

WS-Routing, WS-Referral & DeW Framework

(By Xunfan Xia, xxia@usc.edu, December 5, 2002)

The DeW framework

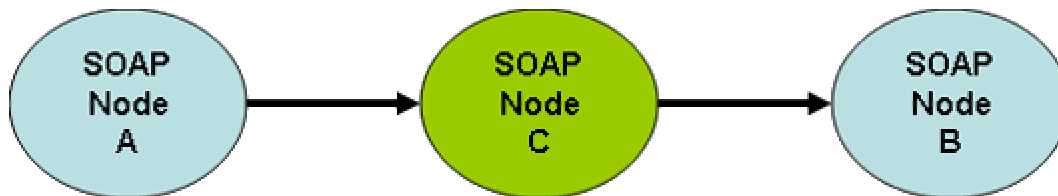
The DeW framework allows the Web Services to be deployed physical-location-independently. The Web Services will register their exception handlers in Dew Registry. The Web Service client will raise Dew Exception when there are dynamic changes of the Web Service, and then the proxy who is handling the Dew Exception will contact Dew Registry, download its handler and execute it. So any location changes of the Web Services will not affect the user.

WS-Routing, WS-Referral

The WS-Routing, WS-Referral are part of GXA specifications. They are mainly aim the same goal as DeW framework, which is the location-independent. The WS-Routing shows how to forward the soap messages to other SOAP node, which will forward the Web Service to another machine. And the WS-Referral shows how to insert/remove the SOAP node dynamically in Routing Server.

The following example shows the insertion of a SOAP node before another node, we can also remove a SOAP node in the path.

```
<r:ref xmlns:r="http://schemas.xmlsoap.org/ws/2001/10/referral">
  <r:for>
    <r:prefix>soap://b.org</r:prefix>
  </r:for>
  <r:if/>
  <r:go>
    <r:via>soap://c.org</r:via>
  </r:go>
  <r:refId>mid:1234@some.host.org</r:refId>
</r:ref>
```



WSDK

Microsoft released Microsoft Web Service SDK Preview(WSDK) in August, 2002. It says it supports [WS-Security](#), [WS-Routing](#) (and [WS-Referral](#)), and [DIME](#) and [WS-Attachments](#). But the WS-Referral is only partially implemented. The WS-Routing server is using the static configuration information defined by WS-Referral, but it doesn't

support dynamically changing the routing information which is specified in WS-Referral specification. It is expected to be implemented in the next WSDK release in mid December, 2002. (Per communication with Byung Chan Kim(bckim@microsoft.com), Jim Huang(jhuang@microsoft.com) from Microsoft on 11/26/2002).

According to them, in order to update the referral Cache of the WS-Routing server dynamically, we need to do the following:

The Client, to add/remove a node:

```
proxy.RequestSoapContext.Referrals.Add(ref);
```

The Server:

Derive custom router from the RoutingServer class

In ProcessRequestMessage function of the custom router:

```
protected override void ProcessRequestMessage(SoapEnvelope
env,Microsoft.Web.Services.Routing.Path outgoingPath)
{
    SoapContext ctxt = env.Context;

    base.ProcessRequestMessage(env, outgoingPath); //Referral Lookup
and update path here

    // if there is any referral and it is targeted for this
actor, try to register them in local referral

    if(ctxt.Referrals.Count > 0 &&
(ctxt.Referrals.Actor.Equals(Soap.ActorNext) ||
ctxt.Referrals.Actor.Equals(ctxt.Actor.ToString())))
    {
        foreach(Referral r in ctxt.Referrals)
            ReferralCache.Register(r);
    }
}
```

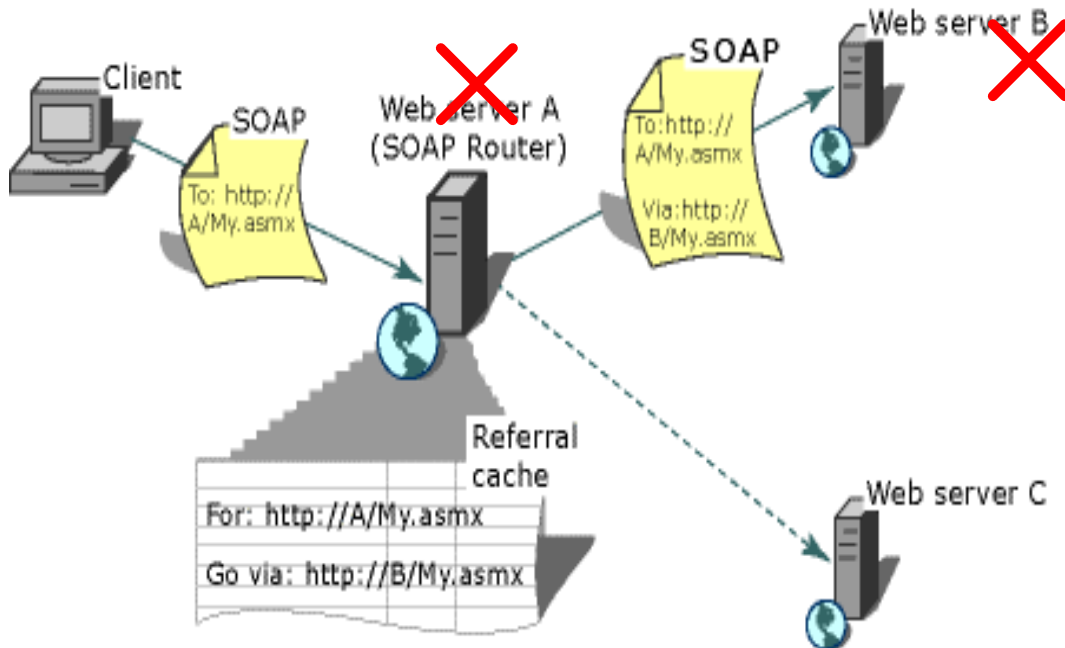
This will update referral cache at Router.

DeW framework and WS-Routing Server

The following picture shows how the WS-Routing Server works. When clients asking a web service from Server A, Server A will actually go to Server B to get it. And Server A can be changed to go to Server C if there is something wrong with Server B.

There are two problems here:

1. If Server A has problem, clients will not be able to access the web service.
2. When Server A is configured to go to Server B and B has problem, there is no dynamically switch from B to C. The clients will also got problem.(Possible solve it by implementing a custom router shown in above section using the new WSDK release).



By combining the DeW framework and WS-Routing Server together will solve these problems. Assume A and B register their handlers in DeW registry, A's handler will forward the request to B, B's handler will forward the request to C.

So for the first problem, when a client accesses A and A got problem, an exception will be raised and the downloaded handler from DeW registry will forward the request to B. This is implemented in our DeW framework demo.

The second problem, A is configured to go to B and B got problem. This time A is acting as a client of DeW framework, it will download the handler and forward the request to C. However, this case is different from the first one. In the first problem, the client access the Web Service using the proxy we provided in DeW framework and it will handle the DeW exception and download the corresponding exception handler and execute it. While in this case, the WS-Routing Server is not built with DeW proxy, it forwards the SOAP messages internally. So we have to implement it differently from the first one.

We are not able to implement this with the current version of WSDK(Preview). Since first, we can't derive from the WSDK RoutingHandler. Second, if we define our own RoutingHandler and use WSDK RoutingHandler in it. We can't get the routing fault code from the context, the response has been written to the response Stream, and the context variable is not being set. There shouldn't be problem with the new release of WSDK.

As we seen in last WSDK section, we should be able to update the referral cache after calling the `base.ProcessRequestMessage()`. We should also be able to check the result of this function in the context, if there is any Routing fault then we will do the same as the DeW exception is raised. The following flow-chart shows how it works:

In custom router ProcessRequestMessage() function:

