

# Visualizing Business Rules

## *SVG in Action*



In the beginning, there was data. Pretty quickly, there got to be so much data that necessity dictated the invention of databases. While the term database generally brings to mind a computer and rows and rows of well-sorted information, both numeric and textual, the concept of databases has been around for literally thousands of years. Libraries full of data existed long before electricity was harnessed to run computers.

Without doing a lot of historical research, it is safe to say that an early form of database was the spreadsheet—one record per row. Although there is plenty of human influence on the spreadsheet (what “object” should be tracked, what information should be collected about each tracked object, how data should be displayed), for a business or a library, the power of the data contained in the spreadsheet lies not in the rows of data but in how each record relates to other records—the business rules.

### **BEARING RESPONSIBILITY**

Entire books could be written about where the responsibility lies for developing business rules, but clearly a company’s leaders bear much of that responsibility. Very likely long before there were computer databases, but certainly since the time that computers have become so ingrained in business, a large communications gap has existed between the leaders who have the business rules and the database administrators who know how to use the tools of the database. This situation is analogous to publishing a newspa-

per or a book. The printers (the database administrators) know how to print the words, but they don’t know the words to print. The writers (the company leaders) know what to say, but have no clue as to how to physically print the paper.

Regardless of where in the business structure you are located, it is important to realize that the data managers (with their technical skills) and the leaders (with their business rules and vision) are a team. There is no right answer or correct path. It is a process of constant iteration and improvement. This article describes a set of tools that the data managers can use to visualize data so that the leaders can make better business decisions. If you are a data manager, the goal is to provide you with some practical ideas for tool building. If you are a leader developing business rules for data relationships, the goal is to provide you with some ideas about how to visualize the relationships of data.

### **STANDARD BEARERS**

Back in the 1960s, IBM started to use a format called the standard generalized markup language (SGML). This type of formatting allowed IBM to standardize document production by putting the formatting information inside of “tags” that could travel along with the document. Fast-forward to 1999: The World Wide Web Consortium ([www.w3c.org](http://www.w3c.org)), a forum for information, commerce, communication, and collective understanding, publishes a standard based on SGML, the eXtensible Markup Language (XML).



by *Richard Hammond*

The XML standard is freely available and is “primitive.” It doesn’t do very much on its own. However, this is a powerful aspect of XML because in order to be useful, XML must be mixed with business rules. Therefore, the data managers cannot really build anything until the business leaders have carefully thought about and described what they want to learn from all that expensive data that every company collects. Libraries are in the same boat, with leaders and data managers collaborating to create appropriate business rules for library data.

#### GETTING YOUR SVG BEARINGS

One of the subsets of XML is the set of graphical tools known as scalable vector graphics, or SVG. What does SVG look like? It is a tagged markup language like all others, so the formatting part sits between tags—the “less than” and “greater than” symbols (< and >)—and the information sits between two tags (the open tag <> and the close tag </>). When SVG is viewed in an SVG compliant web browser, as basically all browsers are these days, the following tag will produce a rectangle. There’s a bit more to the code, but this is the part that does the job.

```
<rect width="50" height="250"/>
```

As a part of the definition of all XML standards, the code must be human-readable. That means the SVG code can be viewed directly. Many times, using a browser’s View Source

option (or equivalent) will reveal the full SVG code. Sometimes the code has been placed inside an HTML wrapper, but in the spirit of open standards, there are a lot of great samples available for viewing.

The SVG standard makes available a very powerful graphic arts tool, even though there are just a few shapes that SVG actually draws. In addition to rectangles, there are standards for drawing circles, lines, polygons, ellipses, and polylines (typically a set of lines that defines an open shape). There are also instructions to add color, shading, and even animation. A Google search of *SVG art* will provide a large store of really great examples.

#### COMBINING VISUALIZATION WITH MARKUP LANGUAGE

However, this article is focused not on creating great art, but on combining the visualization aspects of SVG with the underlying markup language. Looking in more detail at a tag, the container for all markup languages, note that tags come in pairs. They open “<>” and they close “</>.” When appropriate, it is possible to use a shorthand that puts the “/” within the first tag, as in <rect/>. The tag can be extended by adding more information, as in the following example:

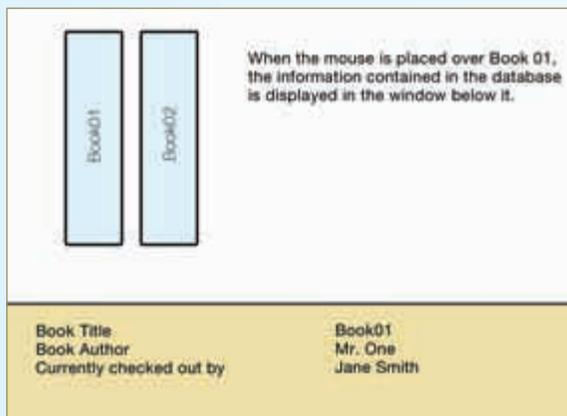
```
<rect id = "book01" x="20" y="20" width="50"  
height="250"/>
```

Without necessarily understanding the details, it is easy to see that once the rectangle has been designated as “book01,”

it can be specifically located in a list of rectangles. Extending the SVG a bit more, we can create the following tag:

```
<rect id = "book01" x="20" y="20" width="50"
height="250">
  <person>
    <last_name>Smith</last_name>
    <first_name>Jane</first_name>
  </person>
</rect>
```

The x and y values map the upper left corner of the rectangle to the underlying grid. Because the grid is human-readable like every other aspect of markup languages, there is a lot that can be done with the locational aspects of SVG as well, which I will explore in more detail at the end of this article. For now, imagine extending these examples to encompass the entire floor of a library, where each book could be listed separately along with its actual physical location within that library.



Here's an example of how extending these examples to encompass the entire floor of a library, where each book could be listed separately, along with its actual physical location within that library, would appear.

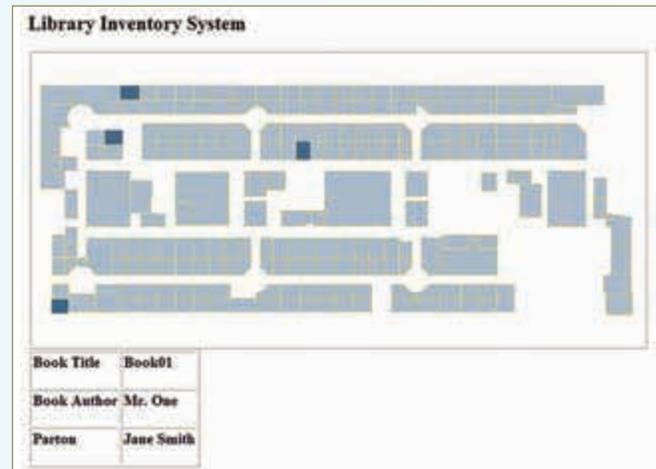
### VISUALIZING BOOKS

My next example has two books, located physically side by side (notice the different values for x) and checked out to two different people.

```
<rect id = "book01" x="20" y="20" width="50"
height="250">
  <author>Mr. One</author>
  <patron>
    <last_name>Smith</last_name>
    <first_name>Jane</first_name>
  </patron>
</rect>
```

```
<rect id = "book02" x="70" y="20" width="50"
height="250">
```

```
<author>Ms. Two</author>
<patron>
  <last_name>Doe</last_name>
  <first_name>John</first_name>
</patron>
</rect>
```



Because the tool is web-based, as each book is checked out and in (and the records are updated), the next time someone refreshes the webpage, that information is reflected immediately.

A very powerful aspect of this type of application is that the object retains its properties no matter where it is located. So even if the visualization of the book changes (for example, if the book were moved to a new location), the information associated with that book travels along with it. It would not be at all difficult to carry the entire history of that book—who had checked it out and when—no matter where it was visually placed in an application. This is a technical possibility, mind you, not the ethical one concerning whether the history of who checked out a book should be captured electronically.

### BUT IT HAS A BEARING ON THE MATTER

At this point, the separation between the data managers and the business managers starts to have some impact. To extend this example further requires the expertise of a person who is very familiar with the rules of a library system. It is important to note that this person does not need to be familiar with writing SVG code. That is certainly an asset, but it's far from a requirement. The example that follows is based on my knowledge of library operations, and I am far from a subject-matter expert on library operations. Remember also, the job of the subject-matter expert is not to understand all the nuances of coding (an author doesn't have to know everything about printing), but rather to carefully explain (sometimes in excruciating detail) the business of the library.

The SVG can be extended essentially endlessly to account for all types of business rules. The part of XML that does the



“programming” is called eXtensible StyLesheet Transformations, or XSLT. Again, it is nothing more than a human-readable tagged format standard. For this article, I am using pseudo code, but it is actually very close to the real thing. The XSLT is carried in a separate file, which is useful because as the rules of the library change, the business rules can be changed in one place. It is not necessary to search all over several applications to make sure the rules have been updated.

The pseudo code for an XSLT file could be the following (gt = greater than):

```
<xslt>
  <select rect id = "book01"/>
  <if return book01 gt Jan 1, 2010>
  send overdue notice and make book01 red and
  blinking
</endif>
</xslt>
```

So not only would an email automatically be sent to the patron (or, of course, if the business rules required, the email could be sent a few days before the overdue date), the graphic of the book would turn red and start to blink (or whatever literally endless combination of visual cues the business managers determine would be useful).

#### BEARING DOWN WITH JAVA, AJAX, AND HTML

Another useful and freely available standard for creating visual databases is JavaScript. The combination of XML and JavaScript is often called Ajax, which stands for Asynchronous JavaScript And XML. That means that the XML can call on the JavaScript as required. Traditionally (although there are plenty of exceptions), programming for databases and computers generally has been sequential, from one step to the next. Ajax is more compartmentalized or object-based. A little piece of JavaScript can be used over and over as required.

There's one more markup language standard, one many readers will already know—HyperText Markup Language, or good old HTML. As just another markup language, HTML is human-readable and uses open and close tags. Use HTML to house the applications and make them easier to put up as a website.

To review briefly, up to this point, I've used open standards—SVG, XSLT, and HTML—meaning they are freely available to everyone and the code is human-readable, and I've used JavaScript, which is not required to be human-readable but is still pretty accessible compared to historical programming languages. After a little practice, the subject-matter expert and the programmer can work very well together to produce prototypes and final operational products. An excellent source of all things XML and Ajax are the W3C schools ([www.w3schools.com](http://www.w3schools.com)).

What can be accomplished when the subject-matter experts and the XML programmers join forces? I have been

involved with several application development tools, which I will describe here to help present the power of these open standards. Remember that because these tools are basically webpages, or are at least housed in a webpage, they are essentially immediately accessible to the entire network.

#### DOCUMENT MANAGEMENT TOOLS

By giving each document a physical location (for example, “Shelf One, Row One”), it is possible to create a visual map of a document collection such as a library. This application allows individuals to see each book and where it is located. Adding in some JavaScript provides functionality to mouse over each book and see whatever data has been recorded about that book—is it checked out, is it overdue, is it in a special location for a promotional book display?

Because the tool is web-based, as each book is checked out and in (and the records are updated), the next time someone refreshes the webpage, the information is reflected immediately. Also, again because it is a web-based application, at the discretion of the business, this list can easily be shared publicly through a simple website.

#### PHYSICAL ASSET CONTROL

Adding radio frequency identification (RFID—small, physical tags glued inside each book cover) to each physical asset allows it to be tracked in a hands-free way. Because RFID is just another piece of data, the original document management tool can be extended to incorporate the individual RFID tags.

```
<rect id = "book01" x="20" y="20" width="50"
height="250">
<rfid>123456</rfid>
</rect>
```

```
<rect id = "book02" x="70" y="20" width="50"
height="250">
<rfid>98765</rfid>
</rect>
```

#### CONTENT MANAGEMENT TOOLS

Though it is not necessarily easy, considering that cataloging has to be done at some level at some time, it is possible to tag internal parts of a document. In fact, it is possible to tag every letter and every space in a document. That is pretty clearly not the right level of granularity unless you need that information for actual physical production of the document. However, I have personally used this application to select a topic from several documents and then combine the contents into a single, synthetic document. That allows me to search for a topic (here the history of XML) and because of the tags in each document, I can find the actual passage from each document. It is good to know where a book is, but it's even better to know the exact passages from every book. The following example illustrates this idea:

# Resources

## carto:net

[www.carto.net](http://www.carto.net)

This site offers students, graphic artists, and internet people the opportunity to share their ideas and projects.

## World Wide Web Consortium

[www.w3c.org](http://www.w3c.org)

The World Wide Web Consortium (W3C) is a forum for information, commerce, communication, and collective understanding that develops interoperable technologies (specifications, guidelines, software, and tools) to lead the web to its full potential.

## Adobe Scalable Vector Graphics

[www.adobe.com/svg](http://www.adobe.com/svg)

Adobe Systems, Inc.'s website provides technical documentation, downloads, and tools to extend the functionality of Adobe products, as well as open standards related to scalable vector graphics.

## Inkscape

[www.inkscape.org](http://www.inkscape.org)

Inkscape is an open source vector graphics editor, with capabilities similar to Adobe Illustrator, CorelDRAW, or Xara Xtreme, using the W3C standard SVG file format.

## SVG Wiki

<http://wiki.svg.org>

This is a moderated wiki designed to help the scalable vector graphics community—developers, artists, designers, viewer implementors, and anyone else interested in SVG—share information and code.

## SVG Open 2009

[www.svgopen.org](http://www.svgopen.org)

The 7th International Conference on Scalable Vector Graphics will take place from Oct. 2 to 4, 2009, in Mountain View, Calif. It will be hosted by Google at the Crittenden Campus.

```
<doc id="doc01">
<text>Here is some text.
    <history>Part of this text mentions the history of
        XML.</history>
    Part of it is about other things.
</text>
</doc>
```

```
<doc id="doc02">
<text>This is quite different text, but even within it,
    <history>there is some historical references to
        XML.</history>
    There are a lot of other things that do not have to do with
        the history of XML.
</text>
</doc>
```

The output would be a webpage.

```
<html>
<table>
<row1>get "doc01/history"</row1>
<row1>get "doc02/history"</row1>
</table>
</html>
```

The ideas of content management lead to the current frontier of XML, semantic accessibility. Other markup language standards, including the Web Ontology Language (OWL) and the resource description framework (RDF), are the building blocks to make content management a reality.

## BARE NAKED PEARL GROWING

I have presented a lot of ideas in this article. My objective is to act like that irritating grain of sand that will eventually grow into a beautiful pearl. Many of us have experienced the frustration of discussing and even arguing about what a business or a library is trying to accomplish, but we've also felt the pleasure of putting those plans into action and having our enterprises succeed. The XML standards provide one tool to help leaders and data managers bridge the communications gap so that programmers are not making decisions about the library's business. The standards are also helpful because they are web-based, so the programmers can rapidly share the tools they develop. It avoids months of development only to find that the tools are not very useful. There are days of development, followed by an email with a web address where everyone on the project can keep an eye on progress.

The power of the XML family is now well-established. As we move forward in the internet world, toward applications such as the semantic web, XML will play an increasingly important role. Once you get the hang of writing markup documents, you can create prototype applications very quickly. And because they are browser-based, you can share them as soon as you get finished developing the tool. Even if you never care to write an XML document, just learning about how it works will allow you to think of new ways to organize and present your data. That exercise alone will prove worth your efforts to become familiar with the XML family.

---

*Richard Hammond* ([rhamm123@bellsouth.net](mailto:rhamm123@bellsouth.net)) is a freelance writer in Atlanta. He writes about knowledge visualization as well as business/technology interactions.

Comments? Email the editor ([marydee@xmission.com](mailto:marydee@xmission.com)).



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.