CHAPTER 4

EXPERIMENTAL RESULTS

4.1 The Implementation Environment

Considering the design in previous chapter, the implementation focuses mainly on the first two cases, where the flow on the sending side knows in advance that there is a particular instance waiting for a message on the receiving side. The implementation has been carried out in the following environments:

4.1.1 The BPEL Enactment Environment

The workflow enactment environment was Oracle BPEL process manager (version 2.0) [3] enactment environment. Oracle BPEL process manager correlates using WS Addressing and that effects the implementation of the prototype system.

4.1.2 The Web Service Implementation

The Gateway Service has been implemented to work as a web service deployed with Apache's Axis [10] which runs on the Tomcat [11] environment which is also from Apache.

4.2 The Experiment

4.2.1 The Sample case: the online tendering service

As also depicted in Figure 4-1, an online tendering service has the following characteristic:

1. The tendering document must first be prepared and approved by both the finance department and the legal department.
2. The final tendering document is obtained.
3. The invitations to enter the bid are sent to some suppliers
4. Then the tender notice is also published on the web

5. Assuming that the suppliers have all registered to the service in advance, they fill in the quote and other details for the tendering, their authenticated ID will also be submitted along with the bid.
6. Any member of the evaluating committee on the client side will not be able to access the bidding information before the deadline is reached.
7. The committee evaluates bids and notifies the supplier of the winning bid.
8. The result of the tendering will also be published on the web.

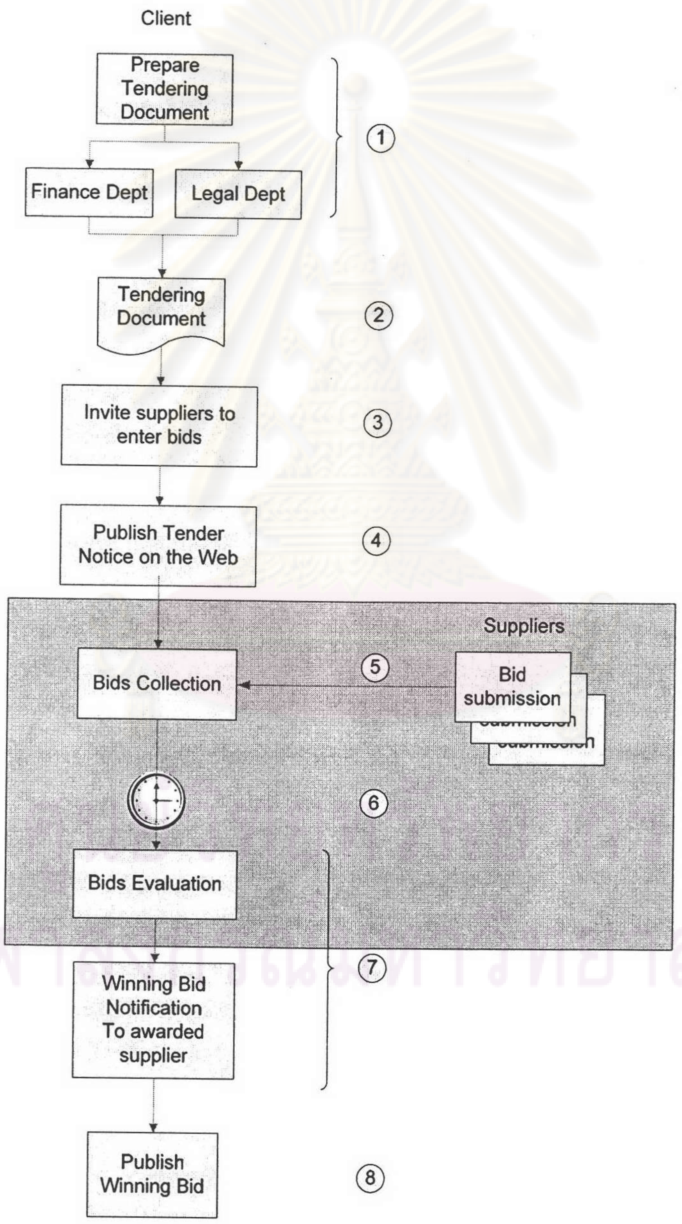
Client
Prepare Tendering Document
Finance Dept
Legal Dept
1
Tendering Document
2
Invite suppliers to enter bids
3
Publish Tender Notice on the Web
4
Suppliers
Bids Collection
5
Bid submission
6
Bids Evaluation
7
Winning Bid Notification To awarded supplier
Publish Winning Bid
8

Figure 4-1 The tendering/bidding process flow

In the case of the tendering service, there are two main flows, the client and the bidders. Considering the client flow structure a problem could occur if the bid is submitted while the flow is still executing the process of publishing the notice to the web. This flow can benefit from the buffering feature of the Gateway Service as the submitted bids can be buffered until the flow is ready. Also the flow could wait until the bid deadline is reached, and notify the Gateway Service to fire back all the buffered messages at the same time so that no bid is accessed illegally before the deadline. The highlighted part of the flow could be modified to use the Gateway Service as in Figure 4-2.

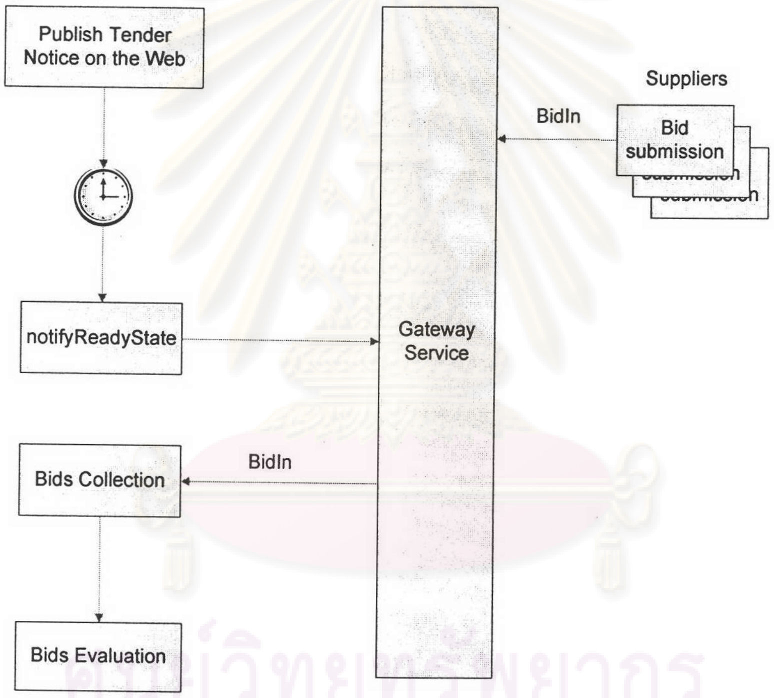

Figure 4-2 The modified tendering service

4.2.2 The Experiment Condition

The problematic subflow (from 2 – 8) has been tested by extending the process of publishing the tendering notice to the time length of 15 minutes, and having a bid arrive while the publishing is still in progress.

4.2.3 The BPEL process definition

The related flows in the tendering service are as followed:

1. The FlowID registration flow – the method registerFlowID of the Gateway Service is a web service and can be called over any web service invocation method. However if called using a BPEL flow, it would look like the following diagram (Figure 4-3).

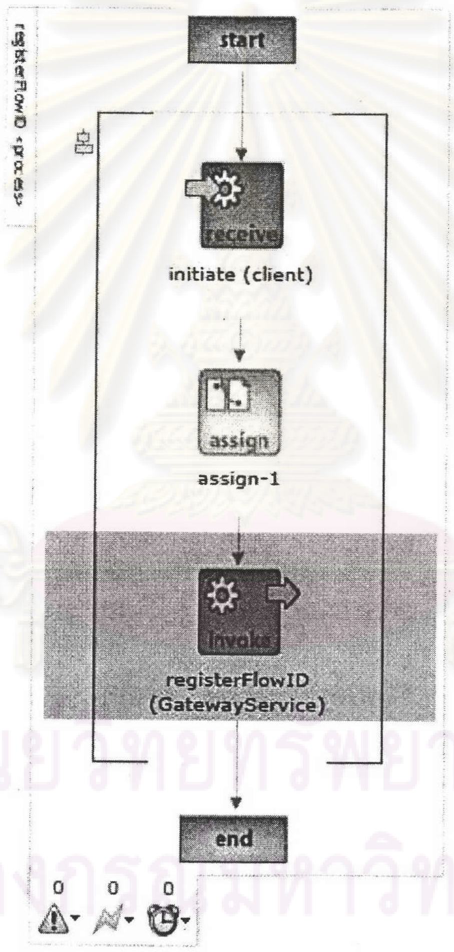

Figure 4-3 The flowID resgistration BPEL process definition diagram

The process of invoking the registeringFlowID is the one related to the Gateway Service. The highlighted part is related to the following code (Figure 4-4).

```
<invoke name="invoke-1"
        partnerLink="GatewayService"
        portType="services:GatewayService"
        operation="registerFlowID"
```

Figure 4-4 The code snippet relating to the registerFlowID method

2. The second process is the bidder side of the E-Tendering. The flow diagram is as followed (Figure 4-5):

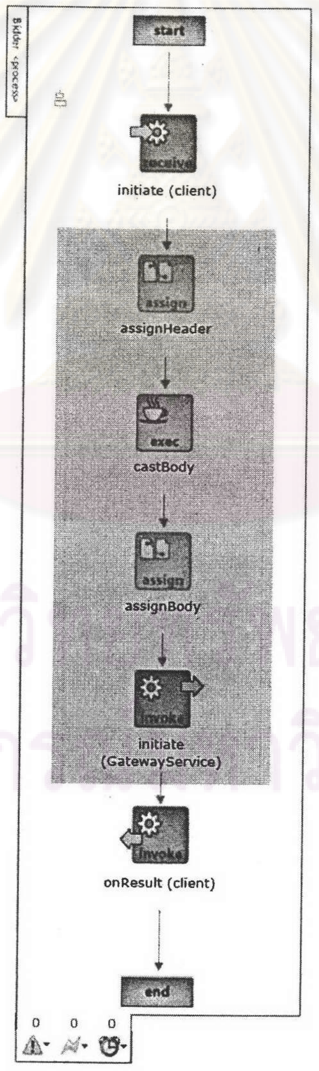

Figure 4-5 The Bidder BPEL process definition diagram

The parts that are related to the gateway service are highlighted. From the process of assigning the MessageHeader, casting the MessageBody to string, and assigning the MessageBody. Then calling the method <invoke> of theGateway Service.

a. The assignHeader activity – this activity basically assign values to each of the mHeader part of the variable gatewayIn of the GatewayMessage format. The code is as depicted in the Figure 4- 6.

```
<assign name="assignHeader">
  <copy>
     <from expression="'1234'">
    </from>
    <to variable="gatewayIn" part="payload"

query="/gateway:gatewayMessage/gateway:MHeader/gateway:jointFlowID"/>
  </copy>
  <copy>
    <from variable="input" part="payload" query="/tns:BidIn/tns:eventID">
    </from>
    <to variable="gatewayIn" part="payload"

query="/gateway:gatewayMessage/gateway:MHeader/gateway:instanceID"/>
  </copy>
  <copy>
    <from expression="'http://pomegranate:9700/orabpel/default/Bidder'">
    </from>
    <to variable="gatewayIn" part="payload"

query="/gateway:gatewayMessage/gateway:MHeader/gateway:callbackURL"/>
  </copy>
  <copy>
    <from expression="'http://pomegranate:9700/orabpel/default/E-
Tendering'">
    </from>
    <to variable="gatewayIn" part="payload"

query="/gateway:gatewayMessage/gateway:MHeader/gateway:serviceURL"/>
  </copy>
  <copy>
    <from expression="'http://acm.org/samples'">
    </from>
    <to variable="gatewayIn" part="payload"

query="/gateway:gatewayMessage/gateway:MHeader/gateway:serviceNameSpace"/
>
  </copy>
  <copy>
    <from expression="'E-Tendering'">
    </from>
    <to variable="gatewayIn" part="payload"

query="/gateway:gatewayMessage/gateway:MHeader/gateway:serviceName"/>
  </copy>
  <copy>
    <from expression="'E-Tendering'">
    </from>
    <to variable="gatewayIn" part="payload"
      query="/gateway:gatewayMessage/gateway:MHeader/gateway:portType"/>
  </copy>
  <copy>
    <from expression="'requestService'">
    </from>
    <to variable="gatewayIn" part="payload"

query="/gateway:gatewayMessage/gateway:MHeader/gateway:soapAction"/>
  </copy>
</assign>
```

Figure 4-6 The code snippet relating to the assignHeader activity

b. The castBody activity – this part of the code casts the body from XML type to plain text. The process definition is as depicted in Figure 4-7.

```
<bpelx:exec xmlns:bpelx="http://schemas.oracle.com/bpel/extension"
  language="java" version="1.4" name="castBody">
  <![CDATA[
       // Java code snippet
  Element  eventID =
      (Element)
getVariableData("input","payload","/tns:BidIn/tns:eventID");
  Element  biddersID =
      (Element)
getVariableData("input","payload","/tns:BidIn/tns:biddersID");
  Element  offeredQuote =
      (Element)
getVariableData("input","payload","/tns:BidIn/tns:offeredQuote");
  Element  biddersEmail =
      (Element)
getVariableData("input","payload","/tns:BidIn/tns:biddersEmail");
  Element  biddersName =
      (Element)
getVariableData("input","payload","/tns:BidIn/tns:biddersName");
  Element  details =
      (Element)
getVariableData("input","payload","/tns:BidIn/tns:details");

  String output = "<BidIn>";
  output+="<eventID xmlns=\"http://acm.org/samples\">"
      +eventID.getNodeValue()+"</eventID>";
  output+="<biddersID xmlns=\"http://acm.org/samples\">"
      +biddersID.getNodeValue()+"</biddersID>";
  output+="<offeredQuote xmlns=\"http://acm.org/samples\">"
      +offeredQuote.getNodeValue()+"</offeredQuote>";
  output+="<biddersEmail xmlns=\"http://acm.org/samples\">"
      +biddersEmail.getNodeValue()+"</biddersEmail>";
  output+="<biddersName xmlns=\"http://acm.org/samples\">"
      +biddersName.getNodeValue()+"</biddersName>";
  output+="<details xmlns=\"http://acm.org/samples\">"
      +details.getNodeValue()+"</details></BidIn>";
  setVariableData("bidInString",output);
    ]]>
</bpelx:exec>
```

Figure 4-7 The code snippet relating to the castBody activity

c. The castBody activity – this activity assign the string bidInString to the mBody part of the variable gatewayIn of the GatewayMessage format (Figure 4-8).

```
<assign name="assignBody"><copy>
    <from variable="bidInString">
    </from>
    <to variable="gatewayIn"
        part="payload"
        query="/gatewayasync:gatewayMessage/gatewayasync:MBody"/>
  </copy>
</assign>
```

Figure 4-8 The code snippet relating to the assignBody activity

d. The invoke activity – this activity invoke the "initiate" operation of the Gateway Service (Figure 4-9).

```
<invoke name="invoke-1"
        partnerLink="GatewayService"
        portType="services:GatewayService"
        operation="initiate"
        inputVariable="gatewayIn"/>
```

Figure 4-9 The code snippet relating to the invoke activity (initiate)

3. The last flow is the main E-Tendering Service. The overview of the flow is as followed:

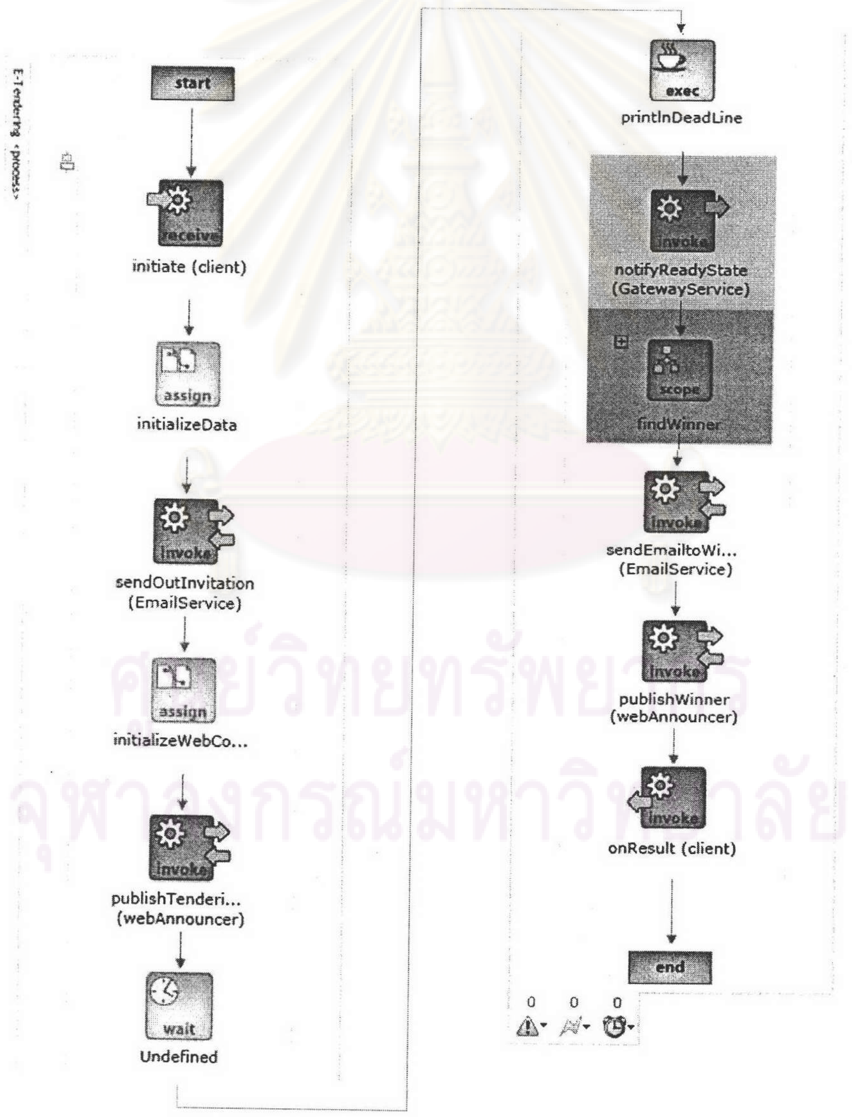

Figure 4-10 The E-Tendering BPEL process definition diagram

The parts that are related to the GatewayService are as depicted in Figure 4-10.

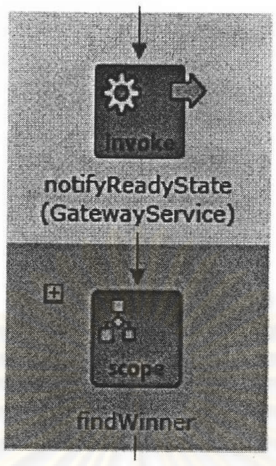

Figure 4-11 The E-Tendering BPEL process definition diagram (2)

Note that the relevant parts are the notifyReadyState and the scope called findWinner.

a. The notifyReadyState is corresponding to the following code:

```
<invoke name="invoke-1"
        partnerLink="GatewayService"
        portType="services:GatewayService"
        operation="notifyReadyState"
        inputVariable="notifyIn"/>
```

Figure 4-12 The code snippet relating to the invoke activity (notifyReadyState)

b. And the scope can be expanded into the following flow logic:

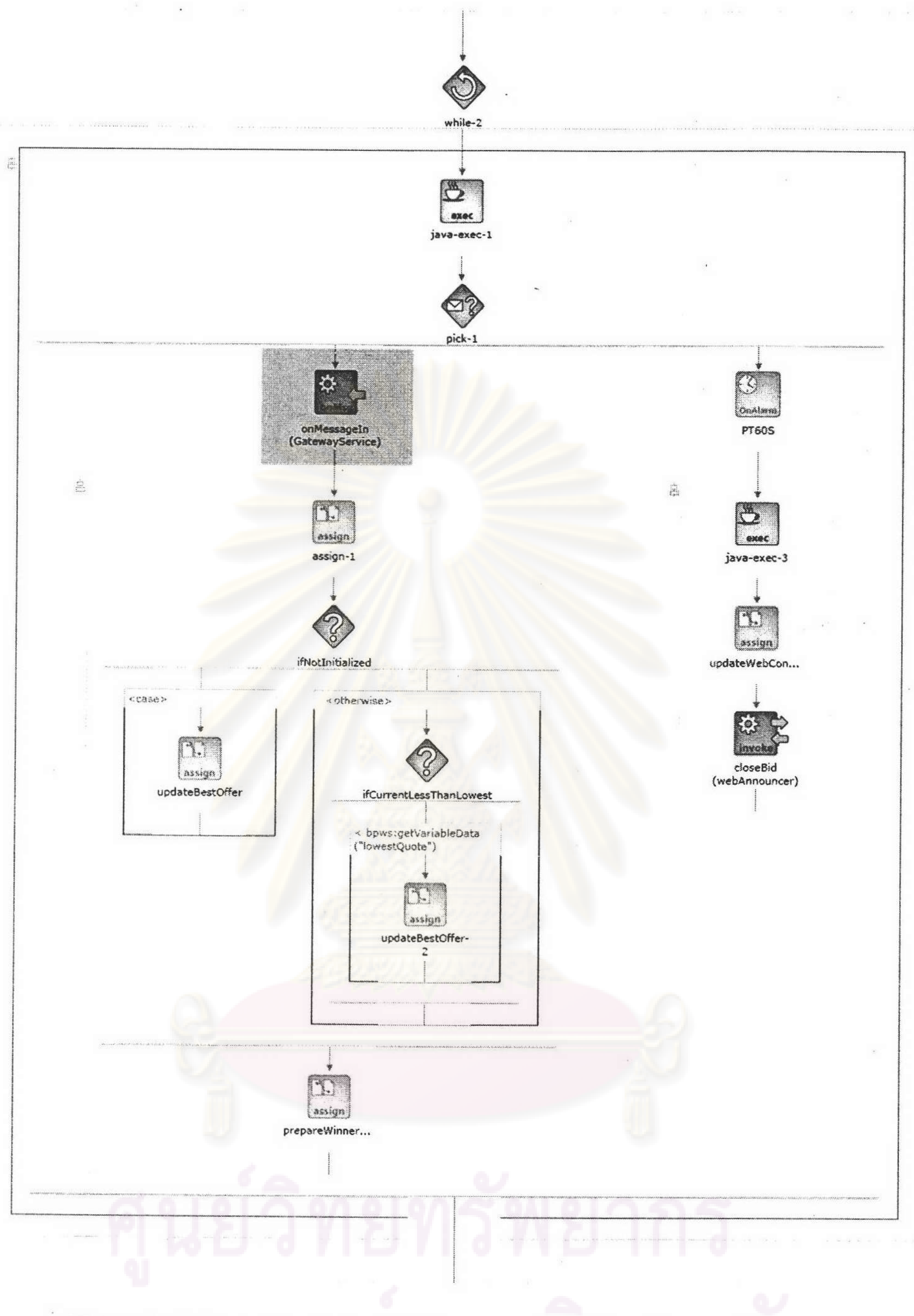

Figure 4-13 The E-Tendering BPEL process definition diagram (3)

The Gateway Service related part is as highlighted, it is the onMsg activity that listens to the gateway when the messages are sent in. The BPEL code is as followed:

```
<onMessage
  partnerLink="GatewayService"
  portType="services:GatewayServiceCallback"
  operation="onMessageIn"
  variable="messageIn">
```

Figure 4-14 The code snippet relating to the onMessage

The rest of the flow is the loops that compare each bid to find the lowest value.

Note: The full BPEL code of the E-Tendering Service is included in Appendix B.