

WS-Coordination Solutions

by Simon Zambrovski

Table of contents

1 Task.....	2
2 Message interceptor.....	2
3 Context creation.....	2
4 Registration.....	2

1. Task

According to the WS-Coordination specification, the initiator has to create the coordination context by its activation service. Each message send to participant of a business activity includes this coordination context, to identify the activity in a unique way.

2. Message interceptor

There are several ways to allow the creation of the coordination context before the initiator begins its business activity. One way is to let the initiator implement the interfaces defined in the WS-Coordination. This approach is difficult for adopting existing systems to the WS-Coordination.

In this framework another way has been chosen. The client (initiator) is extended in that way, that all messages send by it are intercepted. This interception is always active but works like a transparent proxy on the client system. As soon as client sends a **begin** message and receives a response from the transactional middleware the interceptor becomes opaque. Each message send from the client is intercepted on the client side and send to the proxy service. This call is extended with information needed for the proxy service to relay the request to the target service. This approach guarantees that each message send after the begin message is redirected to the proxy service and than send to the target web service. On the proxy service it is possible to attach the coordination context to the message.

The opaque interceptor on the client side becomes transparent as soon as it receives a response on the **end** message, send to the transaction middleware, identifying the end of the business activity.

3. Context creation

All messages send after the **begin** message and before the **end** message are redirected to the proxy service and than delivered to the target web service. The responses to the business messages are taking the same way - they arrives on the proxy service and are then delivered to the client.

As soon the coordination middleware receives a **begin** message it invoke a `createCoordinationContext()` method on the Activation service. The created context is then returned and stored in the proxy service. The proxy service provides some additional information to the client side interceptor. This information (proxy client id number) is then used by the client interceptor to identify itself. All calls coming from the same client interceptor are extended with the coordination context hold by the proxy service.

4. Registration

The first business message with corresponding coordination context supplied in SOAP-header reaches the participant web service. Another interceptor catches the message on the participant side and tries to find an activity corresponding to the coordination context in the participant coordination service. If no activity can be found, it initialises the Register operation to specified Registration service. (Alternative it can use own Registration service, and import the activity, so the Registration service will contact the Registration service of the initiator and act itself as participant (subordinated coordinator)). The Registration message must contain enough information for the Registration service to determine which activity the registration requester wants to participate.

The WS-Coordination specification omits this information, so the registration message is extended with the coordination context identifier.

During the registration the participant and initiator coordination services exchange the coordination protocol endpoint references. After this step the coordination conversation is finished. It prepares the communication parties for further interactions providing them enough information for communication along the established logical connection between the coordination protocol endpoints.