

An introduction to XML and SOAP

A final year project discussion paper (number one)

Table of Contents

About the author	3
Abstract	3
The meeting	3
What is SOAP	3
Resolving concerns about TRIM	4
Response to the talk	5

About the author

Michael's student number is 964076. He can be contacted at mikal@stillhq.com. Copyright (c) Michael Still 2002. This submission is 1256 words, not including the code snippets.

Abstract

As far as Microsoft is concerned, the future of computing is .NET and distributed computing. A large part of their infrastructure for this is XML (Extensible Markup Language) and a protocol known as SOAP (Simple Object Access Protocol). SOAP is the subject of this discussion paper ¹. SOAP is essentially a XML RPC (Remote Procedure Call) over HTTP (Hyper Text Transfer Protocol) protocol.

The meeting

This report is based on a day long tutorial on XML and SOAP which I attended on 25 April 2002, as part of the *V\$Live!* Conference in Sydney, Australia ². The speakers at this session were Brian Randall ³ and Ken Getz ⁴.

What is SOAP

History

Microsoft has toyed with remote procedure call methods for many years. The most common of these at the moment would have to be DCOM (Distributed Component Object Model) ⁵. SOAP is an attempt to develop a RPC mechanism which will work well over the Internet. Interestingly, SOAP is also one of the few examples of a protocol championed by Microsoft which has been published as an open standard ⁶. This means developers outside of Microsoft are empowered to develop their own implementations if they want to, without having to resort to reverse engineering techniques.

One of the perceived problems with DCOM is that it doesn't work well over the Internet, especially because it requires its own network protocol be allowed transit through the intervening firewalls. SOAP attempts to address this by tunneling its RPC calls through HTTP, which makes it easy for many organizations to set it up within their existing firewall infrastructure. Incidentally, there is some discussion at the moment as to whether SOAP should be allowed it's own network protocol as well.

The example

When discussing the use of SOAP, we will use the example of TRIM, which is a heavily RPC orientated client server package developed by TOWER Software in Canberra.

It is a prime candidate for conversion to SOAP, as it currently uses DCOM, the previous Microsoft technology, for RPC calls over the Internet. There are currently concerns at TOWER as to the suitability of DCOM to the application. The main concerns are as follows:

1. Scalability
2. Suitability to low bandwidth and highly latent networks
3. Interoperability with non Microsoft client and server applications
4. Security

This paper will attempt to address all of these issues.

The protocol

SOAP is defined in terms of an XML DTD (Document Type Definition – a way of defining a grammar that your XML stream will obey for a certain type of transaction), which is what has been submitted to the *w3c* as a proposed specification. A RPC call (when using the HTTP transport) takes the form of a request for a given page, which the RPC arguments packed into an XML document which is posted to the web service⁷ using the same mechanism that POSTs use under CGI. The result of the RPC call is returned in the form of another XML document, which would be displayed in the browser if the client was a web browser.

It can be seen from this description that the protocol requires a fair bit of overhead. Not only do both the client and web service need to be able to talk HTTP, they also need to be able to generate and parse XML. Web proxies were also not originally intended for the style of traffic that SOAP will impose on them. At the moment if you browse a web site, then a series of connections to the web server are made⁸ and data is returned. The data is then rendered for the user. Even on a fast Internet connection this can be quite frustratingly slow. Imagine that those requests are now the challenge and response for a logon to TRIM, which currently takes less than a second.

One possible solution to at least part of this problem is to make client server protocols require less calls to perform common operations. It should be noted however that competent client server programmers already attempt to minimize the amount of network traffic incurred by their product.

Resolving concerns about TRIM

As mentioned earlier in this document, TRIM is a prime candidate for conversion to SOAP. Does SOAP help resolve TRIM's DCOM problems? Comparing them point to point:

1. *Scalability*: how to make HTTP connections scale is a well understood area⁹, especially because of the experiences of large web sites such as google.com. It is

important to note that the nature of these sites is often not mission critical however. People often expect google to be unavailable for periods of time, or even merely slow. TRIM is an application which is used all day every day by many people within an organization. I am therefore not convinced that converting the application to SOAP would make TRIM more scalable.

2. *Suitability to low bandwidth and highly latent networks*: compared with many TRIM customers, the Internet is not a low bandwidth highly latent network. Examples of networks on which TRIM must operate include encrypted low speed international network connections for national governments. SOAP packets are bigger than the equivalent DCOM packets, because of the large amount of bandwidth consumed by XML tags and other non-core data. This would increase the load on these already heavily loaded networks.
3. *Interoperability with non Microsoft client and server applications*: SOAP would resolve many of these problems.
4. *Security*: SOAP packets are not naturally encrypted or strongly authenticated unless a secure transport such as SSL is used. This is much the same as the current DCOM implementation.

Overall, it is my understanding at this time that there is no compelling reason to convert TRIM from DCOM to SOAP, as shown by the comparison above.

Response to the talk

The talk, as well as the speakers, were well presented. The conference proceedings includes over 260 slides for this day long session, even though additional slides were used to further explain topics that the audience had difficulty grasping. The speakers understood how to keep the audience engaged and interested for the entire length of the session as well, and there were plenty of well explained examples during the session.

One criticism of the speakers was that they were very Microsoft focused in their discussions, with most of the examples being in Visual Basic (the exceptions were in Microsoft's new language, C#). This is not entirely unexpected at a Microsoft development conference.

Notes

1. XML is considered to be outside the scope of this paper, but it is assumed that the reader has some knowledge of the workings of XML.
2. The conference website can be found at <http://www.vslive.com>. Repeats of this conference are held in Sydney, New York, and Orlando.
3. <http://www.vslive.com/2002/ny/speakers.asp#randall>
4. <http://www.mcwtech.com>
5. The distributed COM implementation, which unfortunately has problems dealing with low latency or low bandwidth network connections – for instance most Internet connections.

6. It is currently a draft specification with the *w3c* <http://www.w3.org/TR/SOAP/>
7. The name given to a piece of software which offers RPC facilities under SOAP.
8. Although HTTP 1.1 allows for all traffic to flow on one connection.
9. Although website scalability is often based on the implementation of server farms, which mandates that state not be stored between page accesses on the server, because the next page request might be answered by a different machine in the server farm.