5. Interactions Among CSP and Service Providers

We saw that in case of composite web services, the various services needed to completely serve a service request are often offered by different service providers. Therefore, processing of a service request in such a situation involves interactions among the Composite Service Provider (CSP) and all the Service Providers (SPs) that offer one or more services of the service request. On careful inspection, we found that during processing of a service request, the CSP may interact with the concerned SPs by using one of the following interaction models:

1. Recursive model
2. Iterative model
3. Transitive model

In this chapter, we describe these three interaction models and present their relative advantages and limitations. Based on this analysis, we then choose the right interaction model for our framework. We also propose two interaction models for SP-to-SP service request and show why the latter model is suitable for use with our framework.
5.1 RECURSIVE MODEL OF INTERACTION

Figure 5.1 illustrates the method of interaction among the CSP and the concerned SPs in this model. Suppose the service request of a customer requires services of three different SPs. The customer makes the service request to the CSP. The CSP neatly partitions the service request into three services and identifies three different SPs for offering the services. It then sends this information to the first SP (SP-1), which offers its service and passes on the result of its service and the information received from the CSP to the next SP (SP-2). SP-2 then offers its service and passes on the results of services offered by SP-1 and SP-2 and the information received from the CSP to the next SP (SP-3). SP-3 then offers its service and returns the result of its service to SP-2, which returns the results to SP-1, which returns the results to the CSP. Finally, the CSP consolidates the results and sends a response message to the customer.

Figure 5.1. Recursive model of interactions among CSP and SPs.
5.2 ITERATIVE MODEL OF INTERACTION

Figure 5.2 illustrates the method of interaction among the CSP and the concerned SPs in this model. As shown in the figure, in this method, SPs do not interact with each other directly. Rather, the CSP retains control over the interaction process and one-by-one interacts with each of the SPs involved in the customer's service request processing process. As in the recursive method, the CSP first interacts with SP-1 to get that part of service request serviced by this service provider for which it has been selected. SP-1 processes its part of the service request and returns the result of processing to the CSP along with any other information that the CSP might require for further processing of the service request. The CSP next interacts with SP-2 and then with SP-3 one-by-one in the same manner. The process continues like this until the CSP has finished processing all parts of the customer's service request. Finally, the CSP consolidates the results and sends a response message to the customer.

Figure 5.2. Iterative model of interactions among CSP and SPs.
5.3 TRANSITIVE MODEL OF INTERACTION

Figure 5.3 illustrates the method of interaction among the CSP and the concerned SPs in this model. As shown in the figure, in this method, when a customer's service request is received by the CSP, it neatly partitions the service request into three services (say for this request only three types of services are required) and identifies three different SPs for offering the services. It then sends this information to the first SP (SP-1), which offers its service and passes on the result of its service and the information received from the CSP to the next SP (SP-2). SP-2 then offers its service and passes on the results of services offered by SP-1 and SP-2 and the information received from the CSP to the next SP (SP-3). SP-3 then offers its service and passes on the results of services offered by SP-1, SP-2 and SP-3 to the CSP. Finally, the CSP consolidates the results and sends a response message to the customer.

Figure 5.3. Transitive model of interactions among CSP and SPs.
5.4 CHOOSING THE RIGHT MODEL FOR OUR FRAMEWORK

To choose a model suitable for our framework, we did a comparative analysis of the three interaction models and concluded that the iterative model of interaction is the right model for our framework. The comparative analysis performed by us is presented below.

In the recursive model of interaction, the CSP has little work to do but the SPs may be involved in processing several requests from different CSPs at the same time. Therefore, the SPs may get overloaded in situations where the number of CSPs is too large as compared to the number of SPs. Hence, this model is not suitable for use in those environments in which the ratio of CSPs to SPs is high. Also, notice that there are eight interaction messages involved in this model for processing a service request involving three SPs.

In the iterative model of interaction, the CSP has more work to do but it retains total control over the interaction process. The CSP may get overloaded in situations where the number of CSPs is too less as compared to the number of SPs. Hence, initially it may appear that this model is not suitable for use in those environments in which the ratio of CSPs to SPs is low. However, this limitation of the model can be easily overcome by creating more instances of CSP to suitably balance this ratio. Also, notice that like the recursive model, this model also requires eight interaction messages for processing a service request involving three SPs.

As in the recursive model, in this model also the CSP has little work to do. Also, notice from Figure 5.3 that the transitive model requires the fewest number of interaction messages (only six messages for processing a service request involving three SPs, as opposed to eight messages required in case of recursive and iterative models). However, a sender (SP) does not receive any acknowledgement message once it passes on the service request to another SP. Therefore, this model should be used in systems with reliable communication. As we are dealing with Internet-
based web services and since the Internet functions on best effort basis without any
guarantee of reliable communication, this model is not suitable for our application
scenario. Moreover, recursive and iterative approaches can be efficiently supported
by RPC-based (Remote Procedure Call-based) communication systems because
they use a “call-response” model. Hence, they are easier to implement.

Based on the above analysis of the three interaction models, we have chosen the
iterative model as the right model for our framework because it has the following
advantages:

1. It allows the CSP to retain total control over the interaction process. This is a
very important advantage for our framework because its primary objective is
privacy protection. Protection of privacy of customers’ private information
can be best handled with this model because the CSP has full control of what
information is shared with which SP.

2. As the CSP directly interacts with each concerned SP in this model,
transaction tracking and registration activity of the framework can be easily
implemented within the interaction process. No separate interaction is
needed with any SP to collect such information.

3. The model automatically supports acknowledgement of each request in the
form of reply for the request. Hence, it is suitable for use for the
environment (Internet-based web services) for which our framework is
designed.

4. The model can be easily implemented and efficiently supported by RPC-
based communication systems because of its “call-response” nature of
interactions.

5.5 INTERACTION MODEL FOR SP-TO-SP SERVICE
REQUEST

There is a possibility that while processing a service request received from CSP, a
SP (say SP-1) requires the services of another SP (say SP-2) and would like to
place a service request on it. SP-1 can continue processing the original service request after SP-2 completes this service request and SP-1 receives its result of processing.

Initially, one would think that this type of interaction requirement is similar to subroutine “Call” and “Return” and subroutine call-return model would be the right model to support this type of interactions in the framework. Figure 5.4 illustrates the flow of control (interaction model) in this model. The steps below correspond to the numbers written on the lines in line diagram of Figure 5.4.

1. Customer makes a service request to CSP
2. CSP processes customer’s service request
3. CSP requests the selected SP (SP-1) for its privacy policies
4. SP-1 sends its privacy policies to CSP
5. CSP does compliance check of customer’s privacy preferences and SP-1’s privacy policies. It also does customer and/or SP-1 enforcement to negotiate for compliance (if required)
6. CSP requests SP-1 to service the request (partially/fully)
7. SP-1 processes the service request
8. SP-1 requests SP-2 to service a part of the request
9. SP-2 processes its part of service request
10. SP-2 returns result of processing to SP-1
11. SP-1 continues processing the request
12. SP-1 returns result of request processing to CSP
13. CSP continues processing the customer’s service request
14. CSP returns result of processing to the customer

However, we found that this model is not suitable for our framework because:

1. Our framework requires compliance check between customer’s privacy preferences and service provider’s privacy policy. Only the CSP has the information (customer’s privacy preferences) and know-how (routines) for doing this.
2. Our framework does consumer enforcement and/or service provider enforcement to negotiate for compliance, if customer's privacy preferences and service provider's privacy policy do not comply. Only the CSP has the know-how (routines) to carry out these types of negotiations.

3. The framework requires the CSP to perform transaction tracking and registration for all information shared with any SP during the processing of...
a transaction. This cannot be guaranteed if SPs interact directly without involvement of CSP.

Hence, for our framework we had to invent a suitable interaction model for SP-to-SP service request. The interaction model that we chose to support this type of interactions in our framework is as follows (see Figure 5.5). The steps below correspond to the numbers written on the lines in line diagram of Figure 5.5.

1. Customer makes a service request to CSP
2. CSP processes customer’s service request to identify suitable SP (or SPs) for servicing customer’s request
3. CSP requests the selected SP (say SP-1) for its privacy policy
4. SP-1 sends its privacy policy to CSP
5. CSP does compliance check of customer’s privacy preferences and SP-1’s privacy policy. It also does customer and/or SP-1 enforcement to negotiate for compliance (if required)
6. CSP sends customer’s service request to SP-1
7. SP-1 processes the service request and realizes that it needs SP-2’s service to complete the request processing
8. SP-1 sends this information to CSP along with SP-2’s ID
9. CSP requests SP-2 for its privacy policy
10. SP-2 sends its privacy policy to CSP
11. CSP does compliance check of customer’s privacy preferences and SP-2’s privacy policy. It also does customer and/or SP-2 enforcement to negotiate for compliance (if required)
12. CSP sends SP-1’s service request to SP-2
13. SP-2 processes the service request
14. SP-2 returns result of processing to CSP
15. CSP forwards this result to SP-1
16. SP-1 continues processing the remaining part of the customer’s service request
Figure 5.5. Illustrating flow of control (interaction model) proposed by us to support SP-to-SP service request in our framework.

17. SP-1 returns result of request processing to CSP
18. CSP continues processing the customer’s service request
19. CSP returns result of processing to the customer

Notice that as compared to the subroutine call-return interaction model of Figure 5.4, the proposed interaction model of Figure 5.5 for SP-to-SP service requests has following advantages (although it requires more interaction messages):
1. It allows the CSP to retain total control over the interaction process. This is a very important advantage for our framework because its primary objective is privacy protection. Protection of privacy of customers’ private information can be best handled with this model because the CSP has full control of what information is shared with which SP.

2. In this model, as the CSP directly interacts with each SP involved in processing of a service request, transaction tracking and registration activity of the framework can be easily implemented with this model.