

WebServices - Axis

1. WebServices - Axis - SOAPVerse

Hi folks!

This is a quick writeup of an idea that a bunch of folks had last week while discussing interoperability demos and tests. It's a pretty simple system which we thought was a) fun, b) technically interesting, and c) quite a compelling demo. I'd like to know what people think of the idea - is this too ambitious, is it something you'd be psyched to help design/implement, is it cool?

The SOAPVerse : A long-term SOAP interoperability demo

[1.0 Introduction - the view from outside]

I'll start explaining the idea by giving a brief scenario. You connect a browser to SOAPVerse.org, which gives you three choices - 1) enter the SOAPVerse, 2) look at the map, and 3) learn about joining. You choose #1, and are offered a list of available clients and "entry portals" (i.e. clients (no, not "IE clients", necessarily...)) on the web. You choose a local entry portal, and a Java applet appears, primarily composed of a text window:

SOAP Tower

You stand in the SOAP tower. The floor's a bit slippery here, but you suspect you could make it to the exits to the NORTH or EAST if you walked slowly.

There is a briefcase sitting here.

(this room lives at foo.ibm.com, and is powered by Tomcat/Apache-SOAP 2.1!) -----

It's a text adventure, much like Zork or Colossal Cave, but a lot simpler. The interesting part happens when you move to the East:

[a strange feeling overcomes you for a moment as you pass through the door]

Campus West

You stand on the Microsoft campus, near building 33. You may ENTER, or travel WEST or SOUTH down the main road.

Others in this room : KeithB

There is a rubber ducky sitting here.

(this room lives at bar.microsoft.com, and is powered by IIS/ASP.NET!)

What just happened is that you smoothly and transparently moved from one SOAP-based server to another. The servers had to interoperate to "pass you off", and anyone who wants to go check out the website can see the deeper technical explanation of what's going on.

If you'd selected the "map" option, you'd see a cool graphical depiction of the whole graph of rooms currently connected to the SOAPVerse, color-coded by host/server technology.

[2.0 Digging a little deeper]

That's the basic idea - a totally distributed text adventure game that demonstrates SOAP interoperability at a number of levels. The actual APIs are pretty simple, and should be implementable in few days at the most.

So if you go to the "join us" section of the site, you end up with several things. First, a description of the structure of the application, in enough detail that you could implement it on your own site. This can (and should) be in as many forms as possible - english text, WSDL, SDL, IDL, etc.... So you build the server to the spec, in any language/environment/platform you happen to have handy.

Next, you find a form which allows you to test your server once you've got it up. This causes the SOAPVerse server to run a series of tests against your endpoint, to see if you can interoperate with it. Assuming that works, you can click "hook me up!" and the SOAPVerse server randomly picks a place on the graph to add your area, and matchmakes a connection between your server and whoever you're connecting to. The tests should get run again between you and this new guy, to make sure you two interoperate (you don't want to just prove interoperation between the "main" server and your impl), and then if everything looks good, you're now a part of the world, and your rooms appear on the master map.

There's some more detail about which kinds of things we're testing with a system like this (data serialization, headers, intermediaries?), actual APIs, etc. but I'll convey my thoughts about that in a design discussion if there's enough community interest in this project.

[3.0 Musings]

This kind of thing serves at least two purposes. First, it can stay up in perpetuity,

WebServices - Axis

demonstrating SOAP interoperability in a fun way. This should be something you can always find, and hook new servers into. Second, it's a good demo for tradeshow-type events.

Obviously there's a lot of opportunity for errors to happen here, so the system shouldn't assume too much about robustness, and should gracefully fail in the face of problems. It's meant as an interoperability demo, not a full-scale game.

None of this is at all carved in stone, we just liked the basic idea. It shouldn't get too complicated, and it shouldn't rely on any particular implementation.

If this could get done by late next month, this could be the actual technology for the "interopathon" demo which has been discussed for NetWorld/Interop in May.

What do you think?

--Glen