* e à sua adição à lista de controlos filhos. Como os *user controls* que representam linhas partilham funcionalidades e recursos comuns, é incluída a classe `WIPUserControl`, derivada de `System.Web.UI.UserControl`, que é a base de todos os *user controls* gerados no WIP.

Web Controls WIP

As páginas WIP são compostas por controlos WIP. Estes constituem-se numa hierarquia de `web controls` ASP.NET que têm como base `WIPControl`, que estende `System.Web.UI.Control` e que implementa a interface de cliente imposta pelo SLD (`IBindServiceClient`). A classe `WIPEditableControl` é a base dos controlos editáveis (que não são de repetição), dando suporte às suas funcionalidades comuns. Na fase de geração de páginas, os elementos HTML das vistas *InfoPath* são transformados em controlos correspondentes existentes nesta hierarquia. A hierarquia de controlos WIP é extensível, incluindo a biblioteca implementações referência para os seguintes controlos: `TextBox`, `Listbox`, `Calendar`, `RepeatingTable` e `ExpressionBox`.

6 Interface do WIP

A interface de acesso aos formulários WIP é realizada através de *Uniform Resource Locators* (URLs) [4]. O URL de acesso a um formulário inclui o nome do *template* gerado pelo *InfoPath*. A vista e instância de formulário são definidos, respectivamente, pelos parâmetros “view” e “instance” na *query string* do URL. O exemplo seguinte representa o URL de acesso à vista “View1” do formulário “Form1.xsn”, com os dados da instância “Instance1”, incluída na aplicação “WIPApp” residente no servidor “someServer”:

```
http://someServer/WipApp/Form1.xsn?view=View1&instance=Instance1
```

Caso os parâmetros `view` e `instance` não constem no URL, é apresentada a vista por omissão com os dados por omissão.

6.1 Integração do WIP em aplicações Web

O processamento de pedidos para formulários WIP é realizada por um *handler* HTTP ASP.NET [9] que processa os pedidos para extensões `xsn`. A inclusão da infraestrutura WIP numa aplicação Web tem como único requisito a configuração deste *handler*.

O processo de publicação (*deploy*) de formulários consiste na cópia dos *templates* gerados pelo *InfoPath* para uma directoria da aplicação Web (configurada com o *handler* WIP).

O *handler* é um módulo de software que consulta o *template InfoPath* identificado pelo URL e redirecciona o pedido para a página ASP.NET correspondente à vista pedida. O *handler*, no primeiro acesso a um *template*, cria uma estrutura de directorias (dentro de uma directoria com o nome do *template*), onde coloca as páginas ASP.NET geradas, correspondentes às vistas, e as instâncias de formulários gravadas pelos utilizadores. O processo de geração da página ASP.NET correspondente a uma vista, é realizado pelo *handler* no primeiro acesso a essa vista, antes de redireccionar o pedido.

Quando o *handler* detecta alterações no *template* de um formulário, invalida todas as páginas geradas correspondentes às vistas que lhe estão associadas, provocando uma nova geração destas nos acessos subsequentes.

7 Conclusões

Apresentaram-se neste artigo aspectos de desenho e implementação do sistema WIP, para a geração automática de formulários web baseados em *templates InfoPath*.

A figura 4 ilustra a utilização deste sistema, apresentando um formulário original, visualizado na aplicação *InfoPath*, e o formulário web gerado pelo sistema WIP, visualizado no *browser*.

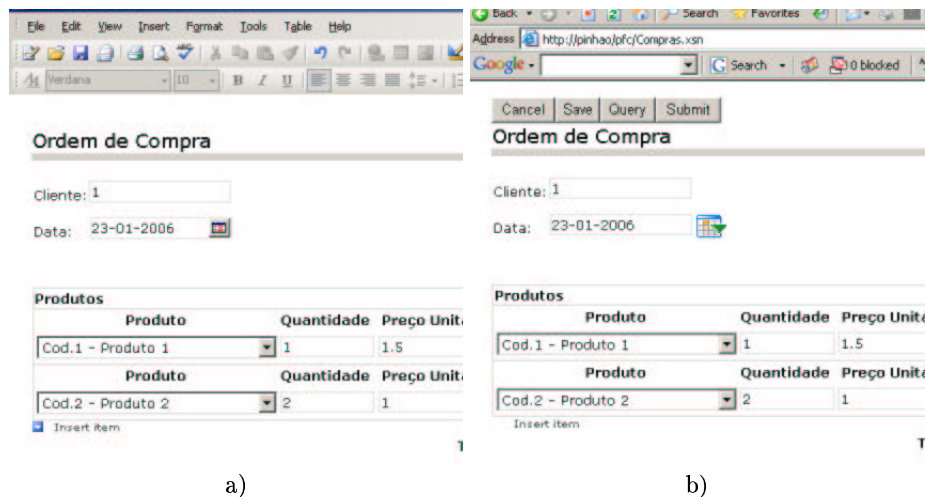


Figura 4. Exemplos de formulários: a) aplicação *InfoPath* ; b) *browser*

O sistema WIP inclui um gerador de páginas ASP.NET a partir das vistas dos formulários, uma biblioteca de classes para suportar execução destas páginas e um *handler* de atendimento de pedidos HTTP para ficheiros com os *templates* dos formulário (extensão *.xsn*).

A arquitectura do sistema WIP não está comprometida com a plataforma ASP.NET. É possível implementar este sistema noutras plataformas de suporte a aplicações web, desde que possuam as seguintes capacidades:

- Definição de *handlers* específicos para pedidos HTTP;
- Definição de controlos de servidor, preferencialmente num idioma XML;
- Mapeamento entre o modelo XML e o modelo relacional;
- Construção e processamento de mensagens SOAP [17];
- Serialização XML.

Destacam-se os seguintes aspectos do sistema WIP:

- São geradas páginas e código específico para cada vista do formulário.

- A tradução de controlos *InfoPath* em controlos ASP.NET é parametrizável pelo ficheiro de configuração na fase de geração. Esta solução possibilita a adequação dos controlos a cenários específicos, como sejam a geração de páginas para apresentação em dispositivos móveis, onde as capacidades gráficas são reduzidas.
- A componente de ligação às fontes de dados e de armazenamento persistente é extensível com novas classes fornecedoras destes serviços. É assim possível ligar a informação do formulário a qualquer fonte com interface de documento XML.
- O *deployment* de formulários requer apenas o registo do *handler* na plataforma ASP.NET e a cópia do ficheiro com o *template* para uma pasta da aplicação web.

As principais limitações deste trabalho são:

- Não suporta a associação de código *script* ao formulário. A implementação desta capacidade implica implementar todo o modelo de dados do *InfoPath* bem como interpretadores para as linguagens desejadas.
- As vistas geradas pelo *InfoPath* incluem mecanismos de formatação gráfica não normalizados, nomeadamente atributos de estilo, resultando em problemas de visualização em *browsers* diferentes do *Internet Explorer*.
- Não suporta todos os controlos do *InfoPath* e não suporta ligações a bases de dados como fontes secundárias. Prevê-se a resolução destas limitações em trabalho futuro.

O sistema WIP, ao definir um intermediário entre as páginas dos formulários e as fontes de dados, cria as condições para a realização de extensões à funcionalidade do *InfoPath*. Perspectivam-se, enquanto trabalho futuro, duas desses adições.

A primeira, consiste na adição de suporte para especificações de segurança dos *web services*, nomeadamente [8], [1] e [3]. São exemplos deste suporte a utilização da política dos *web services* no processamento das mensagens ou a utilização de *username tokens* [8] baseados nas credencias de autenticação do cliente na aplicação web.

A segunda adição perspectivada consiste no controlo do acesso às fontes de dados, realizado no serviço de ligação, usando modelos de autorização específicos para documentos XML [2].

O trabalho descrito neste artigo ilustra a utilização das tecnologias XML na geração automática de interfaces com o utilizador e na ligação destas a fontes de dados.

Referências

1. S. Bajaj, D. Box, D. Chappell, F. Curbera, G. Daniels, P. Hallam-Baker, M. Hondo, C. Kaler, D. Langworthy, A. Malhotra, A. Nadalin, N. Nagaratnam, M. Nottingham, H. Prafullchandra, C. von Riegen, J. Schlimmer, C. Sharp, and J. Shewchuk. *Web Services Policy Framework*, Setembro 2004.

2. Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pi-erangela Samarati. A fine-grained access control system for xml documents. *ACM Trans. Inf. Syst. Secur.*, 5(2):169–202, 2002.
3. G. Della-Libera, M. Gudgin, P. Hallam-Baker, M. Hondo, H. Granqvist, C. Kaler, H. Maruyama, M. McIntosh, A. Nadalin, N. Nagaratnam, R. Philpott, H. Prafull-chandra, J. Shewchuk, D. Walter, and R. Zolfonoon. *Web Services Security Policy Language*, Julho 2005.
4. L. Masinter, M. McCahill, and K. Raeburn. *RFC 1738: Uniform Resource Locators (URL)*. Internet Engineering Task Force, Dezembro 1994.
5. Microsoft. Asp.net development center. <http://msdn.microsoft.com/asp.net/>.
6. Microsoft. Infopath developer portal. <http://msdn.microsoft.com/office/understanding/infopath/default.aspx>.
7. Microsoft. Microsoft office. <http://office.microsoft.com/pt-pt/default.aspx>.
8. OASIS. *Web Services Security: SOAP Message Security 1.0*, Março 2004.
9. Fritz Onion. *Essential ASP.NET with Examples in C#*, chapter 4. Addison Wesley, 2003.
10. Shawn Wildermuth. *Pragmatic ADO.NET: Data Access for the Internet World*, chapter 5. Addison Wesley, 2002.
11. World Wide Web Consortium. *Hypertext Transfer Protocol (HTTP)*.
12. World Wide Web Consortium. *HyperText Markup Language HTML 4.01 Specification*, Dezembro 1999.
13. World Wide Web Consortium. *XML Path Language (XPath) Version 1.0 (Recommendation)*, Novembro 1999.
14. World Wide Web Consortium. *Extensible Stylesheet Language (XSL) Version 1.0 (Recommendation)*, Outubro 2001.
15. World Wide Web Consortium. *Web Services Description Language (WSDL) 1.1*, Março 2001.
16. World Wide Web Consortium. *XML Schema Language (Recommendation)*, Maio 2001.
17. World Wide Web Consortium. *SOAP Version 1.2 (Recommendation)*, Junho 2003.
18. World Wide Web Consortium. *Extensible Markup Language (XML) 1.0 (Third Edition)(Recommendation)*, Fevereiro 2004.