Reality and Reflections

Position Paper
For W3C Workshop on Rules Languages for Interoperability

Author

Waleed K. Abdulla   waleed_abdulla@xrules.org

1.1 Perspective

Our world is a complex creation beyond any imagination, and the rules that govern it are as complex or more. Working with such complexity is quite a challenge even for the magnificent human brain, let alone computer software.

Our brain's solution to deal with this complexity was to create simplified reflections of reality in our minds with just enough details to allow us to process and understand our world. For example, I know that when I throw a stone in the air, it will eventually fall down and, probably, hit something. That's enough information for me to conclude that throwing stones is not a good idea. I don't need to know more facts about the stone such as its mass and structure, and I don't need to be aware of the laws of physics that control the stone movement in the air. In software, we use a similar approach: we create XML representations of the real world with just enough details for the task at hand. For example, if we're building a medical information system, we represent a person by an XML document that contains the person's medical history. And if we're building mortgage processing software, we represent the same person with an XML document that contains his/her financial information, but no medical details. Each XML document is a different reflection of the real person.

When designing a rules language, the first decision to make is weather to target the facts and rules of the real world, or the facts and rules of an XML representation of the real world. I'll refer to the first as Semantic Rules, and to the second as XML Rules. To show the difference, here are two examples:

1. Example of a semantic rule:

```
If     ?X is father of ?A
And    ?X is father of ?B
Then   ?A is sibling of ?B
```

2. Example of an XML rule (syntax in XRules[1]):

```
<xr:calculate target="ItemTotal" value="UnitPrice * Quantity" />
```

Both rules can produce new knowledge from existing facts, but they differ in the type of facts they work with, and each has syntax suitable for its intended use.

1.2 Two Domains, Two Rules Languages

It's my intention to make the point that one rules language cannot fit all. We need two: One for semantic rules, and one for XML rules. My interest is in the area of XML rules, and, therefore, and from experience designing the XRules language, I suggest the following requirements for an XML rules language.

1. Simple syntax for human readability and maintainability. Especially, a syntax that makes it as simple as possible to handle a big number of rules in one document.
2. Uses XPath to identify XML nodes since it's the best language for that purpose.
3. Uses XPath for some constraints and expressions, as it's a powerful language and fits right into this role. Then, complements XPath with both extension functions and XML syntax to express the types of rules that XPath can't express.
4. Supports and utilizes XML Schema data types.
5. Provides a wide range of extensibility options.
6. Makes it easy for other tools and languages (such as XSLT) to tap into the new knowledge generated by the inference and calculations of the rules.

1.3 Position

1. It's my hope that this workshop would result in the formation of two workgroups:
    a. A workgroup to create an XML rules language and ensure that it works well with existing technologies; specifically: XPath, XML Schema, XSLT, and XForms.

    b. A workgroup to create a semantic rules language that is aligned with RDF principles. This language takes input knowledge represented in RDF and produces knowledge that can be represented in RDF.
2. I'm proposing XRules (www.xrules.org) as a starting point for an XML rules language for the following reasons:
    a. Satisfies all the requirements listed earlier in the previous section.

b. Adapts RDF principles (though not its syntax, and there are reasons) in the sense that it treats every XML node as a resource and allows attaching properties to these resources.

```
<xr:bind target="SomeNode" >
    <xr:property name="isValid" dvalue="target() &lt; MaxAllowed" />
</xr:bind>
```

c. Not limited to constraints, but can generate new knowledge within its limits in the XML domain. This is done using the <calculate> rule as shown earlier in this paper.

1.4 References

[1] XRules: an XML rules language.
http://www.xrules.org

About the Author

Waleed K. Abdulla, M.Sc. Computer Engineering

Senior software developer and architect interested in Web services and rules languages. Involved in two vertical standards organizations, and an editor of the STAR Web services specifications used in the automotive and motorcycle industries. And, the author of the XRules XML language for XML rules.

Email: waleed_abdulla@xrules.org