

# Generación dinámica de interfaces de usuario a partir de modelos representados mediante esquemas XML

Gonzalo Zarza  
zagon@fich.unl.edu.ar

Hugo Minni  
hminni@fich.unl.edu.ar

Grupo de Ingeniería Web  
Facultad de Ingeniería y Ciencias Hídricas - Universidad Nacional del Litoral  
Tel: 0054-342 4575234 - Fax: 0054-342 4575224

**Resumen** El objetivo principal de este trabajo es desarrollar un enfoque práctico para la generación dinámica de interfaces de usuario a partir de modelos basados en esquemas XML, de acuerdo a las especificaciones W3C XML Schema Structures and Datatypes. El trabajo forma parte de una de las áreas de R&D dentro del proyecto de desarrollo del Repositorio Institucional de Objetos de Aprendizaje de la Universidad Nacional del Litoral. Para generar las interfaces —a partir del modelo representado mediante esquemas XML— se propone una aplicación Web basada en tecnología Java server-side. La arquitectura de la aplicación se basa en dos bloques principales: Modelo y Vista.

## 1. Introducción

El origen de éste trabajo se encuentra en la necesidad de contar con un enfoque dinámico para la generación de interfaces de usuario. Necesidad que se hizo evidente durante la etapa de diseño del Repositorio Institucional de Objetos de Aprendizaje de la Universidad Nacional del Litoral [1]. Además, en el diseño del repositorio —en base al estándar IEEE 1484.12.1 Learning Object Metadata [2]— se busca aplicar la definición del XML Schema Language Binding for LOM [3].

Actualmente, la generación de interfaces está siendo encarada principalmente a través de soluciones estáticas, específicamente desarrolladas para cumplir con los requisitos impuestos por modelos puntuales. Esta situación hace que sea casi imposible la reutilización de las interfaces debido, principalmente, al esfuerzo necesario para lograr este objetivo y a la complejidad de estas tareas. Aún el más mínimo cambio en los requerimientos del modelo implica una cantidad de esfuerzo considerable para adaptar las interfaces a los mismos.

## 2. Objetivos

El enfoque práctico de este trabajo se enmarca en el proyecto de desarrollo del Repositorio Institucional de Objetos de Aprendizaje. Este repositorio será utilizado en un contexto educativo universitario, en reemplazo de los procedimientos desarticulados que se encuentran en uso en la actualidad.

El proyecto principal está compuesto por varias áreas de investigación y desarrollo. Una de éstas áreas incluye la implementación del repositorio, cumpliendo con la especificación de Learning Object Metadata (LOM) [2], y el almacenamiento canónico de sus metadatos cumpliendo con el XML Schema Definition Language Binding for LOM [3]. El desarrollo de las interfaces basadas en el esquema LOM —origen del presente trabajo— forma parte de un área de investigación distinta.

Considerando el hecho de que las entidades principales de estas propuestas son tanto el conocimiento como así también los recursos humanos, la estructura básica de los metadatos puede cambiar

debido a diferentes necesidades y/o a nuevos requerimientos. Por lo tanto, es necesario encarar este problema mediante interfaces que no dependan de modelos particulares. Esto evitaría la necesidad de adaptar las interfaces a los cambios en la estructura de los metadatos.

### 3. Enfoque propuesto

El enfoque propuesto en este trabajo para la generación dinámica de interfaces de usuario, se sustenta sobre dos pilares fundamentales que deben ser definidos. Por un lado, la representación del modelo de origen que contiene los datos o requerimientos de las interfaces. Por el otro, la aplicación que toma estos datos y a partir de ellos genera dinámicamente las interfaces de usuario, adaptándose de manera automática a cambios en el modelo de entrada. Esto último permite que la aplicación no sea dependiente de modelos particulares y por lo tanto la generación de las interfaces no sea estática.

El sistema completo de generación de interfaces se puede apreciar en la Fig. 1.

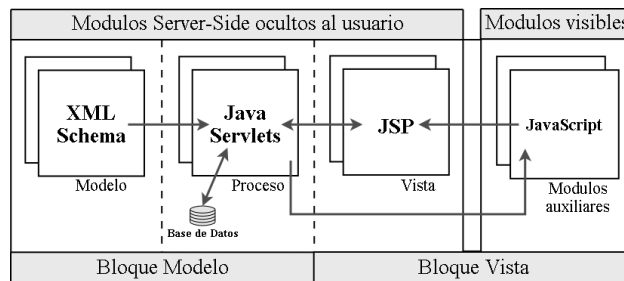


Figura 1. Diagrama arquitectónico del sistema de generación de interfaces

A continuación se describen tanto la representación del modelo como la aplicación propuesta para la generación de las interfaces.

#### 3.1. Representación del modelo

La generación de las interfaces se realizará a partir de los datos aportados por un modelo de entrada. Este modelo debe contener la información necesaria sobre los campos, tipos de datos y multiplicidades, entre otros, a ser generados e incluidos en las interfaces resultantes.

El modelo será representado a través de esquemas XML, definidos según las recomendaciones de Structures [4] y Datatypes [5] del World Wide Web Consortium (W3C).

#### 3.2. Generación de interfaces

Para la generación de las interfaces se propone una aplicación Web basada en tecnología Java [6] de lado servidor. Esta aplicación, toma como datos de entrada la representación del modelo descrita en el punto anterior.

La arquitectura de la aplicación propuesta está compuesta por dos bloques principales: Modelo y Vista.

El bloque Modelo utiliza Java Architecture for XML Binding (JAXB) [9] para transformar y/o adaptar a un modelo de objetos el esquema utilizado como entrada. Una vez finalizada esta tarea, se expone el modelo resultante al proceso de análisis que es llevado a cabo por Servlets Java [8]. El mismo proceso invoca los métodos de validación adecuados según los tipos de datos de los elementos que constituyen el modelo. Finalmente, los datos resultantes son enviados al bloque Vista.

El bloque Vista está compuesto por JavaServer Pages (JSP) [7] y clases de utilidades Web escritas en JavaScript [13]. Estas clases se asocian a los elementos del modelo de objetos; sus métodos aprovechan las ventajas de los comportamientos adaptativos del W3C Document Object Model (DOM) [10] para realizar la verificación de la sintaxis y la validación de los datos.

La aplicación agrega dos características muy importantes a las JSP [7] que las distinguen de las JSP [7] estándar. La primera es su comportamiento adaptativo, basado en el W3C DOM [10], mediante el cual los elementos opcionales y/o compuestos del modelo (que se corresponden con los tipos de datos no requeridos y complejos, respectivamente) se visualizan en base a pedidos en lugar de por defecto. La segunda —y quizás la más importante— característica es la utilización de librerías de etiquetas (tag libraries) definidas especialmente para mejorar los tipos de datos complejos presentes en la definición de esquemas XML [4], [5].

## 4. Conclusiones

Partiendo de los requerimientos iniciales, el enfoque práctico propuesto en este trabajo para la generación dinámica de interfaces de usuario, alcanza resultados interesantes.

La aplicación puede ser utilizada para generar interfaces complejas que se ajusten de manera precisa a los requisitos de un modelo específico que se representa mediante un esquema XML.

Al mismo tiempo, se logra cumplir totalmente con los requerimientos iniciales más importantes: la no dependencia de un modelo particular y la reusabilidad del sistema de generación de interfaces. Esto se debe principalmente a que ya no es necesario realizar cambios en la aplicación ni en las interfaces generadas por la misma para adaptarlas a los cambios que se puedan y/o necesiten introducir al modelo utilizado como entrada.

Otra importante contribución radica en el novedoso enfoque utilizado para definir y manejar las ocurrencias de los tipos de datos complejos de los esquema XML. Para realizar esta tarea, fueron definidas librerías de etiquetas especiales a medida con el objetivo de tratar estas ocurrencias como un único elemento, conservando al mismo tiempo las propiedades de los elementos que la componen.

## 5. Marco y continuación del trabajo

Los desarrollos presentados en este trabajo junto con la implementación de los mismos, forman parte del Proyecto Final de Carrera de Ingeniería Informática del autor Gonzalo Zarza. Dicho Proyecto Final tiene como director al MSc. Hugo Minni (coautor)

Este trabajo se enmarca en una de las áreas de R&D del Proyecto de desarrollo del Repositorio Institucional de Objetos de Aprendizaje de la Universidad Nacional del Litoral.

Ambos autores forman parte del Grupo de Ingeniería Web de la FICH-UNL y del Proyecto de R&D mencionado anteriormente.

## Referencias

1. Godoy R. J., Minni H., Zarza G., y Loyarte H., “*Design Criteria for the Development of an Institutional Learning Object Repository*”, XII Congreso Argentino de Ciencias de la Computación. San Luis, Octubre 2006.
2. IEEE Computer Society, “*IEEE Standard for Learning Object Metadata*”, IEEE Standard 1484.12.1, New York. 2002.
3. IEEE Computer Society, “*IEEE Standard for Learning Technology - Extensible Markup Language (XML) Schema Definition Language Binding for Learning Object Metadata*”, IEEE Standard 1484.12.3, New York. 2005.
4. W3C Recommendation, “*XML Schema Part 1: Structures Second Edition*” W3C Working Draft, Octubre 2004.
5. W3C Recommendation, “*XML Schema Part 2: Datatypes Second Edition*” W3C Working Draft, Octubre 2004.
6. Sun Microsystems Inc., “*The Java Language Specification, Third Edition*”, Mayo 2005.
7. Sun Microsystems Inc., “*JavaServer Pages Specification Version 2.1*”, Final Release, Mayo 2006.
8. Sun Microsystems Inc., “*Java Servlet Specification Version 2.4*”, Final Release, Noviembre 2003.

9. Sun Microsystems Inc., "*The Java Architecture for XML Binding (JAXB) 2.0*", Early Draft v0.4, Junio 2004.
10. W3C Recommendation, "*Document Object Model (DOM) Level 1 Specification (Second Edition)*", W3C Working Draft, Septiembre, 2000.
11. W3C Recommendation, "*HTML 4.01 Specification*", W3C Working Draft, Diciembre 1999.
12. W3C Recommendation, "*Cascading Style Sheets (CSS) 2.1 Specification*", W3C Working Draft, Noviembre 2006.
13. Mozilla Foundation, "*Core JavaScript 1.5 Reference*", Mozilla Developer Center, Noviembre 2006.