CHAPTER 5

CONCLUSION AND DISCUSSION

5.1 Conclusion

This thesis proposes a design of a framework in detecting and handling synchronous exceptions in sending and receiving message which is the main part in any workflow collaboration. The design is based on the idea of a Gateway Service, which buffers incoming messages, in the exceptional case of the messages arriving before the receiving flow is ready. In order to avoid centralization, each organization has its own Gateway Service responsible to the service it is providing. The Gateway Service uses correlations data to deliver message to the correct recipients. Moreover, the Gateway Service redirects the service request from one BPEL flow to the location of corresponding Gateway Service where the requested service resides.

5.2 Discussion

The designed framework is able to perform the redirecting and buffering task well. However, the real business-related complex data type has to be type-casted to string in order to make the service generic, as any message being sent through SOAP has to be of a known type. Some BPEL enactment services, including the Oracle BPEL Process Manager, have built-in functions which support the parsing and casting on both directions. The service buffers the incoming messages on the condition that the message bears the valid jointFlowID and correlates by using the instanceID and map it to the operation name in case of branched flows with multiple entry points. Although the implementation focuses mainly on cases where messages are sent to a service that has already been instantiated, little modification can be add to extend supports to cases where incoming messages fire off new instances. The service can be well applied to loosely coupled collaboration where most messages are sent in asynchronous pattern. The service itself has been implemented on an event-based web service which spawns new threads at each incoming request, and thus helps dealing with cases where high volume of incoming messages are directed to the same instance of a service simultaneously. Although BPEL has some ability to handle event based procedure call, it is unclear from the specification [4] what to be done in case of an event that is raised while still processing another one. By the use of the proposed design, simultaneous calls could be processed sequentially. Also as there different methods of correlation employed by different BPEL enactment engines, for instance, some BPEL enactment engine uses WS-Addressing's conversation ID and some do not. If any collaboration among these BPEL engine are to occur, compatibility issue will have to be resolved. By using the method of matching up the instanceID, with some modification in the design this issue could be resolved.

The design could be applied to real cases to help facilitate message passing between flow collaboration. For example, in the study of the book closing flow of the Thailand Security Depository Company, the following cases could benefit from the application of the design:

- Case 1: The after hour service when the deadline for sending in the dividend payment information has been passed, the Gateway service could be applied so that the data could be retrieved in the next morning. As currently a massager must deliver the information in person when the deadline has passed.
- Case 2: In certificate printing the service can be applied to be able to take the request outside the office hours and print the certificate at day time when people operating the machine are ready. Also the service can buffer incoming when the printer crashes or need to stop temporarily.
- Case 3: Also in the case of cheque printing, if there is a problem with the printer or need to stop, the proposed service could be employed.
- Case 4: In crediting the dividend to the share holders' account, the service could buffer the request and release it to the bank when it's open.

5.3 Future Works

1. Security measures, for instance, WS-Security could be added to ensure the safety of the data transfer.
2. The designed framework could act as an access controller for a certain web service once an authentication method is added to the messages.

3. The Service could be modified to be able to temporarily pause or resume buffering in order to serve other purpose such as in case where the BPEL flow is temporarily not functioning.