Differences between SOAP 1.1 and 1.2
http://hadleynet.org/marc/whatsnew.html#S1.

RPC

This section describes the differences between the SOAP 1.2 and SOAP 1.1 RPC conventions.
Use of RPC on the World Wide Web

A common implementation pattern for SOAP 1.1 is to deploy a single SOAP endpoint at a particular URI and then internally dispatch SOAP requests to application logic. This approach hides all of the resources provided by the endpoint behind a single URI and prevents direct addressing of individual resources. E.g. a travel company might provide the following services via SOAP: flight searches, flight booking, hotel searches, hotel booking, car booking, itinerary management, travel insurance services, etc. However if all of these services are hidden behind the same URI then they are effectively invisible in URI space. This approach effectively reduces the world wide web to a transport layer and negates the impressive robustness and scaleability of the current web architecture.

SOAP 1.2 recommends that, where practical, separate resources are identified by separate URIs such that SOAP endpoints fit into the web architecture in the same way as other web accessible resources. This has the added advantage that SOAP resources are now suitable for use with the HTTP GET method instead of being tied to HTTP POST (see HTTP GET Support).

HTTP GET Support

The HTTP binding for SOAP 1.1 only discusses use of the HTTP POST method for carrying SOAP messages. This has been widely criticized since it negates the optimisation possibilities conveyed by the semantics HTTP GET – namely that the action performed in response to the GET is both safe (information retrieval only) and idempotent (side effects of N>0 identical requests are the same as for a single request). In practice this makes GET requests good candidates for cacheing and also permits use of read locks in databases etc.

SOAP 1.2 adds support for the use of HTTP GET in the SOAP HTTP binding. The semantics of HTTP GET are respected such that the action performed by SOAP endpoint responding to a request transmitted via HTTP GET should be both safe and idempotent.

An example of a HTTP GET request is shown below:

GET /itinerary?reservation=1234567890 HTTP/1.1
Host: travelcompany.com
Accept: application/soap+xml

Axis and WSDL support

http://www-128.ibm.com/developerworks/xml/library/x-tipgetr.html

At the time of writing, (Mar 2004) Axis supports SOAP 1.1 only, but still implements a limited form of GET. You can invoke any request through its URL, followed by a method parameter and the other parameters.

For example, suppose I have implemented the order status service at the URL http://joker.psol.com/axis/order/Status.jws. The service has two methods, track and detailTrack, which take the order number (onumber) as a parameter. I can invoke the service through http://joker.psol.com/axis/order/Status.jws?method=track&onumber=56544322.

WSDL 2.0 (currently under development at the W3C) adds a pattern attribute to operation definitions. This attribute indicates which SOAP MEPs a service supports (WSDL calls MEP patterns so there's no confusion between SOAP and WSDL). For example, the tracking service may be defined as in Listing 1.

Listing 1. WSDL excerpt

```
<operation name="track" pattern="http://www.w3.org/2003/11/wsdl/out-only">
  <output message="trackResponse"/>
</operation>
```