Towards the Development of Text-based Format Converters for Object Representation

Carlos R. Jaimez-González, Betzabet García-Mendoza
Departamento de Tecnologías de la Información
Universidad Autónoma Metropolitana, Unidad Cuajimalpa
Mexico City, Mexico
Email: {cjaimez, bgmendoza}@cua.uam.mx

***Abstract*—Interoperability is the ability to communicate between applications that are written in different programming languages. The communication is established through the exchange of data, which has normally a representation in a standard format. The interoperability problem can be solved by implementing serializers from objects to standard formats, which can then be deserialized in any programming language. This paper presents a project that proposes a solution to address the interoperability problem, which is closely related to the implementation of object serializers. The project proposes to address two important aspects: 1) the representation of objects in standard text formats, independent of the programming language to allow interoperability between applications; and 2) the teaching of the processes involved to represent objects, through the incorporation of these topics in some of the courses of an undergraduate degree in Information Technologies and Systems.**

Keywords—object representation, interoperability, converters, text-based formats, programming languages

I. Introduction

Serialization is the process of converting an object to a form in which it can be stored in a medium such as memory, a file, a database, or a stream for transmission over a network. The main purpose of serialization is to store the state of an object in order to later rebuild it when required. The reverse process is called deserialization.

Many programming languages have serialization support built in as part of the language or through some library; this library or program is called a serializer. In an ideal situation, the serialization and deserialization support included in programming languages would be enough to represent the state of an object that contains data types and basic language constructs; however, in some programming languages, serialization is performed partially, since they do not allow more complex constructions of the language to be represented in the object state, such as pointers or references to objects, collections, enumerations, among others.

Serialization and deserialization are widely used processes when working with distributed systems or applications that require data exchange. An example of this is passing an object to a remote application using a web service or through a remote procedure call. There are also applications that require exchanging objects and are written in different programming languages, for which it is necessary to have a language-independent representation of objects, that is, a standard format to describe serialized objects.

The representation of objects independent of the programming language is not a straightforward process, due to several reasons: 1) objects must be properly restored in terms of their single or multiple inheritance; 2) complex data structures must be adequately reconstructed, mainly those where an object could be referenced multiple times by various pointers or references; 3) collections of objects must be restored in an appropriate way, including for example, lists and dictionaries; 4) the size of numeric data types must be handled appropriately; among others.

Regarding the formats for the representation of serialized objects, they can be divided into two main categories: those that are text-based and binary formats. Some examples of the most widely used text-based formats to represent objects are JavaScript Object Notation (JSON) [1], Extensible Markup Language (XML) [2], and YAML Ain't Markup Language (YAML) [3]. Binary formats are more dependent of their implementation and programming language, so they are not considered standards. The text-based formats are human understandable, which allows for manual inspection and usually facilitates portability between programming languages, although it should be noted that serializing objects to text regularly takes more time and uses more storage space. With respect to their applicability, the three mentioned text-based formats serve the same purpose, which is to provide a common means for the representation of structured data and a mechanism for data exchange independent of the programming language.

Interoperability is precisely the ability to communicate between applications that are written in different programming languages. The communication is established through the exchange of data, which has normally a representation in a standard format. The interoperability problem can be solved by implementing serializers from objects to standard formats, which can then be deserialized in any programming language. It should be noted that this problem has already been addressed in this way by the first author of this paper.

Therefore, the research project presented in this paper proposes a solution that is closely related to the implementation of serializers to address the interoperability problem. This project proposes to address two important aspects: 1) the representation of objects in standard text formats, independent of the programming language to allow interoperability between

2769-5654/22/$31.00 ©2022 IEEE
DOI 10.1109/CSCI58124.2022.00338

applications; and 2) the teaching of the processes involved to represent objects, through the incorporation of these topics in some of the modules of the undergraduate degree in Information Technologies and Systems at our university.

The first aspect is related to the importance of interoperability between applications, in which the most used text-based formats will be considered, the relationship they have with each other to represent simple and complex structures, and the previous work that has been developed. The second aspect is concerned to the teaching about the processes that are carried out in the design and implementation of tools for the representation of objects in one or more text formats; in addition to promoting the use of the tools developed for educational purposes. The second aspect is very relevant for undergraduate students, since it will allow them to know the implications and limitations that exist to achieve interoperability among systems, in addition to being able to use the tools developed to build their own applications.

The rest of the paper is organized as follows. Section II provides the background of the research project proposed; the Web Objects in XML (WOX) framework [4] is introduced; the mechanism used by WOX is described; the serialization and deserialization processes to serialize and deserialize objects are explained; the WOX extensions implemented are presented; and finally, the studies carried out to compare the use of resources and the performance of applications that use JSON, XML and YAML to exchange data are presented. Section III presents the research project proposed; the activities of the project are described; and the schedule of activities is presented. Finally, Section IV provides some conclusions.

II. Background

The background of the research project proposed is the Web Objects in XML (WOX) framework [4], which was created for managing distributed objects and web services. WOX is a framework that has been developed in recent years; initially it only had the functionality for the creation and management of remote objects and web services; it currently has relevant features for the area of distributed systems and, in particular, for interoperability between systems. It should be noted that the WOX framework and its components have been used systematically to support the teaching of several courses of the undergraduate program in Information Technologies and Systems at our university, in particular they have been used in the modules of Object Oriented Programming, Dynamic Web Programming, Systems Integration and Thematic Laboratories.

WOX is a framework that uses the HTTP protocol for communication between clients and servers, uses XML as the format for object representation and keeps the objects available through their own identifiers (URL), inspired by the principles of the Representational State Transfer (REST) architectural style [5]. This section briefly describes some features and functionalities of the WOX framework.

The mechanism used by WOX in a method invocation on a remote object is shown in Figure 1 and the steps performed are described in the following list:

1) The client program calls a method on a remote reference (the way in which the client calls a method on a remote reference is exactly the same as on a local object).

2) The WOX dynamic proxy takes the request, serializes it to XML, and sends it over the network to the WOX server where the remote object is located.

3) The WOX server receives the request and deserializes it to a WOX object.

4) The WOX server loads the object into memory and executes the requested method.

5) The result of the method invocation is returned to the WOX server.

6) The WOX server serializes the result to XML and the actual result or a reference to it is returned to the client program. The result is stored on the server in case a reference to the object has been sent.

7) The WOX dynamic proxy receives the result and deserializes it to the appropriate object, either a real object or a remote reference.

8) The dynamic proxy returns the result to the client program.

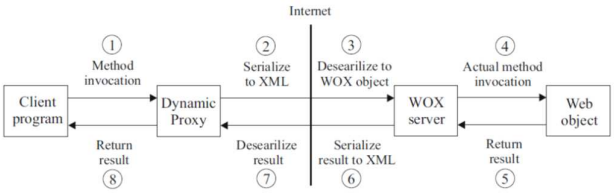

Fig. 1. Invocation of a method on a WOX remote object.

The process of serializing an object to XML and deserializing XML to an object is carried out by the WOX serializers [6, 7], which were initially implemented for the Java and C# programming languages, so it is possible serialize Java objects to XML, deserialize them from XML to C#, and vice versa. WOX serializers are independent libraries, which generate XML representations of objects in a format that is independent of the programming language, and are freely available for download [8]. It should be noted that Microsoft's MSDN magazine published an article [9] about interoperability between Java and .NET applications, in which they used WOX serializers to exchange objects among their applications; the article recommends the use of WOX to integrate heterogeneous applications.

In order to serialize objects to XML, WOX serializers use the following process:

1) The name, type and value of each of the object attributes is obtained. This is done through reflection, which is the ability of a program to observe and optionally modify its high-level structure; by means of this capacity it is possible to access the information of the objects, knowing and/or executing their public attributes and methods, all at runtime. Introspection is also used, which allows obtaining the data type of an object attribute.

2) Once the name and value of each attribute of the object to be serialized is obtained, it is written in an XML document. If the value is not a primitive type but an object, all the attributes of this object must also be represented in the XML document.

Figure 2 shows an example of a class called *Product*, with various attributes of different data types. The code snippet shown in Figure 3 is the XML representation of an object of the *Product* class, once it has been serialized with a WOX serializer. The root of the XML document is an *object* element, with its *type="Product"* attribute, which is the class of the object. The *id* attribute is used to handle references to objects (in this case there is only one object). Each attribute on the object is represented by *field* elements, which have the following attributes: *name* (the name of the attribute in the class), *type* (the WOX data type of the attribute), and *value* (the value of the attribute for that particular object). Primitive or simple types are represented as *field* elements in WOX. The serialization rules of the WOX serializers, as well as a series of examples of serialized objects, can be consulted in [7] and [8].

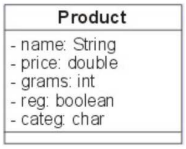

Fig. 2. Product class with various attributes and their data types.

```
<object type="Product" id="0">
    <field name="name" type="string" value="Corn" />
    <field name="price" type="double" value="3.98" />
    <field name="grams" type="int" value="500" />
    <field name="reg" type="boolean" value="true" />
    <field name="categ" type="char" value="\u0041" />
</object>
```

Fig. 3. XML representation of a Product object.

The deserialization process is the opposite of serialization; In this process, the following steps are followed: 1) the information about the object of the XML document is extracted; 2) a class is created with the information obtained from the XML document; 3) an object is created in the corresponding programming language, with the information obtained from the XML document and using the class created in step 2. This process is carried out by WOX deserializers [7].

The WOX framework was extended with the incorporation of asynchronous communication [10], which is used by processes that take a considerable amount of time to be completed by the server. This asynchronous communication allows a client program to continue working without blocking, while the process is running on the server; when the process has completed its execution, the client can retrieve the result of the process. WOX serializers are also present in this scenario, since they take care of the serialization (storage) and deserialization of the results. An asynchronous methods monitor [11] was designed and developed, which allows client programs to monitor the processes that are being executed on the server; the method monitor can be used from a client program or through a web browser.

In order to give continuity to the research work described so far, a research project entitled "Interoperability in object-oriented programming languages" was carried out, which resulted in the inspection and navigation of objects through a web browser [12], since WOX objects are stored on the server through their own URL. With these tools, it is possible to visualize the XML document that represents a particular object or a part of it, by navigating the object with XPath expressions [13]. In the same way, the interface through the web browser also allows to display and execute the methods that belong to an object that has been stored in a WOX server [14]. Additionally, two serializers and deserializers were designed and implemented in the Python [15] and PHP [16] programming languages, and two websites were built with information and examples of their use, where they can also be freely downloaded [17, 18].

The research work described up to this point summarizes the background of the research project proposed in this paper. On the other hand, there are various studies that highlight the importance of exchanging data between applications written in different programming languages, which reside on different devices and/or platforms. This makes it clear that in order to maintain consistent communication between different applications and devices, objects or data in general need to have a representation in a standard format such as JSON, XML or YAML, which are currently the most used text-based formats for data exchange.

The studies carried out in [19, 20, 21] compare the use of resources and the performance of applications that use JSON, XML and YAML to exchange data, in which it is reported that the JSON and YAML formats use less memory than XML to represent objects and that the time to serialize an object to JSON or YAML is less than the time to serialize an object to XML. Object serialization in XML and JSON with different libraries is covered in [22]; an example object is used and serialized with the proposed libraries, measuring the size of the serialized file and the time the serialization process takes; the study concludes that there is no better solution than another and that each library is appropriate in the context in which it was developed; it is also pointed out that both JSON and XML provide interoperability between different programming languages. A review of the process of serializing objects to JSON format and vice versa is described in [23]; it is stated that the JSON format is more efficient than the XML format, due to the little space it occupies compared to XML, and the serialization that takes less time.

The research carried out in [24] performs an analysis of XML and JSON in the context of decision-making computational systems; it concludes that both technologies have their own advantages and disadvantages, and suggests that for applications that use simple data structures JSON is more appropriate than XML, but for those applications with complex structures XML is indicated. In the comparative studies carried out in [25, 26], the results indicate, as in the rest of the studies, that the JSON and YAML formats are more effective than XML in terms of size and serialization time; however, it is noted that for some applications that require sending complex data structures, the XML format offers better support for representing data that cannot be represented in JASON or YAML. Additionally, it is pointed out that as part of the future work of these investigations, the development of converters from XML

representations to JSON, and vice versa, is proposed; XML to YAML, and vice versa; JSON to YAML, and vice versa.

III. Proposed project

The aim of the proposed project is to design and implement interoperable tools to represent objects in standard text formats, which can be used in the teaching and learning process of the students of the undergraduate program in Information Technologies and Systems at our university. Some of the specific objetives are the following: 1) develop a converter from the XML format used by WOX objects to the JSON format and vice versa; 2) develop a converter from the XML format used by WOX objects to the YAML format and vice versa; 3) perform functionality, interoperability and performance tests on the developed tools; 4) develop websites with documentation, examples and laboratory practices.

The activities proposed for this research project are described in the following paragraphs, taking into account that the purpose is to address two important aspects: 1) the representation of objects in standard text formats, independent of the programming language to allow interoperability between applications; and 2) the teaching of the processes involved to represent objects in some of the modules of an undergraduate degree in Information Technologies and Systems. Additionally, having reviewed the previous research work on the WOX framework and the relevant literature related to the representation of objects in standard text formats, this research project will aim to provide representations of objects in the standard text formats most widely used (JSON and YAML), which will allow interoperability of the existing WOX framework with applications that currently use the JSON and YAML formats to exchange data. The proposal is to create a series of converters between the JSON and YAML text formats and the object representation used by WOX, so that applications built with WOX can interoperate with other applications. As part of the research project, the limitations of these conversions will be studied in terms of the representation of simple and complex data structures (lists, dictionaries), as well as pointers or references. The following stages are proposed for the project:

1. Development of a WOX-JSON converter. In this stage, the corresponding modules will be created for the development of a converter from the XML format used by the WOX objects to the JSON format and vice versa. This stage includes an analysis of the implications and limitations of this conversion between formats.

2. Development of a WOX-YAML converter. In this stage, the corresponding modules will be created for the development of a converter from the XML format used by WOX objects to the YAML format and vice versa. This stage includes an analysis of the implications and limitations of this conversion between formats.

3. Functionality, interoperability and performance tests. In this stage, the different functionality, interoperability and performance tests will be carried out on each of the developed tools: WOX-JSON converter and WOX-YAML converter.

4. Website development with documentation, examples and laboratory practices. In this stage, websites will be created that will have the following information about each of the developed tools: installation instructions, description of each of the tool's modules, information about the tool's scope and limitations, examples of use, interoperable tools. In this stage, laboratory practices will also be created as didactic material for the teaching and use of the developed tools. The generated material will be used in various courses of our undergaduate program.

Regarding the schedule of activities, the research project proposed has an expected duration of three years.

In the first year, the following activities are planned: 1) study the tools to serialize and deserialize objects to and from JSON and YAML text-based formats, in object-oriented programming languages; 2) literature review on serialization and deserialization with JSON and YAML text-based formats; 3) design and implementation of an object converter in WOX format to a representation in JSON format.

In the second year, the following activities are considered: 1) tests of functionality, interoperability and performance of the WOX-JSON converter; 2) design and implementation of an object converter in JSON format to a representation in WOX format; 3) functionality, interoperability and performance tests of the JSON-WOX converter; 4) design and implementation of an object converter from WOX format to a representation in YAML format; 5) functionality, interoperability and performance tests of the WOX-YAML converter; 6) design and implementation of an object converter in YAML format to a representation in WOX format.

Finally, in the third year, the following activities are planned: 1) functionality, interoperability and performance tests of the YAML-WOX converter; 2) development of websites with documentation and examples for the tools developed, which will include the tools for free download; 3) creation of didactic material for teaching and use of the developed tools.

The research project proposed is directly linked to some of the courses taught in the undergraduate program of Information Technologies and Systems at our university. It is specifically linked with courses such as Object Oriented Programming; Dynamic Web Programming; Systems integration; Distributed Systems; Thematic Laboratories I, II, III, and IV; and Final Projects I, II, and III. In the case of the Thematic Laboratories, which are mainly courses that integrate knowledge aquired by students previously and the work is donde in teas, subprojects can be proposed based on this research project. In the case of Final Projects, subprojects can also be defined to be carried out individually by the students of our undergraduate program.

IV. Conclusions

This paper provided the background of the research project proposed; the WOX framework [4] was introduced; the mechanism used by WOX was described; the serialization and deserialization processes to serialize and deserialize objects were explained; the WOX extensions implemented were presented; and finally, the studies carried out to compare the use of resources and the performance of applications that use JSON, XML and YAML to exchange data were presented. It was also presented the research project proposed; the activities of the project were described; and the schedule of activities was presented. Further work is needed to carry out the activities planned in the proposed research project.

REFERENCES

[1] Introducing JSON. http://www.json.org/

[2] T. Bray, J. Paoli, C. Sperberg-McQueen, E. Maler, F. Yergeau, "The Extensible Markup Language (XML) 1.0", 2013, available at: https://www.w3.org/TR/xml/

[3] YAML. https://yaml.org/

[4] C. R. Jaimez-González, S. M. Lucas, "Implementing a State-Based Application Using Web Objects in XML", in: Meersman R., Tari Z. (eds) On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS. OTM 2007. Lecture Notes in Computer Science, vol. 4803, pp. 577-594, 2007, Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-76848-7_40

[5] R. Fielding, "Architectural Styles and Design of Network-Based Software Architectures", PhD thesis, USA, 2000.

[6] C. R. Jaimez-González, S. M. Lucas, "Interoperability of Java and C# with Web Objects in XML", in Proceedings of the International Conference e-Society (ES 2011), pp. 518-522, Avila, Spain, March 2011.

[7] C. R. Jaimez-González, S. M. Lucas, E. López-Ornelas, "Easy XML Serialization of C# and Java Objects", Balisage: The Markup Conference 2011, Montréal, Canada, August 2011, in Proceedings of Balisage: The Markup Conference 2011, Balisage Series on Markup Technologies, vol. 7. https://doi.org/10.4242/BalisageVol7.Jaimez01

[8] C. R. Jaimez-González, S. M. Lucas, "Web Objects in XML (WOX): Efficient and easy XML serialization of Java and C# objects", http://woxserializer.sourceforge.net/

[9] I. Khan, "Interoperability: Runtime Data Sharing Through an Enterprise Distributed Cache", MSDN Magazine, vol. 25, No. 10, October 2010, http://msdn.microsoft.com/en-us/magazine/gg232763.aspx

[10] C. R. Jaimez-González, S. M. Lucas, "Asynchronous Method Invocations Using HTTP Polling and HTTP Streaming", in Proceedings of the International Conference on Applied Computing 2011 (AC 2011), pp. 536-540, Rio de Janeiro, Brazil, November 2011.

[11] C. R. Jaimez-González, W. A. Luna-Ramírez, S. M. Lucas, "A Web Tool for Monitoring HTTP Asynchronous Method Invocations", in Proceedings of the IEEE International Conference for Internet Technology and Secured Transactions, pp. 127-132, London, December 2012, https://ieeexplore.ieee.org/document/6470883

[12] C. R. Jaimez-González, "A Simple Web Interface for Inspecting, Navigating, and Invoking Methods on Java and C# Objects", Research in Computing Science: Advances in Computing Science, vol. 81, pp. 133-142, 2014, https://www.rcs.cic.ipn.mx/2014_81/RCS_81_2014.pdf

[13] J. Robie, D. Chamberlin, M. Dyck, J. Snelson, "XML Path Language (XPath)", W3C Recommendation, 2014.

[14] J. M. Hernández-Salinas, C. R., Jaimez-González, "Herramienta Web para Almacenar y Visualizar Objetos Distribuidos", Research in Computing Science, vol. 125, pp. 63-74, 2016.

[15] A. I. Rodríguez-Martínez, C. R. Jaimez-González, "Serializador de Objetos a XML en el Lenguaje de Programación Python", Avances de Ingeniería Electrónica 2013, pp. 444-451, 2013.

[16] L. Hernández-Piña, C. R. Jaimez-González, "Serialización de Objetos PHP a XML", Research in Computing Science, vol. 125, pp. 87-95, 2016.

[17] C. R. Jaimez-González, A. I, Rodríguez-Martínez, "Web Objects in XML in Python (PyWOX): Serializador de Objetos a XML en el lenguaje de programación Python", 2014, http://pywoxserializer.sourceforge.net/

[18] C. R. Jaimez-González, L. Hernández-Piña, "Web Objects in XML - PHP (PHPWOX): Serialización XML de objetos en PHP y viceversa", 2014, http://phpwoxserializer.sourceforge.net/

[19] N. Nurseitov, M. Paulson, R. Reynolds, C. Izurieta, "Comparison of JSON and XML Data Interchange Formats: A Case Study", in Proceedings of the 22nd International Conference on Computer Applications in Industry and Engineering (CAINE 2009), November 2009, San Francisco, California, USA.

[20] M. Ericksson, V. Hallberg, "Comparison between JSON and YAML for data serialization", BSc Thesis, Sweden, 2011.

[21] G. Goyal, K. Singh, K. Ramkumar, "A detailed analysis of data consistency concepts in data exchange formats (JSON & XML)", in Proceedings of the International Conference on Computing, Communication and Automation (ICCCA), May 2017, https://doi.org/10.1109/CCAA.2017.8229774

[22] K. Maeda, "Performance evaluation of object serialization libraries in XML, JSON and binary formats", in Proceedings of the Second International Conference on Digital Information and Communication Technology and its Applications (DICTAP), May 2012, Bangkok, Thailand, https://ieeexplore.ieee.org/document/6215346

[23] J. Mora-Castillo, "Serialización/deserialización de objetos y transmisión de datos con JSON: una revisión de la literatura", Tecnología en Marcha, vol. 29, No. 1, pp. 118-125, 2015, https://doi.org/10.18845/tm.v29i1.2544

[24] Z. Haq, G. Khan, T. Hussain, "A Comprehensive analysis of XML and JSON web technologies", New Developments in Circuits, Systems, Signal Processing, Communications and Computers. pp. 102-109, 2015.

[25] A. Breje, R. Gyorodi, C. Gyorodi, D. Zmaranda, G. Pecherle, "Comparative Study of Data Sending Methods for XML and JSON Models", International Journal of Advanced Computer Science and Applications (IJACSA), vol. 9, No. 12, pp. 198-204, 2018, https://doi.org/10.14569/IJACSA.2018.091229

[26] K. Grochowski, M. Breiter, R. Nowak, "Serialization in Object-Oriented Programming Languages", Introduction to Data Science and Machine Learning, IntechOpen, pp. 1-18, 2019, http://dx.doi.org/10.5772/intechopen.86917