

Analysis and Design

5.1 Introduction

In previous chapter it has been discussed the proposed approach with reference to users, input, output, process and features of the system. There were two major parts of the process namely SOAP message optimization process and data clustering process. In this chapter it will be discussed about the requirement analysis and high level design of the project. First it will mention the set of high level requirements that has been identified. The design section will discuss about four major modules of the system which are corresponding to four major functionalities of the system. It will also discuss about the multi agent system in high level and then about the ontology.

5.2 Functional requirements

During the analysis phase of this research project the following set of high level requirements has been identified.

- Capture the SOAP message that is being transferred on web service application
- Analyze the SOAP message and search for pre-identified patterns
- Automatically identify content patterns of new SOAP messages.
- Automatically identify new tags added to the existing messages and by doing that automatically update the existing templates.
- Automatically identify new context (data changes of the message) and update the existing templates.
- Automatically distribute the recognized patterns to other nodes of the system.
- To maintain the consistency of the knowledge, store the patterns on the ontology and then use it when system start again.
- To maintain the consistency of the knowledge, at the startup of each node (either client or server) inquire other nodes to find new knowledge of templates that is not stored in its own ontology.

5.3 Analysis

There are four major processes that can be identified in this approach to optimize the SOAP message. First one is to recognize common content pattern of SOAP message and store them as static content of the message. Then it has to distribute these patterns among other parties involved in the communication. As the major process this system requires to identify correct patterns from that collection and apply them to optimize the SOAP message at the sender end. Then at the receiver end it will re-build the SOAP message using the same content pattern.

According to the above analysis four major modules of the system has been identified which are mentioned bellow.

- Message organizer
- Pattern Recognizer
- Pattern Distributor
- Message Re-builder

The high level representation of the interaction and message communication of above modules are represented on the figure 5.1.

5.3.1 Message Organizer

This module processes the message being transmitted, to identify whether there is a matching pre-recognized pattern associated with the message. If it is not matching with any pre-built pattern, then the message will be queued as a candidate to recognize patterns. This module will take the SOAP message as the input and output either same SOAP message (if there are no matching patterns) or optimized message. This module also output data to the Pattern Recognizer module to use in pattern recognizing process. Figure 5.2 shows the data flow diagram of the message sender. The message organizing process can be seen on this diagram.

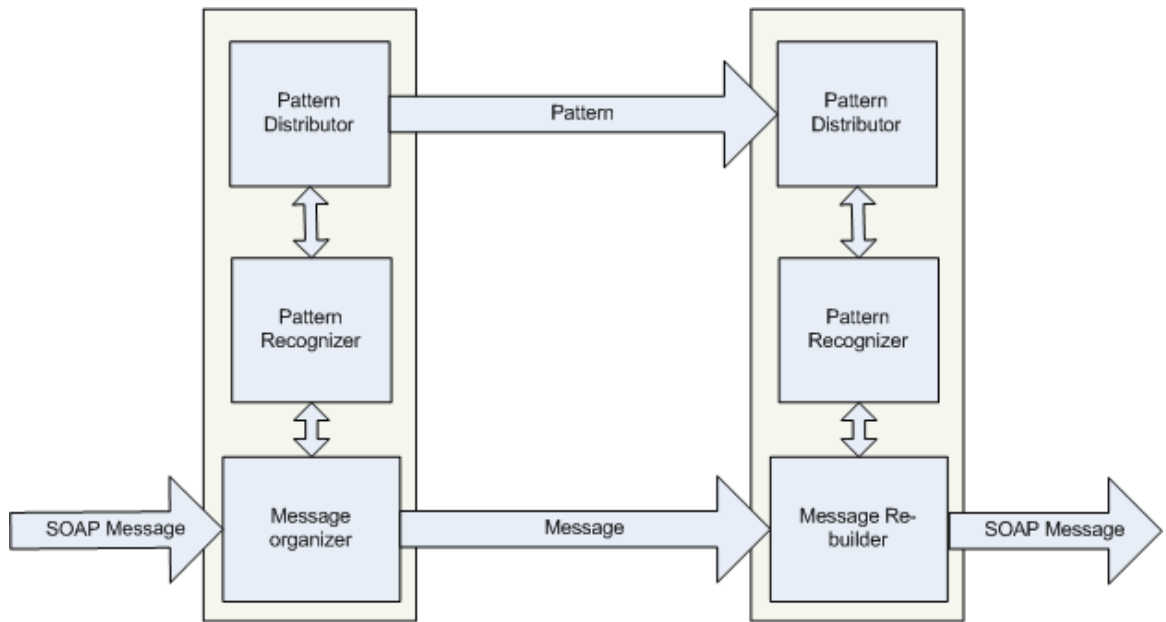


Figure 5.1 : Message flow diagram of core modules

5.3.2 Pattern Recognizer

This module will recognize common patterns on message being transferred and then identify the static content of messages using those patterns. Then it will build the static content format that is to be distributed and output this static content document to distributor module as well as the message organizer module. Multi Agent Negotiation method will be used in this module to identify the optimal content patterns. This module will first process the XML document to identify the basic data structures of the XML document. Then it will collect the content data against each data structure by processing their instances on the XML document. It will continue on collecting data using all the messages of same type passing through the web-service. This module will periodically start the clustering process and identify new context patterns. It will also use the Genetic Algorithms within the cluster agents to pick the most suitable set of text to be used to generate the cluster pattern. Then those patterns will be used as the static content of data and will be reduced from message while transferring them.

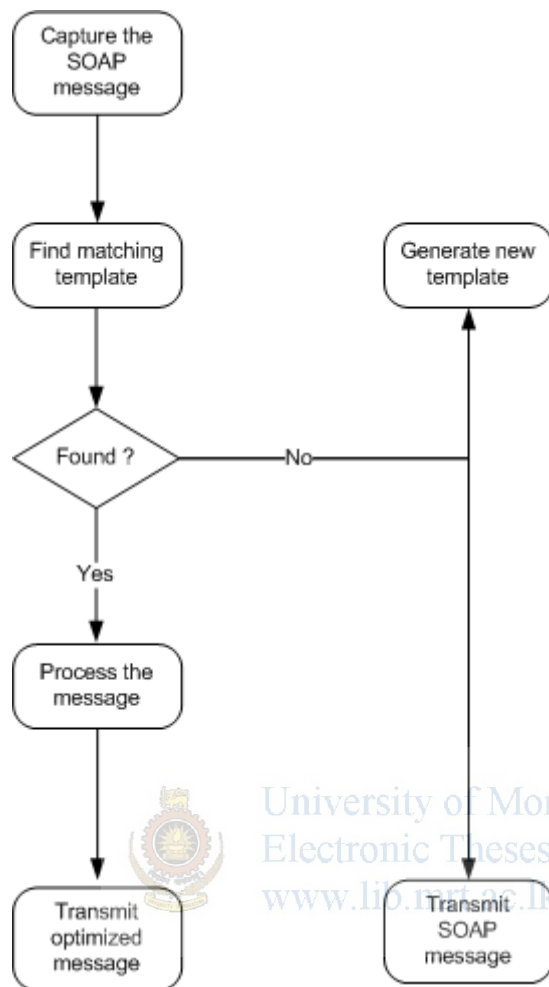


Figure 5.2 : Data flow diagram of message sender

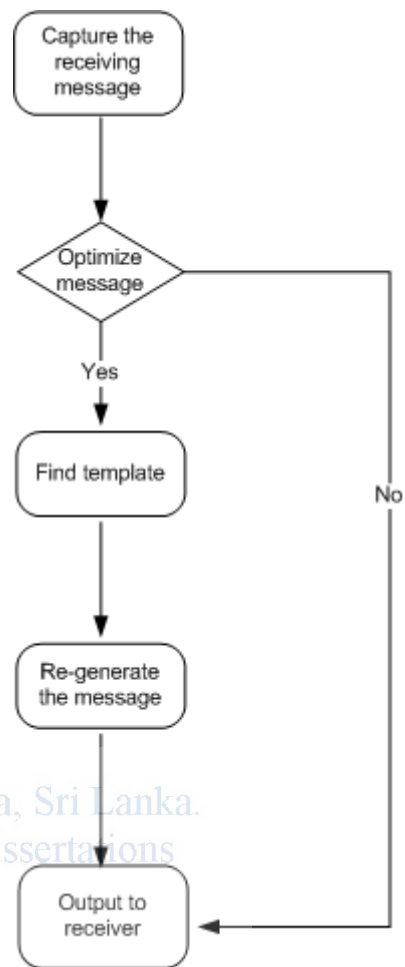


Figure 5.3 : Data flow diagram of message receiver

5.3.3 Pattern Distributor

This module will distribute the pattern documents within the cluster of computers. It will also cache the pattern documents so that when a web-service client is started and connected to the web-service, the client will be able to retrieve the unknown patterns from the server. The pattern distribution is done using the Agent messages. The distributed agent group facility on MADKit[18] environment will be used for this purpose. The two MADKit kernels running on the web-service application server and client module are connected via the communicator toolkit provided on MADKit

environment. Then a distributed group is created and Pattern Distributor agents are registered on that group.

5.3.4 Message Re-builder

This module will re-build the SOAP message at the receiver end. It will use the predefined pattern document and the transmitted “Optimized message”. It will identify the correct static content pattern to use by using the unique key in the optimized message. Then it will insert the correct static content into message. The optimized message is the input to this module and complete SOAP message will be the output from this module. Figure 5.3 shows the data flow diagram of the message receiver. The message re-builder process is shown there.

5.4 Design

Since it has been decided to use multi agent systems to build the systems, agent oriented design methodologies are used in the design phase. In the SOAP optimization process the multi agent negotiation methodology is applied mainly in two modules. That is when identifying the applicable pattern template in message organizer module and on data clustering process in pattern recognizer module.

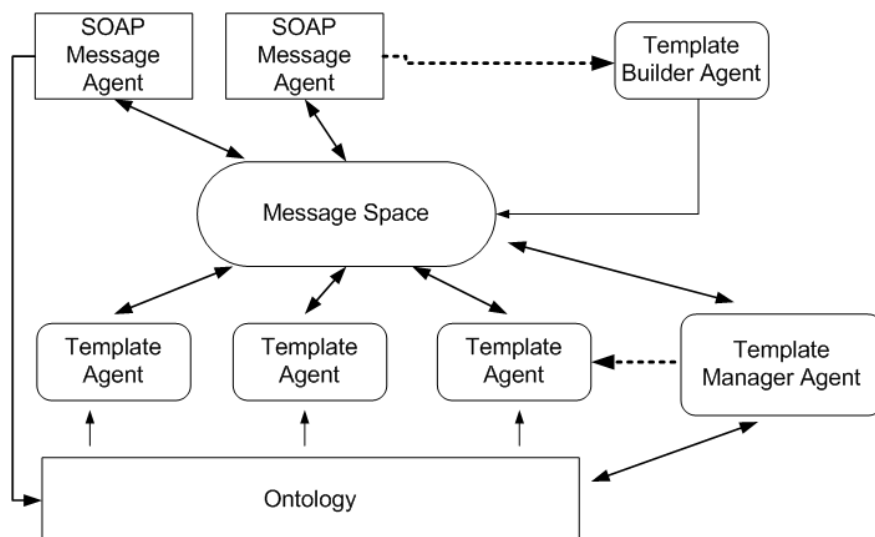


Figure 5.4 : Architecture diagram – Message optimization process

Inside the message optimizer module it will search for pre-identified context pattern for a SOAP message when it is due to transfer through internet. There can be multiple patterns applicable for a given message and it is required to find the suitable pattern that can be used to optimize the given message. For that it will generate a SOAP message agent as the request agent and Template Agents for each pre-identified template. The architecture diagram for this process is in figure 5.4. Template agents will negotiate with SOAP message agent to identify the correct template. This process is discussed in more detail in the next chapter.

During the pattern recognition process it is required to identify the common pattern of text contained within the xml tags and for that system will collect the data of same type in to a list and then it will cluster them. Then system will find the common pattern of each cluster as the content pattern for the text contained in that cluster. For this clustering process system will use the agent negotiation process. The agent architecture diagram for data clustering process is mentioned on the figure 5.5. In the clustering process it will create Cluster Agents and Text Agents. Each Text Agent's target is to find a suitable cluster and each Cluster Agent's target is to accept only the text that increases the energy level of the cluster. The clustering process is discussed in detailed in next chapter.

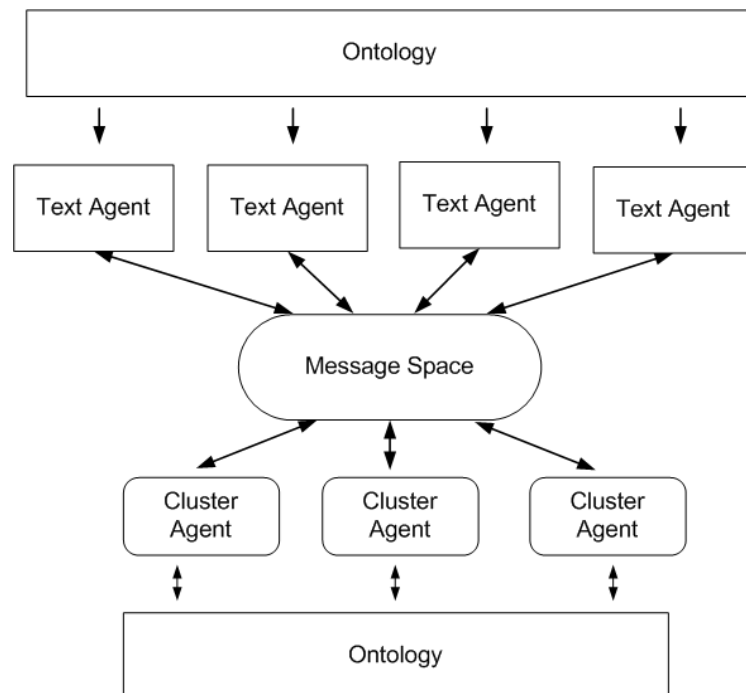


Figure 5.5 : Architecture diagram – data clustering

5.5 Ontology

Ontology plays a main role of the Multi Agent System. As the hard disk is the permanent storage of the computer system, ontology is the permanent storage of the Multi Agent System. The ontology is used to store the operational knowledge gathered through the multi agent communication and also business rules that are specific to each individual agents. The pre recognized patterns will be stored on the ontology. Then when agent starts again it will use those stored patterns as its predefined knowledge.

5.6 Summary

In this chapter it has been discussed about the analysis and design phase of the project. During the analysis and design phase four major modules has been identified which corresponds to four major functionalities of the system. Then the agent oriented design of the system has been discussed about. The agent oriented design has focused on two major areas. Those are SOAP message optimization process and data clustering process. Finally this chapter has discussed about the ontology. The next chapter will discuss about the implementation process of the system.

