Position Paper for Workshop on Web Services for Enterprise Computing

Torry Harris Business Solutions
Ground Floor, Sona Towers, 71, Millers Road,
Bangalore 560052. India
www.thbs.com

Torry Harris
Distributed e-enterprise Computing

Use Case 1: Provision of a time-out parameter for a web service

Problem Formulation

At present, there is no generic way to associate a "***time-out***" parameter with a Web-service (or the *operation*, to be more precise) contract. This could lead to inconsistent and perhaps even incorrect handling of service invocations in a Web-Services based architecture.

In conjunction with the loosely-coupled, distributed nature of Web-service based systems and allowing for suitable network latencies too, the provision of associating and knowing the SLA in terms of processing time for an invocation of a Web-Service operation would greatly enhance the quality of service provided.

Use Case

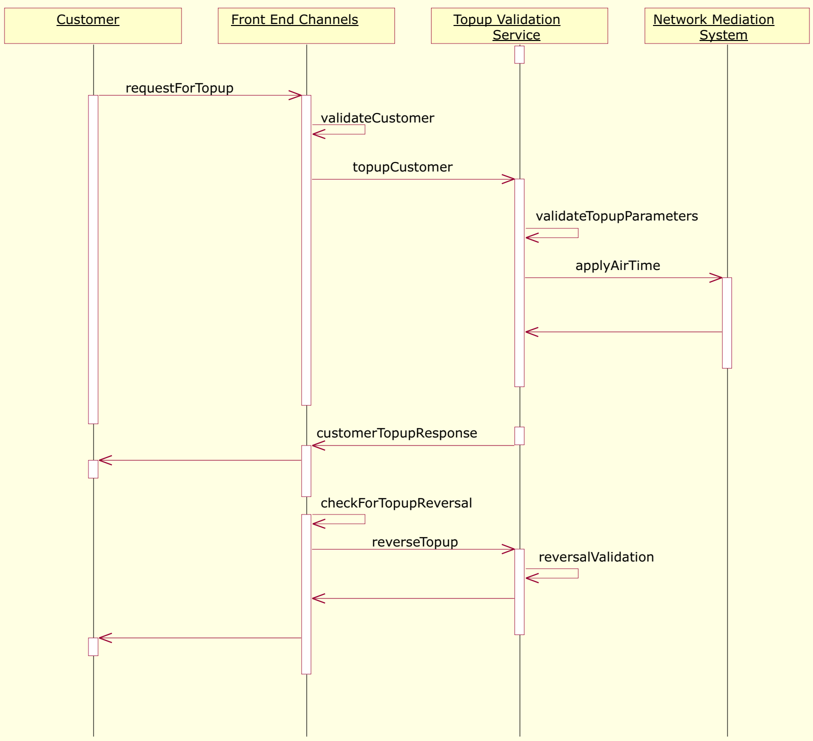


Figure 1. Sequence Diagram of ApplyAirtime Business Process

The sequence diagram explains the ***ApplyAirtime Business Process*** use case, commonly found in the Wireless Telecommunication domain. It involves application of a top up in terms of talktime to a customer's account, who has paid for the same, through an appropriate Front End channel. The scenario is explained as follows:

1. The Front end channel typically has an SLA in terms of elapsed time by which it has to inform the customer whether his top up request has been processed successfully or not.
2. The Front end channel invokes the backend logic via the Topup Validation Service. Keeping in mind the SLA with the customer, the Front end channel sends a failed response to the customer if the SLA is

not met (by the Topup Validation service), even though the topup may have been successfully completed by the network mediating service.

3. After sending the failed response, the Front end channel initiates another request to revoke (**reversalValidation** in **Figure 1**) the topup operation.

Issues

1. If the above mentioned use case were to be constructed by way of having the Topup validation service as a web service, the onus of implementing the above mentioned logic in terms of SLA compliance would lie with the caller of the service i.e the front end channel.
2. There could be numerous front end channels that would be invoking this service and all of them would need to implement the same logic.
3. The Front end channel implementation team(s) may not have any knowledge about the actual processing time required by the Topup Validation service. This is could lead to a high volume of SLA breach cases taking place.

Derived Requirements

Based on the above use case, in order to provide a better customer experience and also to provide a cleaner implementation, the following could be considered:

- Having a ***"time-out"*** or an ***"Upper Limit on Response-Time"*** SLA parameter to be associated with every Web Service (the Topup Validation Service in the above case), and desirably, at the Operation level. This should perhaps be part of the service contract.

  Having such a parameter associated with a service will enable the clients of such a service to gracefully handle cases in such a way as to avoid any (highly probable) inconsistencies & undesirable side-effects. This property would allow for a web service operation to specify itself as to what is the SLA that it would adhere to. This appears quite natural as the operation would know best as to what SLA it can support. This SLA parameter would also simplify calculating process SLAs for processes that orchestrate the invocation of several web services.

Specifications affected

- The WSDL specification

Recommendations for W3C

The "***time-out***" parameter applied at a web service operation level would be a good feature to have. This parameter would be a natural way for an operation to specify the SLA that it would adhere to and would also allow the caller to know the expected SLA when this operation is to be invoked. This could be achieved in the following way:

We could have a ***time-out*** element associated with each operation of a Web-Service in the WSDL file associated with it. Better still, we could have a separate "SLA" (XML) element associated with every operation, so that in the future, we could add more parameters pertaining to the SLA of a Service Provider. This approach is better since it is clearly easy to extend in the future as we identify more such parameters. The SLA element could initially be made optional too.

Also, the Web-Service Container could (desirably) take appropriate action(s) when it "*discovers*" (from the WSDL element, which will now (optionally) have the SLA element) the *time-out* as an SLA parameter for a given operation of the Web-Service deployed on it. For example, if the Container did NOT receive any response for the mandated time-period, it could convey the appropriate kind of "*Exception*" or an "*Error Message*", thus informing the client that the "intended operation invocation '***timed-out***' ", so that the invoking party can take suitable action.

Use Case 2: Provision of B2B Business Semantics

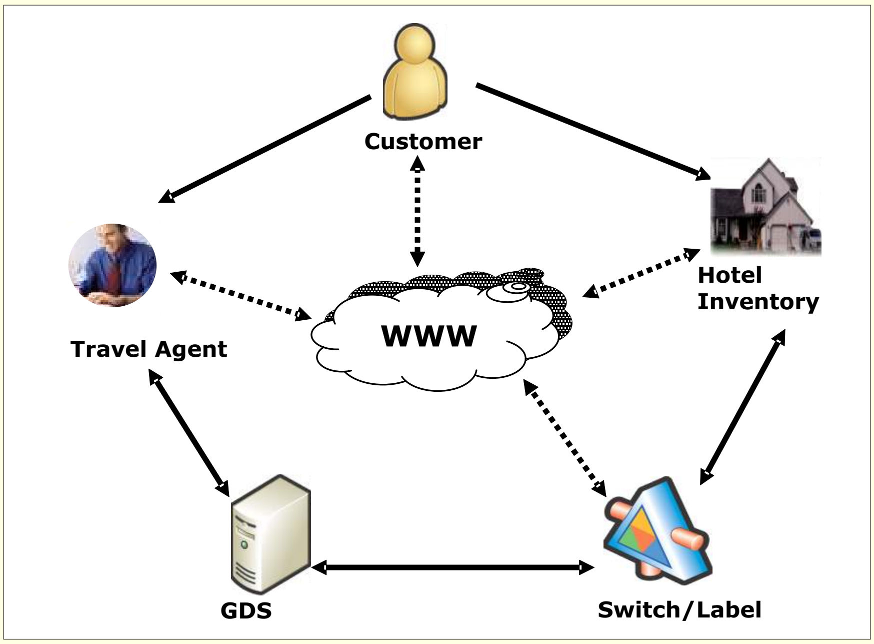
Problem Formulation

At present all the specifications in place or being formulated for the web services stack deal with addressing the underlying 'plumbing' for web services. 'Plumbing' in this context refers to setting in place the mechanism for web services based interactions between systems in a inter/intra enterprise scenario. This includes tackling issues like security, reliability, atomicity and others with respect to web services based transactions.

For the web services model to become ubiquitous the role that they can play in implementing business transaction semantics between enterprises perhaps need to be appreciated and the issues therein addressed.

Use Case

The travel and hospitality segment has witnessed explosive growth in terms of web based models being used for business transactions. The main players here are depicted in the diagram below:



Global Distribution Systems (GDS) form the heart of the travel industry. They host static information and provide access to the inventories and rates of the offerings from the various service providers. The reservation systems of individual service providers like Hotels, Airlines etc. have lacked the ability to reach the customer base worldwide and hence rely on the GDSs and the network of travel agents that interact with the GDS. Examples of companies owning GDSs are Amadeus, Sabre and Gallileo.

Switch/Label providers support the capability for individual reservation systems of small hotel chains / rental agencies to interface with the GDSs. THISCO and Wizcom are two of the better known companies offering this service.

Travel agents interact with the GDSs using proprietary software unique to each GDS. Customers normally use the internet to interface with travel portals which in turn interact with the GDSs.

Issues

Increase in web based interactions: This sector is experiencing rapid growth in terms of the internet being used for business transactions. Due to interoperability issues, the implementation of interactions between service providers and GDSs is a costly and time consuming affair which is inhibiting the growth of web based transactions.

Lack of Interoperability: Different GDS platforms are built on disparate technologies and support proprietary applications / data formats for travel agents and service providers to interface with them. The same applies to interactions between the switch companies and the GDSs.

Proprietary knowledge: Due to the complexities of the GDS systems and the custom manner in which one interfaces with them, any interaction requires human intervention in terms of resources that are very conversant with these complicated systems.

Derived Requirements

While the above mentioned issues clearly demonstrate that web services can play a major role in the interactions between the different players, there is a more fundamental issue. This is the lack of uniform and widely accepted semantics governing the interactions between the different players. Explaining this further, let us take the interactions between the travel agents and the GDSs. In most cases this would involve activities like (a) query inventory status and rates (b) make a booking (c) cancel booking (d) upgrade booking. As can be seen the number of interactions are **finite** and the data exchanged in these interactions can also be clearly **classified**. These characteristics lend themselves to the fact that taxonomies can be developed for each of these interactions which would address:

(1) Service areas and operations within them

(2) Data expected and returned by the operations

(3) Security and other policies governing these interactions

The above mentioned taxonomy could be in the form of WSDLs for each service area with associated XSDs for the data to be exchanged. Some of the benefits of this approach would be:

- Clear and transparent information on the operations for business interactions and the data required for these interactions
- Easier and faster implementation of interoperability between the players affected
- It allows for innovative new players to enter a domain faster and offer new propositions and services to customers.

Though the use case described above refers to one industry vertical, the issues highlighted are present in several other verticals e.g. insurance and standardization of interactions within enterprises in a vertical could lead to far greater and faster adoption of the web being used as a platform to conduct business.

Specifications affected

None of the existing specifications are impacted.

Recommendations for W3C

In order to make the web an ubiquitous platform for conducting business, an initiative from W3C to address business semantics across different verticals would be very useful and timely. While there are certain initiatives of this nature in a few verticals e.g. Acord for the insurance domain, they have not gained widespread acceptance. What is perhaps required is the strong backing from an organization like W3C for leading and executing such initiatives.