

Service Oriented Computing Week7 Lab

Q: Anything that you can enhance/add to an ESB? Turn in a one-page document, explaining your ideas.

During this lab, I take a short experience on different ESB frameworks, like Apache Servicemix, Mule ESB, and Zeto (which is a pure python based framework), what I feel is that how these frameworks are different from each other. The difference is not only on the user interface or the design interface, like Mule has its own IDE, which is a mod of Eclipse, while Servicemix is completely isolated with GUI, you need to put blueprints or other component to feed it. The underlying of different platforms varies a lot. The attached files illustrate how different the structures are in Servicemix and mule.

The inconsistency between different ESBs not only add the workload integrating a service into different ESBs but also results in the burden connecting different ESBs into a "greater ESB".

The purpose of ESB is to integrate different services of different interfaces thus reducing the workload of each individual transactions between different services. But doesn't it sounds strange that ESBs themselves don't have a uniform interface? Currently, ESB is used to carry the services registered in it and routing the messages or managing the transactions in the service bus. When compared with TCP/IP network, it sounds like a switch which only take charge of the message routing in its own network or service bus here. What we can enhance here is that whether we can improve the ESB to have the function of a router. We know that one of the differences of a switch and a router is that switch is only functional is one subnet, while routers can route messages between different subnets. We can imagine one ESB and its following services compose one subnet, we can connect different ESB platforms to become a network including different ESBs.

There are maybe two feasible ways to achieve it. One is setting up a central Bus which is able to connect different ESBs from different vendors. But the centralized solution seems can become a single point of failure which may result in blackout in the whole network (which is also an inevitable problem in current ESB frameworks).

The other solution maybe implementing the ESB as a service so that an ESB can be registered to another ESB as a service. Thus message routing over different ESB can be achievable like package transport in TCP/IP networks. But this requires an uniform interface to carry on all these ESB platforms which is also difficult to achieve now.

Weiyi Wang

Weiyi.wang@sv.cmu.edu

10/16/15