

SOAP - Simple Object Access Protocol

A brief overview by Rob Bracken

What is SOAP?

In the good old days, programs were self-contained and did everything they needed to do – get information from the user, manage databases, etc. Then networking became common - the database side was split out and we had client/server systems. The client program still did most of the work, but a database server handled the heavy stuff. Next, the CORBA and COM technologies appeared, allowing us to split the programs into smaller parts – even placing different parts on different computers. Now, systems are becoming distributed outside the immediate enterprise, across the Internet.

When we try and use DCOM or CORBA's IIOP to talk to objects in other enterprises, however, we find a number of problems. The first is that we usually want to use a TCP port that is blocked by a firewall. Security administrators are – quite rightly - very nervous about opening ports on their firewalls, and the more enterprises we want to talk to, the more difficult the problem becomes. Even if we succeed and connect to the other enterprise, there's no guarantee that DCOM will work without further configuration – assuming, of course, that they're running Microsoft servers. Even different CORBA ORBs often have trouble talking to each other.

What's needed is a protocol that has a common implementation on all platforms and which can be carried by a commonly used transport. That's where SOAP comes in.

SOAP is a specification, which defines how to convert an object interface to an XML structure. The resulting XML can be sent to the target object by any transport protocol – such as HTTP or SMTP. Most firewalls allow external clients to send HTTP packets on port 80 (the standard port for a Web server), so SOAP packets will usually get into the enterprise which is their final destination. Once there, a SOAP server will be able to parse the XML, using pre-defined rules. An independent body specifies the rules, and a SOAP server is reasonably easy to write, so they should be available for a wide range of platforms. (SOAP was originally proposed by Microsoft, and relied on Microsoft-specific technology, but its scope has now been widened and the spec. submitted to the IETF as an Internet Draft.)

How does SOAP work?

The principles behind SOAP are straightforward. Suppose a user interface (the *client* object) wants to find out how many widgets are in stock at a supplier (the *server* object). The user interface creates some XML, which contains a reference to the server object, the name of the interface to the server object, the name of the method it wants to call, and any parameters to the method. It then uses HTTP to send the XML to the supplier (using the HTTP POST method). The supplier parses the XML, calls the requested method and gets the results. It creates some more XML - containing the response to the user interface's request – and uses HTTP (probably with the original connection) to return the results to the client. The client parses the XML and extracts the result.



Figure 1



Figure 2

This may seem like a lot of trouble to go to, but it's actually much the same as COM and CORBA do behind the scenes.

As mentioned, the client can use any method to send the XML. The most common will probably be HTTP, but – for instance – it could put the XML into a mail message and mail it to an SMTP server. The SMTP server would pass the message to a POP3 server. The mail recipient would parse the XML and carry out the request. The results would be mailed back to the original client, which would get them from its own POP3 server. In theory, it would even be possible to send the XML as a text file on a floppy disk, but I don't think it would be very practical!

SOAP Implementation

The SOAP standard only details how the data is structured. It does not specify how it should be implemented. Typically, clients will call methods on a proxy, which will convert them to XML and send them to a Web server. A CGI program or ISAPI DLL will parse the XML and will either instantiate a class object, or will call a method on an out-of-process object, using COM or CORBA. The object will do the method and return the results to the CGI/ISAPI, which will convert them to XML and pass them back to the client's proxy. The proxy will parse the XML, extract the results and give them to the client. Various XML parsers are available for a wide range of languages, so it shouldn't be difficult to write the proxy or server programs.

Of course, the CGI/ISAPI may simply pass the XML to the target object, which would contain its own proxy to convert method calls. This would be a more general solution than making the server program do the call.

Microsoft are using the ".sod" extension to identify SOAP requests (this evidently doesn't have the same meaning in the USA as it does here).

SOAP Security

One problem which SOAP developers will need to be aware of is security. XML is a plain text protocol. Anyone can read the contents, so anyone with a packet sniffer could – potentially – see anything that one object sends to another. We could use encryption – e.g. HTTPS – to send our SOAP packets, but this has an effect on performance.

Another aspect of security is that SOAP is designed to make it easy to pass through firewalls. Firewalls try to keep malicious users out of systems, so anything that reduces their effect must be a security risk. It is possible to configure them so that they block SOAP packets, but it's a hole that – I'm sure – will be exploited at some time.

More Information

You can find more information about SOAP at the following WWW addresses:

“What the heck is SOAP, anyway?” by David S. Platt - <http://www.byte.com/column/BYT20000816S0001>

A Young Person's Guide to The Simple Object Access Protocol - <http://msdn.microsoft.com/msdnmag/issues/0300/soap/soap.asp>

SOAP: The Simple Object Access Protocol (article)- <http://www.microsoft.com/mind/0100/soap/soap.asp>

Apache SOAP implementation – <http://xml.apache.org/soap>

DevelopMentor (Perl & Java implementations, SOAP mail forum, links to SOAP specifications) - <http://www.develop.com/soap/>

SOAP: Simple Object Access Protocol (specification) - <http://msdn.microsoft.com/xml/general/soapspec.asp>

Simple Object Access Protocol (SOAP) 1.1 (W3C note) - <http://www.w3.org/TR/SOAP/>

Want to know more? See Craig Murphy's article in this issue, and stand by for further episodes of SOAP next time.

Ah! – Acronym Heaven (otherwise known as the Glossary)

CGI – Common Gateway Interface

COM – Component Object Model

CORBA - Common Object Request Broker Architecture

DCOM – Distributed Component Object Model

HTTP – HyperText Transfer Protocol

HTTPS – HyperText Transfer Protocol (Secure)

IETF – Internet Engineering Task Force

IIOOP – Internet Inter-Orb Protocol or Internet Interoperable Object Protocol

ISAPI – Internet Server Application Programming Interface

POP3 – Post Office Protocol 3

SMTP – Simple Mail Transfer Protocol

SOAP – Simple Object Access Protocol

TCP – Transmission Control Protocol

WWW – World Wide Web

XML – eXtensible Markup Language



Rob Bracken is Managing Director of Bracken Software Limited and has over 10 years' experience of trying to make different programs talk to each other. You can contact him at rob_bracken@bramway.freeserve.co.uk.