

Presentación de Proyecto de Investigación

1. Datos generales

1.1 Título del proyecto

Interoperabilidad en Lenguajes de Programación Orientados a Objetos

1.2 Resumen del proyecto

La interoperabilidad es una característica importante en los sistemas distribuidos basados en objetos, ya que permite la comunicación de programas (clientes y servidores), escritos en diferentes lenguajes de programación orientados a objetos. Existen algunos problemas fundamentales que tienen que ser resueltos por los diferentes lenguajes para alcanzar la interoperabilidad. Algunos de estos problemas están relacionados con el mapeo de tipos de datos, la representación de los objetos, los mensajes, la serialización y la deserialización.

Para resolver los problemas de interoperabilidad antes descritos, en este proyecto se desarrollarán herramientas para serializar y deserializar objetos a XML, las cuales serán bibliotecas independientes, y serán capaces de serializar objetos en un conjunto de lenguajes de programación orientados a objetos (Python, Ruby, PHP) a XML, y de regreso. Una de las principales características de estas herramientas será la generación de representaciones XML estándar, las cuales serán independientes del lenguaje de programación.

1.3 Nombre y datos personales de los participantes

1.3.1 Participantes internos al proyecto

Universidad Autónoma Metropolitana, Unidad Cuajimalpa, México
Departamento de Tecnologías de la Información

- Dr. Carlos Roberto Jaimez González (**Responsable del proyecto**)
Experiencia en servicios Web, interoperabilidad en sistemas distribuidos y lenguajes de programación. Colaborará en todas las actividades.
Tiempo de dedicación a este proyecto: 4 horas a la semana.

También podrán participar estudiantes de Licenciatura y Maestría de la Universidad Autónoma Metropolitana.

2. Justificación y planteamiento del objeto de estudio

La interoperabilidad es una característica importante en los sistemas distribuidos basados en objetos, ya que permite la comunicación de programas (clientes y servidores), escritos en diferentes lenguajes de programación orientados a objetos. Existen algunos problemas fundamentales que tienen que ser resueltos por los diferentes lenguajes para alcanzar la interoperabilidad. Algunos de estos problemas están relacionados con el mapeo de tipos de datos, la representación de los objetos, los mensajes, la serialización y la deserialización.

Mapeo de tipos de datos. Los tipos de datos son uno de los principales problemas para alcanzar la interoperabilidad entre diferentes lenguajes de programación. Debe de existir un acuerdo de mapeo entre los tipos de datos de los lenguajes de programación que se desean comunicar. Una forma de resolver este problema es mediante una tabla de mapeo con los diferentes tipos de datos soportados por los lenguajes de programación.

Representación de objetos. Se debe establecer una forma estándar de representar objetos, ya sea que estén escritos en Java, C#, C++, o algún otro lenguaje de programación orientado a objetos. El formato estándar debe considerar la representación de las estructuras y tipos de los diferentes lenguajes de programación, tales como clases, tipos de datos primitivos, arreglos, y clases definidas por el usuario.

Mensajes. Los mensajes representan la forma en que los clientes y servidores se comunican, a través de solicitudes o recepción de respuestas. Los mensajes también deben de estar escritos en una forma estándar para que puedan ser entendidos por ambas partes.

Serialización y deserialización. En el contexto de almacenamiento de datos y transmisión, la serialización es el proceso de transformar un objeto a un estado en el que pueda ser almacenado permanentemente a un medio tal como un archivo, una base de datos, o un flujo para ser transmitido a través de la red. Deserialización es el proceso inverso, el cual transforma la versión serializada del objeto en un objeto vivo. La serialización y la deserialización son procesos ampliamente utilizados cuando se trabaja con sistemas distribuidos basados en objetos.

Para resolver los problemas de interoperabilidad antes descritos, en este proyecto se desarrollarán herramientas para serializar y deserializar objetos a XML, las cuales serán bibliotecas independientes, y serán capaces de serializar objetos en un conjunto de lenguajes de programación orientados a objetos (Python, Ruby, PHP) a XML, y de regreso. Una de las principales características de estas herramientas será la generación de representaciones XML estándar, las cuales serán independientes del lenguaje de programación. Cabe señalar que el XML generado por la herramienta, será interoperable con el XML generado por los serializadores WOX [1, 2] (Web Objects in XML) existentes, los cuales fueron desarrollados en los lenguajes de programación Java y C#, y pueden ser descargados del sitio web de WOX [3].

3. Antecedentes históricos, teóricos y conceptuales

En esta sección se explican los conceptos de serialización y deserialización, así como sus procesos. Además, se proporciona una explicación de los conceptos y características que tendrán las herramientas que se desarrollarán en este proyecto, mismas que servirán para evaluarlas.

La serialización consiste en un proceso de transformación de un objeto en un medio de almacenamiento (como puede ser un archivo, o un buffer de memoria) con el objetivo de transmitirlo a través de una red como una serie de bytes o en otro formato como XML. La serie de bytes o el formato pueden ser usados para crear un nuevo objeto (deserializarlo), el cual será idéntico al original.

3.1 Serialización de objetos a XML

Para serializar objetos de algún lenguaje de programación orientado a objetos a XML, básicamente se sigue el siguiente proceso.

1. Se obtiene el nombre, tipo y valor de cada uno de los atributos del objeto. Esto se realiza mediante la *reflexión*, la cuál es la capacidad que tiene un programa para observar y opcionalmente modificar su estructura de alto nivel; por medio de esta capacidad es posible acceder a la información de los objetos, conociendo y/o ejecutando sus atributos y métodos públicos, todo ello en tiempo de ejecución. También se utiliza la *introspección*, la cual se utiliza para la obtención del tipo de dato de un atributo del objeto.

2. Una vez que se obtiene el nombre y valor de cada atributo del objeto a serializar, se escribe en un archivo XML. En caso de que el valor no sea tipo primitivo sino un objeto, habrá que representar también en el documento todos los atributos de este objeto.

3.2 Deserialización de objetos desde XML

El proceso de deserialización es el contrario al de serialización; en dicho proceso se siguen los siguientes pasos.

1. Se extrae información acerca de un objeto del archivo XML.
2. Se crea una clase con la información obtenida del XML.
3. Se crea una instancia el objeto con la información obtenida del XML y utilizando la clase creada.

3.3 Características de las herramientas

Las herramientas de serialización y deserialización deben cubrir las siguientes características, mismas que servirán para evaluarlas.

1. Interoperabilidad: Es la posibilidad de serializar un objeto en un lenguaje de programación y deserializarlo en uno distinto, y viceversa. Por ejemplo: Serializar un objeto en lenguaje PHP y deserializarlo en Python.

2. Convertir directamente objeto a XML: La posibilidad de serializar un objeto *automáticamente* por medio de un método, sin necesidad de modificar el objeto directamente, ni tampoco crear código fuente adicional.

3. Convertir directamente XML a objeto: La posibilidad de obtener uno o más objetos *automáticamente* a partir de un método y un archivo XML, sin necesidad crear código fuente adicional.

4. XML bien formado: Esta característica se refiere a un documento en el cual las etiquetas están correctamente anidadas.

3.4 Herramientas similares

No existen herramientas que serialicen y deserialicen objetos a y desde XML, las cuales sean interoperables entre diferentes lenguajes de programación. En esta sección presentamos algunas herramientas que serializan objetos Java a XML; sin embargo, ninguna de ellas es capaz de generar una representación estándar para poder interoperar con otros lenguajes de programación.

XStream. Es una biblioteca implementada en Java [4] para serializar objetos a XML y de regreso. XStream es capaz de serializar la mayoría de los objetos Java sin necesidad de mapeos personalizados.

Koala. Koala XML serialization [5] es una aplicación Java que proporciona una forma para serializar y deserializar cualquier objeto Java a XML. El mecanismo de serialización de Koala utiliza la interfaz *Serializable* de Java.

XMOP. XML Metadata Object Persistence [6] permite cierto grado de interoperabilidad entre Java y lenguajes Microsoft únicamente. En el caso de Java, proporciona serialización solamente para clases muy sencillas. Una gran desventaja es que si las clases contienen atributos con tipos de datos cadena o arreglos, los desarrolladores deben de implementar los métodos de serialización.

JSX. Java Serialization to XML [7] serializa solamente objetos Java a XML en un formato entendible, además de utilizar las técnicas de la serialización de objetos Java.

Castor. El proyecto Castor [8] es un framework de datos de código abierto escrito en Java, el cual proporciona serialización Java a XML y persistencia Java a SQL. Una gran restricción es que las clases Java deben seguir las convenciones de Java Bean para poder ser serializadas.

JAXB. La arquitectura JAXB [9] es un framework para procesamiento de documentos XML. JAXB es capaz de deserializar un documento XML a un árbol de objetos, con sus nodos que corresponden a elementos XML, los cuales contienen atributos y el contenido son variables de instancia. JAXB requiere de la definición de esquemas XML para mapear a objetos Java.

Aunque existen otras bibliotecas para serializar objetos o mensajes a XML y viceversa, son parte de frameworks de servicios web, tales como XML-RPC [10], o implementaciones de SOAP [11]. Estos frameworks utilizan serialización XML internamente, y no son bibliotecas independientes.

4. Preguntas y supuestos de investigación

Dentro de las preguntas de investigación tenemos las siguientes:

1. ¿Qué ventajas se tienen al proporcionar interoperabilidad entre aplicaciones escritas en diferentes lenguajes de programación?
2. ¿Qué tipo de aplicaciones pueden hacer uso de estas herramientas?
3. ¿Qué problemas se encuentran al desarrollar estas herramientas para lograr interoperabilidad?

Dentro de los supuestos del proyecto tenemos los siguientes:

1. Estas herramientas facilitarán la interoperabilidad de aplicaciones escritas en diferentes lenguajes de programación.
2. Estas herramientas proporcionarán una representación XML de objetos que será interoperable entre diferentes lenguajes de programación orientados a objetos.
3. La documentación y ejemplos que se proporcionarán en los sitios Web será suficiente para utilizar estas herramientas.

5. Objetivo(s)

Objetivo General

Diseñar e implementar herramientas interoperables para serializar y deserializar objetos a y desde XML.

Objetivos específicos

1. Desarrollar serializadores de objetos a XML en un conjunto de lenguajes de programación orientados a objetos (Python, Ruby, PHP).
2. Desarrollar deserializadores de XML a objetos en un conjunto de lenguajes de programación orientados a objetos (Python, Ruby, PHP).
3. Desarrollar módulos para generación de clases en un conjunto de lenguajes de programación orientados a objetos (Python, Ruby, PHP), a partir de la representación XML de un objeto.
4. Realizar pruebas de funcionalidad, interoperabilidad y desempeño a las herramientas desarrolladas.
5. Desarrollar sitios Web con documentación y ejemplos para las herramientas.

6. Metodología

Para el diseño e implementación de las herramientas interoperables se consideran 5 etapas para cada una de ellas.

6.1. Desarrollar serializador de objetos a XML

En esta etapa se creará un módulo en el lenguaje de programación elegido (Python, Ruby, PHP), de un serializador capaz de serializar objetos de ese lenguaje a una representación XML. Los tipos de datos que podrán ser serializados y no serializados así como su justificación serán especificados a través de una tabla de mapeo.

6.2. Desarrollar deserializador de XML a objetos.

En esta fase se creará un módulo en el lenguaje de programación elegido (Python, Ruby, PHP), para un deserializador capaz de deserializar un objeto a partir de su representación XML. El formato XML estará basado en la serialización de Web Objects in XML (WOX). Los tipos de datos que podrán ser serializados y no serializados así como su justificación serán especificados a través de una tabla de mapeo.

6.3. Desarrollar módulo para generación de clases.

En esta etapa se creará un módulo en el lenguaje de programación elegido (Python, Ruby, PHP), para la generación de clases a partir de la representación XML. Este módulo tomará la estructura del archivo XML para generar una clase con el nombre indicado en XML, sus atributos y constructor correspondientes.

6.4. Realizar pruebas de funcionalidad, interoperabilidad y desempeño.

En esta fase se realizarán las diferentes pruebas a cada una de las herramientas desarrolladas: de funcionalidad, interoperabilidad con otras herramientas y desempeño al serializar y deserializar.

6.5. Desarrollar sitio Web con documentación y ejemplos.

En esta etapa se desarrollará un sitio web que tendrá la siguiente información acerca de cada una de las herramientas desarrolladas: instrucciones de instalación; descripción de cada uno de los módulos de la herramienta; información acerca de alcances de la herramienta y limitaciones; tabla de mapeo; ejemplos de uso; herramientas con las que se tiene interoperabilidad.

7. Bibliografía

- [1] Carlos R. Jaimez-González, Simon M. Lucas, "Easy XML Serialization of C# and Java Objects". In Proceedings of the Balisage The Markup Conference 2011, Balisage Series on Markup Technologies, Volume 7 (2011), Montreal, Canada, 2-5 August 2011.
- [2] Carlos R. Jaimez-González, Simon M. Lucas, "Interoperability of Java and C# with Web Objects in XML". In Proceedings of the International Conference e-Society (ES 2011), ISBN: 978-972-8939-46-5, pp. 518-522, Avila, Spain, 10-13 March 2011.
- [3] Web Objects in XML (WOX) serializer website. Disponible en: <http://woxserializer.sourceforge.net/>. Último acceso en Junio 2014.
- [4] XStream: A Java Library to Serialize Objects to XML, 2010. Disponible en: <http://xstream.codehaus.org/index.html/>.
- [5] Koala XML serialization, 1998, available at <http://old.koalateam.com/xml/serialization/>.
- [6] XMOP: XML Metadata Object Persistence, 2000. Disponible en: <http://www.openhealth.org/documents/xmop.htm/>.
- [7] JSX: Java Serialization to XML, 2002. Disponible en: <http://jsx.org/>.
- [8] Castor. The Castor Project, 2010. Disponible en: <http://www.castor.org/>.
- [9] JAXB: The Java Architecture for XML Binding, 2010. Disponible en: <https://jaxb.dev.java.net/>.
- [10] Winer D. XML-RPC specification, 1999, available at <http://www.xmlrpc.com/spec>.
- [11] WWW Consortium. Latest SOAP Versions, 2003, available at <http://www.w3.org/tr/soap/>.

8. Metas (expresadas en productos de investigación)

Este proyecto generará cinco tipos de resultados a lo largo de sus tres años de duración: el desarrollo de varios prototipos de herramientas interoperables para serializar y deserializar objetos a y desde XML, la publicación de estos resultados en congresos (nacionales e internacionales) y en revistas indexadas, y la formación de recursos humanos mediante la dirección de proyectos terminales.

1. Prototipos: A lo largo del proyecto se desarrollarán varios prototipos de herramientas interoperables para serializar y deserializar objetos a y desde XML, en un conjunto de lenguajes de programación orientados a objetos (Python, Ruby, PHP).

2. Publicaciones en Congresos: La publicación de los resultados de este proyecto será plasmada en al menos tres artículos en congresos arbitrados nacionales o internacionales, tales como *ACM/IFIP/USENIX Middleware Conference*, la *International Conference on Web Information Systems and Technologies (WEBIST)*, *Balisage: The Markup Conference 2014*, la *XML London Conference*, la *International Conference on Applied Computing*, la *International Conference on WWW/Internet*, o el *Congreso Internacional de Ciencias de la Computación*.

3. Publicaciones en Revistas Indexadas: Se tendrán al menos dos publicaciones en revistas indexadas, tal como el *Journal Expert Systems with Applications (JCR-Elsevier)*, o el *Journal Research in Computing Science (Latindex)*.

4. Dirección de Proyectos Terminales: Se dirigirán al menos 2 proyectos terminales de Licenciatura, asociados a los temas de investigación y desarrollo de este proyecto. También se buscará la posibilidad de dirigir un proyecto terminal de Maestría.

9. Cronograma de actividades (trimestral y anual)

Este proyecto tiene una duración prevista de tres años.

Primer año, Primer trimestre:

- Estudio de las herramientas para serializar y deserializar objetos a y desde XML, en un conjunto de lenguajes de programación orientados a objetos (PHP, Ruby, Python). Revisión de literatura sobre temas como: serialización, deserialización, interoperabilidad en lenguajes de programación orientados a objetos.

Primer año, Segundo trimestre:

- Desarrollo de serializadores de objetos a XML en un conjunto de lenguajes de programación orientados a objetos (Python, Ruby, PHP).

Primer año, Tercer trimestre:

- Finaliza desarrollo de serializadores de objetos a XML en un conjunto de lenguajes de programación orientados a objetos (Python, Ruby, PHP).
- Desarrollo de deserializadores de XML a objetos en un conjunto de lenguajes de programación orientados a objetos (Python, Ruby, PHP).

Segundo año, Primer trimestre:

- Finaliza desarrollo de deserializadores de XML a objetos en un conjunto de lenguajes de programación orientados a objetos (Python, Ruby, PHP).

Segundo año, Segundo trimestre:

- Desarrollo de módulos para generación de clases en un conjunto de lenguajes de programación orientados a objetos (Python, Ruby, PHP), a partir de la representación XML de un objeto.

Segundo año, Tercer trimestre:

- Finaliza desarrollo de módulos para generación de clases en un conjunto de lenguajes de programación orientados a objetos (Python, Ruby, PHP), a partir de la representación XML de un objeto.

Tercer año, Primer trimestre:

- Integración de todos los módulos desarrollados.
- Pruebas de funcionalidad, interoperabilidad y desempeño a las herramientas desarrolladas.

Tercer año, Segundo trimestre:

- Finalizan pruebas de funcionalidad, interoperabilidad y desempeño a las herramientas desarrolladas.
- Desarrollo de sitios Web con documentación y ejemplos para las herramientas.

Tercer año, Tercer trimestre:

- Finaliza el desarrollo de sitios Web con documentación y ejemplos para las herramientas desarrolladas.

10. Requerimientos y justificación de los recursos solicitados

10.2 Infraestructura, equipamiento y recursos materiales (expresados en rubros de gasto)

A partir de esta propuesta de proyecto de investigación, se ha elaborado el siguiente presupuesto. Los montos aproximados por año, son los siguientes:

Primer año

Inscripción a congreso internacional especializado, en el cual se presentará un artículo de investigación. Posibles congresos: <i>ACM/IFIP/USENIX Middleware Conference</i> , la <i>International Conference on Web Information Systems and Technologies (WEBIST)</i> , <i>Balisage: The Markup Conference 2014</i> , la <i>XML Conference</i> , la <i>International Conference on Applied Computing</i> , la <i>International Conference on WWW/Internet</i> , o el <i>Congreso Internacional de Ciencias de la Computación</i> .	\$10,000
--	----------

Gastos de viaje. Pasaje de avión (ida y vuelta) para asistir a congreso internacional especializado, en el cual se presentará un artículo de investigación.	\$22,000
Viáticos para asistir a congreso internacional especializado, en el cual se presentará un artículo de investigación.	\$13,000
TOTAL	\$45,000

Segundo año

Inscripción a congreso internacional especializado, en el cual se presentará un artículo de investigación. Posibles congresos: <i>ACM/IFIP/USENIX Middleware Conference</i> , la <i>International Conference on Web Information Systems and Technologies (WEBIST)</i> , <i>Balisage: The Markup Conference 2014</i> , la <i>XML Conference</i> , la <i>International Conference on Applied Computing</i> , la <i>International Conference on WWW/Internet</i> , o el <i>Congreso Internacional de Ciencias de la Computación</i> .	\$10,000
Gastos de viaje. Pasaje de avión (ida y vuelta) para asistir a congreso internacional especializado, en el cual se presentará un artículo de investigación.	\$22,000
Viáticos para asistir a congreso internacional especializado, en el cual se presentará un artículo de investigación.	\$13,000
Inscripción a congreso nacional especializado, en el cual se presentará un artículo de investigación.	\$5,000
Gastos de viaje. Transporte para asistir a congreso nacional especializado, en el cual se presentará un artículo de investigación.	\$4,000
Viáticos para asistir a congreso nacional especializado, en el cual se presentará un artículo de investigación.	\$4,000
TOTAL	\$58,000

Tercer año

Inscripción a congreso internacional especializado, en el cual se presentará un artículo de investigación. Posibles congresos: <i>ACM/IFIP/USENIX Middleware Conference</i> , la <i>International Conference on Web Information Systems and Technologies (WEBIST)</i> , <i>Balisage: The Markup Conference 2014</i> , la <i>XML Conference</i> , la <i>International Conference on Applied Computing</i> , la <i>International Conference on WWW/Internet</i> , o el <i>Congreso Internacional de Ciencias de la Computación</i> .	\$10,000
Gastos de viaje. Pasaje de avión (ida y vuelta) para asistir a congreso internacional especializado, en el cual se presentará un artículo de investigación.	\$22,000
Viáticos para asistir a congreso internacional especializado, en el cual se presentará un artículo de investigación.	\$13,000
TOTAL	\$45,000

10.3 Opciones adicionales de financiamiento

Se planea someter este proyecto para obtener aprobación y financiamiento de instituciones nacionales, tales como el Consejo Nacional de Ciencia y Tecnología (CONACyT); el Programa de Mejoramiento del Profesorado (PROMEP) de la Secretaría de Educación Pública (SEP); o la Secretaría de Ciencia, Tecnología e Innovación del Distrito Federal.

11. Vinculación con los planes y programas de estudio de la División y de la Unidad.

Este proyecto de investigación se vincula de manera directa con algunas de las UEA que se imparten en la Licenciatura en Tecnologías y Sistemas de Información de la UAM Cuajimalpa. Específicamente está vinculado con UEAs tales como Programación Orientada a Objetos; Programación de Web Dinámico; Integración de Sistemas; Laboratorios Temáticos I, II, III, y IV; y Proyectos Terminales I, II, y III.

En el caso de los Laboratorios Temáticos, los cuales son UEAs integradoras de conocimientos y trabajo en equipo principalmente, pueden plantearse sub-proyectos a partir de este proyecto, en los cuales los alumnos puedan desarrollar algunos de los editores de ejemplos interactivos para lenguajes de programación. En el caso de los Proyectos Terminales, también pueden definirse sub-proyectos para ser realizados de manera individual por los alumnos de la Licenciatura en Tecnologías y Sistemas de Información.

12. Vinculación institucional

Hasta el momento no se tiene colaboración externa en este proyecto, pero se buscará establecerla a partir de la difusión de los sitios web que se desarrollarán.