Like most things in computer science, we start with an acronym: one that has achieved record buzzword status over the past few years: XML, which stands for eXtensible Markup Language. What does this particular trend do for computer simulation? Maybe we should back up a bit and first talk about what XML is, and what is driving it, and then we can tie it to the needs of simulation. XML is a “markup language” which defines knowledge, data, and metadata about an application domain. For example, here is some XML:

```
<customers state="Florida">
  <customer id="Fred" email="fred@myplace.com"/>
  <customer id="Mary" email="mary@place2.univ.edu"/>
</customers>
```

This is a list of 2 customers from Florida. An XML file is a flat file that contains data, which must be processed in some fashion to yield output, usually to a web page. The history of XML began with SGML (Standard Generalized Markup Language), which was developed to assist in typesetting and general documentation. Things began to heat up when the web developed, and instead of SGML taking charge, HTML (HyperText Markup Language) was used, instead, to create the content for web pages. You may be familiar with HTML if you have ever edited a web page, but if you have not, you are in good company. If you wish to venture into the unknown, open your web browser and view the content for the document. In Internet Explorer, you do view→source and in Netscape, it is view→page-source. On many pages, this will show you HTML, but on some pages you will see XML. The web is increasingly turning over its content to XML for several reasons. A key reason is that XML stresses a separation of content from presentation. Content is what is defined above in the customer example, and presentation how that content is presented to you on the web page. HTML is primarily a presentation language, so if you specify something in HTML, you are defining how it will look in a web page. Take a look the figure on this page. The large text labeled “Simulation” is drawn using Bezier curves, which are encoded as `<path>` elements in an XML language for 2D drawing called SVG (Scalable Vector Graphics). I used the program SodiPodi, which is an open source package for 2D SVG editing, to create this. I took some of the SVG content and glued it underneath of the “Simulation”
text. You can see a minor resemblance between SVG and the customer XML that I first showed you: the angle brackets are used to define elements, and then each element may have one or more attributes whose values are expressed within quotation marks.

Well, what does this have to do with simulation? The field of simulation is broad, but let’s sub-define it into the following areas: Modeling, Execution, and Analysis. Modeling is the process of building models; execution reflects a sequential or parallel program execution, and analysis is where we employ statistical and otherwise mathematical tools to test, verify, and validate our models. Models can be specified using XML. One has to create a kind of grammar for a model language (termed a schema) and then, as with all types of XML, one would then need to translate the XML into a presentation. Here is an example. Let’s suppose you have a queuing model. First, you would need to define an XML language (i.e., called an XML application) for the queuing model. It might contain an element with an attribute, like `<server mean="5.6">`. Then you would need to translate this into a presentation, perhaps in SVG, and into code that, when executed, would simulate this model. Your XML might be spread over several servers or clients. If you then turn your XML into a mathematical structure in MathML (the mathematics markup language), you might use an analysis program that is capable of processing MathML to yield statistical results.

XML is no panacea, to be sure, but it does offer an opportunity to create building blocks, which may be processed using XML style sheets, and integrated into the web. Most companies have adopted XML in their business practices, and there are a large number of scientific and engineering disciplines that have adopted XML for defining content. A key thing to remember with XML is that it is not just for documents, at least in the way you may think about the word “document.” XML, and the suite of tools you can find at www.w3.org, will help you organize knowledge about simulation models and their analyses. Unlike the old web, where we had lots of HTML presentation content, we are evolving into what the web founders term a semantic web. For modeling and simulation, this is just the ticket.