

# The Mathematical Semantic Web

Massimo Marchiori<sup>1,2</sup>

<sup>1</sup> The World Wide Web Consortium, MIT, Cambridge, USA

<sup>2</sup> Department of Computer Science, University of Venice, Italy  
`massimo@w3.org`

**Abstract.** How can Mathematics and the Semantic Web effectively join? In this paper we provide an account of the key standard technologies that can foster the integration of Mathematical representation into the Semantic Web.

## 1 Introduction

Modern approaches to Mathematics on the web, nowadays, have in common with XML the well-known dualism between semantics and presentation: the media should be detached from the meaning ([1]). With the advent of the Semantic Web, however, the potential of this dualism seems to raise even higher, as that “semantics” part can possibly be more than itself, be part of a whole, be a single brick in a big building, the World Wide Web. For Math, this means that not only the Semantic Structure and Presentation Structure play their obvious classical role, but new possibilities come from the integration of various mathematical sources, allowing more powerful global search capabilities, but also associated metadata (context), possibility of computable math (algebraic manipulation / calculation) and so on. In order to achieve this, however, there is the need for integration: integration of the mathematical knowledge expertise/techniques, with the mainstream technologies that constitute the Semantic Web. Here, we focus on those Web technologies that could be potentially low-hanging fruits: RDF, OWL, XML-Schema, XML-Query and Functions and Operators.

## 2 RDF

When one talks about the Semantic Web, the first thing that comes to mind is RDF, the milestone on which the classic Semantic Web architecture lies. RDF, or Resource Description Framework (cf. [9, 8, 3]), is essentially an enriched entity-relationship model, designed to encode data (and, so to say, “meaning”) in the World Wide Web.

So, while there are already XML representations for Mathematics on the Web (the most prominent example being MathML), in order to fully enter the Semantic Web bandwagon, there is the need to go beyond a normal XML representation, and try instead to link a bridge towards RDF-land.

This doesn't mean that existing XML dialects for Mathematics have to be abandoned, or that future versions and improvements have to be based on RDF. Rather, it means that a suitable representation to and from RDF should be given, so that applications dealing with Mathematical knowledge can export their knowledge to the Semantic Web (using an RDF representation), and conversely, that such an RDF representation can be mapped back<sup>3</sup> An example of such approach is for example given by the P3P standard ([11]), which has its own XML syntax, and a corresponding alternate RDF representation ([12]).

Many straightforward representations of Math knowledge to RDF are possible, given that RDF expresses, as said, a generic graph structure (entity-relationship) that can be made to fit any generic structure. In particular, it is rather evident that mathematical formulas, for their same nature of being usually serializable in text form, fit rather well in the RDF graph model. For example, the representation of a function  $p(2, 6)$  expressed in MathML as

```
<apply>
  <csymbol encoding="text"
    definitionURL="http://www.mathsw.org/scalarplus">
    p
  </csymbol>
  <cn> 2 </cn>
  <cn> 6 </cn>
</apply>
```

could be given an RDF representation like:

```
:_1 <http://www.w3.org/TR/MathML2#apply> :_2
:_1 <http://www.w3.org/TR/MathML2#csymbol> "p"
:_1 <http://www.w3.org/TR/MathML2#definitionURL>
      <http://www.mathsw.org/scalarplus>
:_1 <http://www.w3.org/TR/MathML2#encoding> "text"
:_2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#:_1> "2"
:_2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#:_2> "6"
```

Here, we have used the so-called "triple" representation of an RDF graph, where each row represents a triple

*entity1 relationship entity2*

(see e.g. [8] for more details).

Much like *proof nets*, the RDF graph structure nicely leads to encoding mathematical structures (it allows for example easy (and effective) processing of bound variables). For example, the function  $\lambda x. x + x$  could be encoded in MathML with an explicit bound variable like this:

<sup>3</sup> Although, what is really needed is the mapping from the XML dialect to RDF, as the reverse mapping could be (depending on the specific application) of less use, or even impossible in some cases (when the domain of discourse is enlarged in RDF).

```

<lambda>
  <bvar><ci> x </ci></bvar>
  <apply>
    <plus/>
    <ci> x </ci>
    <ci> x </ci>
  </apply>

```

RDF, on the other hand, allows a neater representation of the bound variable:

```

:_1 <http://www.w3.org/TR/MathML2#apply> :_3
:_1 <http://www.w3.org/TR/MathML2#lambda> :_2
:_3 <http://www.w3.org/TR/MathML2#definitionURL>
      <http://www.w3.org/TR/MathML2#plus>
:_3 <http://www.w3.org/1999/02/22-rdf-syntax-ns#:_1> :_2
:_3 <http://www.w3.org/1999/02/22-rdf-syntax-ns#:_2> :_2

```

The technical details of any such mapping to RDF are unnecessary here, as first they are rather obvious<sup>4</sup>, and second, many equivalent choices are possible. What is important, though, is to note that the mathematical structure of objects can/should be given an RDF formulation, so to foster integration between the Math world and the Semantic Web. With an RDF representation for Math, we can use all the tools developed/in development for the Semantic Web, like for example search, inference, annotation, and so on. So, general-purpose tools can be reused to operate with mathematical objects, and no special-purpose tools have to be developed. Most importantly, putting mathematics inside RDF allows a nice and smooth integration with all various other sources of data: in other words, proper integration with the Semantic Web via RDF allows for a seamless definition of *contexts*, and linkage to various other metadata related to the math object. This allows to go way beyond the usual way of allowing metadata in XML dialects, i.e., using specific attributes where to fit more information. So, RDF really allows making Mathematics integrated in the World Wide Web.

### 3 OWL

Mathematical objects "per se" can be very difficult objects to categorize, if appropriate metadata is not provided. This is a general problem of many kinds of data present on the Web, so much that W3C has addressed this issue by chartering a new effort for the production of a standard devoted to Ontologies for the Web: the Web Ontology Language (also called "OWL"), see for instance [7]. OWL allows data to be attached with ontological meaning: in other words, it allows powerful mechanisms to categorize data on the web into specific classes and subclasses, therefore greatly facilitating the proper handling/search/query of web data.

<sup>4</sup> Although, note that even from the simple examples shown here, some subtleties are present, and we have been for example sloppy in the treatment of datatypes, as we will hint at later.

What this means from a Math viewpoint is that mathematical objects, that can by their nature be rather dense and not easily categorizable at a first inspection, can instead be nicely associated to some specific categories. Such mathematical ontologies can provide a formidable help for the task of searching for mathematics, and for the task of issuing similarity searches, avoiding the intrinsic difficulties of extracting this information from the semantics/structure of the object (which might obviously be just plain impossible). OWL is flexible enough to allow for complete mathematical hierarchies to be formed, and also to allow for alternate hierarchies, so that not just one categorization has to be used, but many can happily coexist.

## 4 XML Schema

Types are an essential component in mathematics, and it comes by no surprise that even when attempting to formalize mathematics on the Web, types have entered the battlefield. Numeric types like integer, real, complex and so on have been defined in MathML, while the OpenMath Consortium has also experimentally tried to give more complex typing facilities using the Extended Calculus of Constructors ([6]). On the other hand, type systems have also been introduced in the Web, and more specifically, one type system is now the world standard: XML Schema ([13, 4]). It is tempting therefore to analyze whether there are differences, and where convergence is possible here.

The first big distinction that has to be done is between what XML-Schema calls *primitive* datatypes. In a sense, primitive types are the basic building blocks, from which more complex types can then be created. XML-Schema primitive datatypes much correspond, in spirit, to MathML types. However, we can see that there are differences, as MathML focusses on numeric types, and neglects other specific XML datatypes. Now, here convergence is not only possible (primitive datatypes can be merged with no problems), but strongly needed for the future integration of mathematics on the web. In fact, RDF is already adopting XML-Schema primitive datatypes in the Semantic Web view of the world, which makes even more urgent for mathematical knowledge to align. Note that numeric datatypes currently not present in XML Schema can be added using so-called user defined types, so the merge of the two needs (XML-Schema primitive types, and “primitive types” of the mathematical community) can fit within XML-Schema.

The other part of the XML-Schema type system is more powerful, but here, the two types of schemas (XML, and math) diverge in a sense, as the first one deals with structural constraints, while the second deals more with signature definition. It seems like XML-Schema type system could accommodate some math type systems, but the interaction here is deeper, as some desired type system for math (like the ECC mentioned above) are more fine-grained than XML-Schema’s type system in some circumstances, so convergence here should be explored further.

## 5 Functions and Operators

Types are useful for a variety of task, last but not least, validation. But another very useful functionality they provide is just to provide some signatures, so that *executable specification* of the corresponding function/operator can be safely activated, even when no full or accurate type system is present. And one of the possible cool functionalities that math functions and operators could have is just this: to be *computable*, in the sense that they could, somehow, be activated (evaluated), and the appropriate result returned. There are of course various models that would allow to have executable mathematics on the Web, from the central server model (where requests are sent, for example using a Web Service architecture, or classic HTTP PUT/POST methods), to the local server model, where applets could be for example downloaded, and executed in situ. What matters most for our discussion is, besides the need for a uniform way (or, uniform multiple ways) to perform such activation, the necessity to have uniformity.

The first key to successfully represent mathematics on the web is a single acronym: URIs. As Web Architecture dictates, URIs should be used to denote every relevant thing in the Web. Restated, in the mathematical environment every function/operators/relations and so on, should be given an appropriate URI, that is to say, an appropriate name in Web space. It is for this reason that the current MathML standard ([5]) allows for new symbols to be classified using a URI: this allows for much better search on the web, as well as, for example, the possibility of automatic execution and/or algebraic manipulation between different sources of math. Without URIs, we have to stick to local names, and therefore we have names that are useful only within a single document, or within a single application.

But, if URIs provides us with good “global addresses” along the Web, the other crucial factor for their success is that, as far as possible, people don’t introduce new URIs that always denote the same thing. If this happens, the utility of the URI boils down to just a single identifier, making the web just an isolated bunch of points, with no connections.

In our case, this means that it is of little use to use URIs for functions and operators, if each of us creates such URIs from scratch, without reusing existing URIs. There is the need for some appropriate standardization of common operators, so that people can reuse it, search tools can link different occurrences in different parts of the web, execution of a vast number of functions can be performed. [10] is an example of a first catalog of functions and operators that are going to be standardized, for uniform usage over the Web. Similar efforts could be undertaken to extend such collection to more mathematical functions, where first-order functions and operators could be considered first. In any case, it is obviously important that uniformity prevails, and that there is convergence between the XML/database world, and the mathematical environment on the web (think for example at the collection in [14]).

## 6 XML Query

From what seen previously, it would seem like RDF and OWL already provide all what we need for a proper integration of Mathematical knowledge into the Semantic Web. But of course, the complete picture is far from being that simple. While, certainly, RDF and OWL can do a lot to foster reuse of Semantic Web tools with mathematical objects, there are still some drawbacks that one has to take into proper account.

The primary of such drawback is the complexity of the tools. Simply having an RDF representation doesn't solve our needs, as then we need to use some tools that allow us to query and manipulate the mathematical objects. Therefore, another aspect has to be considered: the tradeoff between the complexity of such tools, and our needs.

As far as the functional requirements are concerned, we envisage that, to start with, the primary applications in the use of mathematical knowledge on the World Wide Web in the large will be forms of *simple search*, possibly *extraction*, and easy *manipulation*.

Moreover, always talking in the large, the other factor to consider here is the scalability of the solution. The RDF representation make mathematical objects migrate to a more general space, the RDF graph, at a price: it can complicates quite a lot the structure, as compared to the original XML representation. When talking about search/extraction/manipulation in the large, therefore, we risk that staying in the RDF graph for such operations will not scale so easily, leading to *failure of responsiveness*.

The possible way out to this is not to give away the XML representation when we need it, as in this case. Such representation can in fact be traversed much more efficiently than an RDF representation, and as such, specific tools that operate on XML can be used. In particular, XML Query (also known as XQuery, cf. [2]), the future world standard for querying XML data, could be profitably reused. XML Query itself operates on a variant of the relational model, adapted to the specific XML data model, and has been designed with all the needs that database vendors seek out in current SQL-like systems, like, in primis, speed and scalability. It is an interesting exercise to see, given a certain XML representation for math (like, say, MathML), what is the power of queries and related operations that XQuery can perform<sup>5</sup>. For sure, XQuery fits perfectly in the use cases of simple search and extraction, which makes it a good candidate for query/search application of XML Mathematical knowledge in the Web (so, in the large). In fact, for more sophisticated applications, one could imagine the possibility of a *hybrid* system, where the mathematical objects are just seen in the RDF graph as XML literals. This still allows the mathematical objects to interact with the semantic web, to be given context, annotations and so on. What we would lose is easy access to the internal structure of the objects, within the RDF world. For that access, we could activate an XQuery processor, that could quickly find the result we need. And in fact, such XML literals could always be exploded "on demand" to their RDF

---

<sup>5</sup> Indeed, XQuery, although *Turing complete*, is not *functionally complete*.

structure, like in lazy functional programming, when really needed. Therefore, the hybrid model would work well in all those situation where full RDF access to the internal structure of the mathematical objects is seldom needed.

## 7 Success?

Having seen the possible synergies, the final questions to ask is: can this integration be performed with success?

The first factor to consider is merely technical, and has to do with the problem of *vastity of domain*.

Mathematics is a vast field. This means that potentially, Semantic Web application like search, similarity search, manipulation, inference, can perform badly if not properly assisted with precise Semantic Web annotations. There is probably the need to identify some critical subsets of mathematical knowledge that can benefit more from semantical structuring (pretty much the same kind of selection that MathML somehow had to do when facing the creation of an XML math dialect ); this seems even more important when we go to the higher layers (computability, typing, algebraic manipulation). Constructive mathematics here obviously will play a much more relevant role.

The second factor to consider is societal, and concerns the *cost/benefit vicious circle*.

Information encoded using the semantic web has a higher cost, like any form of evolved semantic encoding, than a merely syntactic formatting. So, the cost to put information on the Web gets higher. In order to make people accept this extra cost, the cost/ratio benefit must stay low. But crucially, in the semantic web the benefits usually depends on adoption: the more, the better. Therefore, we have this vicious circle: people will not be very likely to contribute to a mathematical semantic web, unless critical mass is reached. It is this vicious circle that might slow down too much progress in the field, if not carefully weighted.

## References

1. A.Asperti, L.Padovani, C.S.Coen, I.Schena, *XML, Stylesheets and the rem-mathematization of Formal Content*, Proceedings of “Extreme Markup Languages 2001 Conference”, 2001. Available at <http://www.cs.unibo.it/helm/extreme2001.ps.gz>.
2. S.Boag *et al.* (Eds.), *XQuery 1.0: An XML Query Language*, World Wide Web Consortium Working Draft, November 2002. Latest version at <http://www.w3.org/TR/xquery>.
3. D.Beckett (Ed.), *RDF/XML Syntax Specification (Revised)*, World Wide Web Consortium Working Draft, November 2002. Latest version at <http://www.w3.org/TR/rdf-syntax-grammar>.
4. P.V.Biron and A.Malhotra (Eds.), *XML Schema Part 2: Datatypes*, World Wide Web Consortium Recommendation, 2001. Available at <http://www.w3.org/TR/xmlschema-2/>.

5. D.Carlisle, P.Ion, R.Milner, N.Poppelier, *Mathematical Markup Language (MathML) Version 2.0*, World Wide Web Consortium Recommendation, 2001. Available at <http://www.w3.org/TR/MathML2>.
6. O.Caprotti and A.M.Cohen, *A Type System for OpenMath*, The OpenMath Consortium, 1999. Available at <http://monet.nag.co.uk/cocoon/openmath/standard/ecc.pdf>.
7. M.Dean *et al* (Eds.), *Web Ontology Language (OWL) Reference Version 1.0*. World Wide Web Consortium Working Draft, November 2002. Latest version at <http://www.w3.org/TR/owl-ref>.
8. P.Hayes (Ed.), *RDF Semantics*, World Wide Web Consortium Working Draft, November 2002. Latest version at <http://www.w3.org/TR/rdf-mt/>.
9. O.Lassila and R.Swick (Eds.), *Resource Description Framework (RDF) Model and Syntax Specification*, World Wide Web Consortium Recommendation, 1999. Available at <http://www.w3.org/TR/REC-rdf-syntax>.
10. A.Malhotra *et al*. (Eds.), *XQuery 1.0 and XPath 2.0 Functions and Operators*, World Wide Web Consortium Working Draft, November 2002. Latest version at <http://www.w3.org/TR/xquery-operators/>.
11. M.Marchiori (Ed.), *The Platform for Privacy Preferences 1.0 (P3P1.0) Specification*, World Wide Web Consortium Recommendation, 2002. Available at <http://www.w3.org/TR/P3P>.
12. B.McBride *et al.*, *An RDF Schema for P3P*, World Wide Web Consortium Note. Latest version at <http://www.w3.org/TR/p3p-rdfschema/>.
13. H.Thompson *et al.* (Eds.), *XML Schema Part 1: Structures*, World Wide Web Consortium Recommendation, 2001. Available at <http://www.w3.org/TR/xmlschema-1/>.
14. Wolfram Research, *Mathematical Functions*. Available at <http://functions.wolfram.com/>.