

## References

- [1] Advantages & Disadvantages of Webservices. [Online]. Available. <http://social.msdn.microsoft.com/Forums/en-US/asmxandxml/thread/435f43a9-ee17-4700-8c9d-d9c3ba57b5ef>
- [2] Apache XMLBEANS. [Online]. Available. <http://xmlbeans.apache.org/>.
- [3] Christian Werner AND Carsten Buschmann AND Stefan Fischer, (2008) *Efficient Encodings for Web Service Messages*, in *Web Services Research and Practices*, pp 1 - 22
- [4] D. Megginson and D. Brownell, Maintainers SAX 2.0.1: The Simple API for XML. [Online]. Available. <http://www.saxproject.org>.
- [5] Distributed Applications: A Vital Tool for the Modern Enterprise. [Online]. Available. [http://www.progress.com/progress\\_software/products/docs/idc\\_executive\\_briefing.pdf](http://www.progress.com/progress_software/products/docs/idc_executive_briefing.pdf)
- [6] E. Lumer and B. Faieta. (1994). *Diversity and Adaptation in Populations of Clustering Ants*, Proceedings of the 3rd Int. Conf. on Simulation of Adaptive Behaviour: From Animals to Animats, 3, pp 501 - 508
- [7] Ernst von Glasersfeld (1995), *Radical Constructivism: A Way of Knowing and Learning*, The Falmer Press - London & Washington. ISBN 0 7507 0387 3
- [8] G.Andrighetto, P. Assogna. (2004) *Why we understand each other: the role of the context*, Centro 16th Conference on Advanced Information Systems Engineering, Riga, Latvia, 3, pp 293 - 296
- [9] George Rzevski, Peter Skobelev, Igor Minakov, Semen Volman. (2007). *Dynamic Pattern Discovery using Multi-Agent Technology*, Proceedings of the 6th WSEAS International Conference on Telecommunications and Informatics, pp 75 - 81.
- [10] Giambattista Vico. [Online]. Available. [http://en.wikipedia.org/wiki/Giambattista\\_Vico](http://en.wikipedia.org/wiki/Giambattista_Vico)
- [11] GlassFish Community. [Online]. Available. <https://glassfish.dev.java.net/>
- [12] GlassFish Metro JAX-WS. [Online]. Available. <https://jax-ws.dev.java.net/>

- [13] H. Liefke and D. Suci, (2000) *XMill: an efficient compressor for XML data*, Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD2000), pp 153 - 164.
- [14] Indiana University, Extreme! Computing Lab. Grid Web Services. [Online]. Available. <http://www.extreme.indiana.edu/xgws/>.
- [15] Introduction to StAX. [Online]. Available. <http://today.java.net/pub/a/today/2006/07/20/introduction-to-stax.html>
- [16] Is it an Agent, or just a Program: A Taxonomy for Autonomous Agents. [Online]. Available. <http://www.msci.memphis.edu/~franklin/AgentProg.html>.
- [17] JGAP—Java Genetic Algorithms Package. [Online]. Available <http://jgap.sourceforge.net/>
- [18] MADKit. [Online]. Available. <http://www.madkit.net/>
- [19] N. Abu-Ghazaleh, M. J. Lewis, and M. Govindaraju. (2004), *Differential Serialization for Optimized SOAP Performance*. Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing (HPDC-13), pp 55–64
- [20] Nayef AbuGhazaleh and Michael J. Lewis (2005) *Differential Deserialization for Optimized SOAP Performance*, Proceedings of ACM/IEEE conference on Supercomputing, pp. 21-31.
- [21] P. Deutsch, (1996) GZIP file format specification version 4.3, [Online]. Available. <http://www.ietf.org/rfc/rfc1952.txt>.
- [22] Plotkin JB, Nowak MA. (2000). *Language evolution and information theory*. Journal of Theoretical Biology, 205, pp 147 - 159
- [23] Poess, M., & Floyd, C. (2000). *New TPC benchmarks for decision support and Web commerce*. ACM SIGMOD Record archive, 29, pp 64 - 71
- [24] S.M. Kamruzzaman and Farhana Haider. (2004). *A Hybrid Learning Algorithm For Text Classification*. Proceedings of 3rd International Conference on Electrical & Computer Engineering ICECE 2004. pp 577 - 580
- [25] Sam's String Metrics. [Online]. Available <http://www.dcs.shef.ac.uk/~sam/stringmetrics.html>
- [26] Sangyoon Oh, Mehmet S. Aktas, Marlon Pierce, Geoffrey C. Fox. (2006) *Optimizing Web Service Messaging Performance Using a Context Store for Static Data*. Proceedings of 5th WSEAS International Conference on TELECOMMUNICATIONS and INFORMATICS, 520(269), pp 334 - 342

- [27] Shahram Ghandeharizadeh, Christos Papadopoulos, Min Cai, Runfang Zhou, Parkshit Pol (2005) *NAM : A Network Adaptable Middleware to Enhance Response Time of Web Services*. 11th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems. pp 136 - 150
- [28] Siedlecki, W. and J. Sklansky. (1989) *A note on genetic algorithms for large-scale feature selection*, Pattern Recognition Letters Archive, 10 (5), pp 335 - 347
- [29] Te-Sheng Li. (2006). *Feature selection for classification by using a GA-Based neural network approach*. Journal of the Chinese Institute of Industrial Engineers, 23 (1), pp 55 - 64
- [30] Vizine, A. L., de Castro, L. N., Gudwin, R. R. (2005). *Text Document Classification using Swarm Intelligence*. Proceedings of International Conference on Integration of Knowledge Intensive Multi-Agent Systems, pp 134 - 139
- [31] World Wide Web consortium. Document object model. [Online]. Available. <http://www.w3c.org/DOM>.



## Appendix A

### Evaluation Data

It has been collected data into logs while the evaluation is going on. The collected data are listed on the table A.1. In that table the SC indicates the search criteria for which the data has been retrieved. Then it indicates whether the optimization has been used or not. Y means Yes and N means No. Then it has included the lengths of the messages in bytes and the times on milliseconds

SC	Opt Used	Org. Msg. Length (byte)	Transferred Msg. Len (byte)	Message Transfer Time (ms)	Total Time (ms)
LK	Y	1332	292	921	968
LK	Y	1332	292	610	625
LK	Y	1332	292	656	656
LK	Y	1332	292	765	765
LK	Y	1332	292	593	609
US	Y	307112	79018	2406	2953
US	Y	307112	79018	2390	2812
US	Y	307112	79018	2625	2984
US	Y	307112	79018	2391	2672
US	Y	307112	79018	2344	2641
CA	Y	73631	18770	4250	4313
CA	Y	73631	18770	750	781
CA	Y	73631	18770	610	672
CA	Y	73631	18770	4500	4563
CA	Y	73631	18770	703	860
AU	Y	86083	21378	703	750
AU	Y	86083	21378	1312	1375
AU	Y	86083	21378	750	828
AU	Y	86083	21378	906	969

AU	Y	86083	21378	1438	1594
JP	Y	13503	3261	765	781
JP	Y	13503	3261	406	422
JP	Y	13503	3261	937	937
JP	Y	13503	3261	360	375
JP	Y	13503	3261	1000	1000
LK	N	1332	1332	609	609
LK	N	1332	1332	688	688
LK	N	1332	1332	641	641
LK	N	1332	1332	688	688
LK	N	1332	1332	3594	3594
US	N	307112	307112	7031	7031
US	N	307112	307112	3500	3500
US	N	307112	307112	3297	3297
US	N	307112	307112	3094	3094
US	N	307112	307112	3968	3968
CA	N	73631	73631	4422	4422
CA	N	73631	73631	2187	2187
CA	N	73631	73631	2125	2125
CA	N	73631	73631	2360	2360
CA	N	73631	73631	2203	2203
AU	N	86083	86083	4375	4375
AU	N	86083	86083	1844	1844
AU	N	86083	86083	2515	2515
AU	N	86083	86083	1954	1954
AU	N	86083	86083	3079	3079
JP	N	13503	13503	4547	4547
JP	N	13503	13503	1125	1125
JP	N	13503	13503	844	844
JP	N	13503	13503	7578	7578
JP	N	13503	13503	797	797

Table A.1 : Data collected during evaluation

Comparison of data when the SOAP Optimization tool used and when the SOAP optimization tool is not used is listed on the table A.2.

Search Criteria	Original Message Length (byte)	Optimization used			Optimization not used		
		Transfer message length (byte)	Transfer time (ms)	Total time (ms)	Transfer message length (byte)	Transfer time (ms)	Total time (ms)
LK	1332	292	921	968	1332	609	609
LK	1332	292	610	625	1332	688	688
LK	1332	292	656	656	1332	641	641
LK	1332	292	765	765	1332	688	688
LK	1332	292	593	609	1332	3594	3594
US	307112	79018	2406	2953	307112	7031	7031
US	307112	79018	2390	2812	307112	3500	3500
US	307112	79018	2625	2984	307112	3297	3297
US	307112	79018	2391	2672	307112	3094	3094
US	307112	79018	2344	2641	307112	3968	3968
CA	73631	18770	4250	4313	73631	4422	4422
CA	73631	18770	750	781	73631	2187	2187
CA	73631	18770	610	672	73631	2125	2125
CA	73631	18770	4500	4563	73631	2360	2360
CA	73631	18770	703	860	73631	2203	2203
AU	86083	21378	703	750	86083	4375	4375
AU	86083	21378	1312	1375	86083	1844	1844
AU	86083	21378	750	828	86083	2515	2515
AU	86083	21378	906	969	86083	1954	1954
AU	86083	21378	1438	1594	86083	3079	3079
JP	13503	3261	765	781	13503	4547	4547
JP	13503	3261	406	422	13503	1125	1125
JP	13503	3261	937	937	13503	844	844
JP	13503	3261	360	375	13503	7578	7578
JP	13503	3261	1000	1000	13503	797	797

Table A.2 : Evaluation data comparison

# Appendix B

## Clustering Viewer

The Cluster agents are represented with a circle and text agents are represented with a rectangle on this viewer. Initially Cluster Agents and Text Agents are created. The Cluster patterns of the cluster agents are null. Text Agents broadcast the Membership Request message. At the right side of the screen it is possible to view the messages passed during the data clustering. Figure B.1 shows the Text Agents sending messages requesting for a cluster and Cluster agents fetching relevant messages from message space.

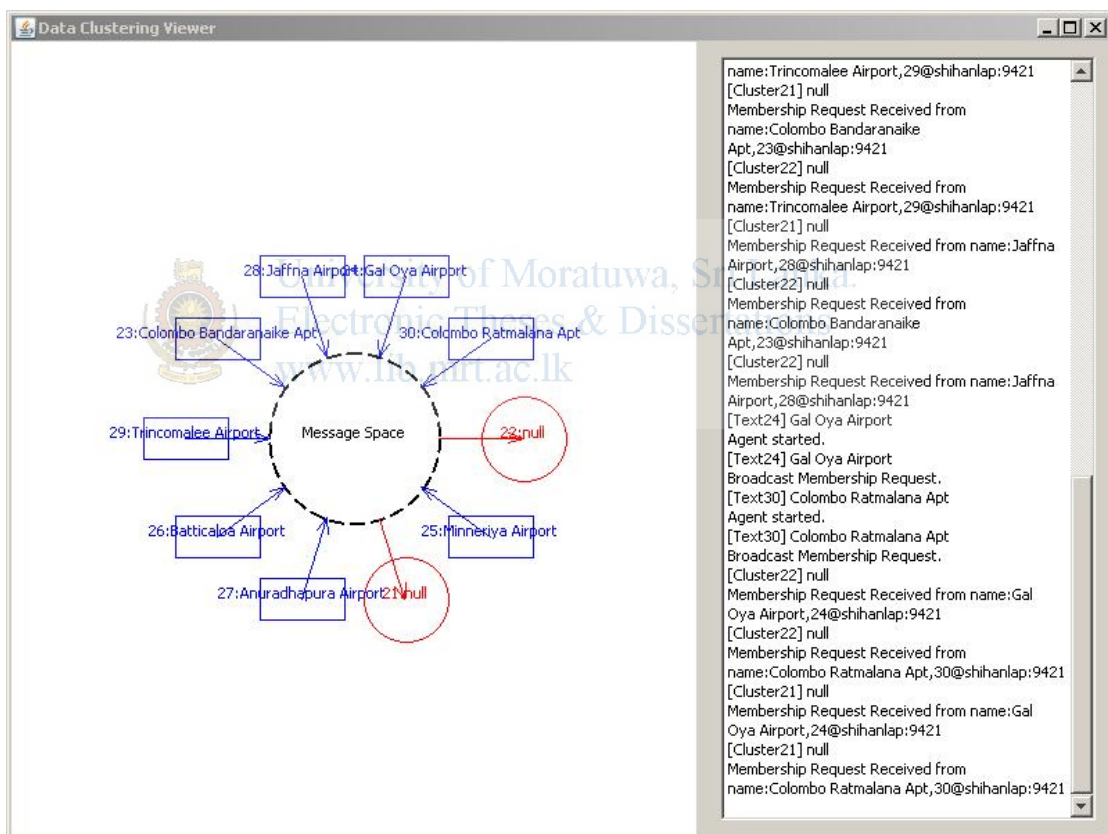


Figure B.1 : Data clustering – Text agents seek for cluster

Figure B.2 shows some of the Cluster agents respond to the cluster membership request message and relevant Text agents fetch the message from the message space.

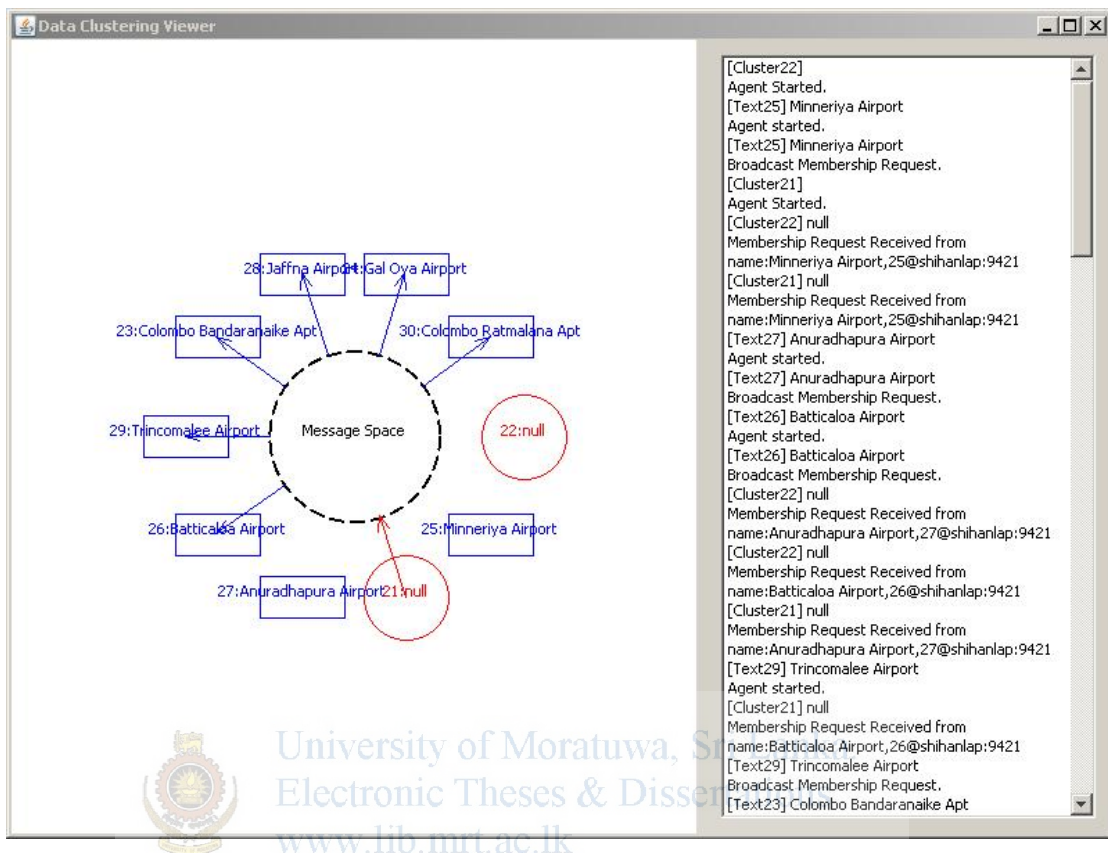


Figure B.2 : Data clustering – A cluster agent response for requests

While the clustering is in progress one cluster has been assigned the pattern on java regular expression format “(.\*) Airport”. That means any string ends with “ Airports” is a member of this cluster. The text agents belonging to that cluster are finished their lifecycle because they have found the cluster. But the cluster agent still exists because there are two more text agents in the system struggling to find a cluster. Please see the figure B.3 for details.



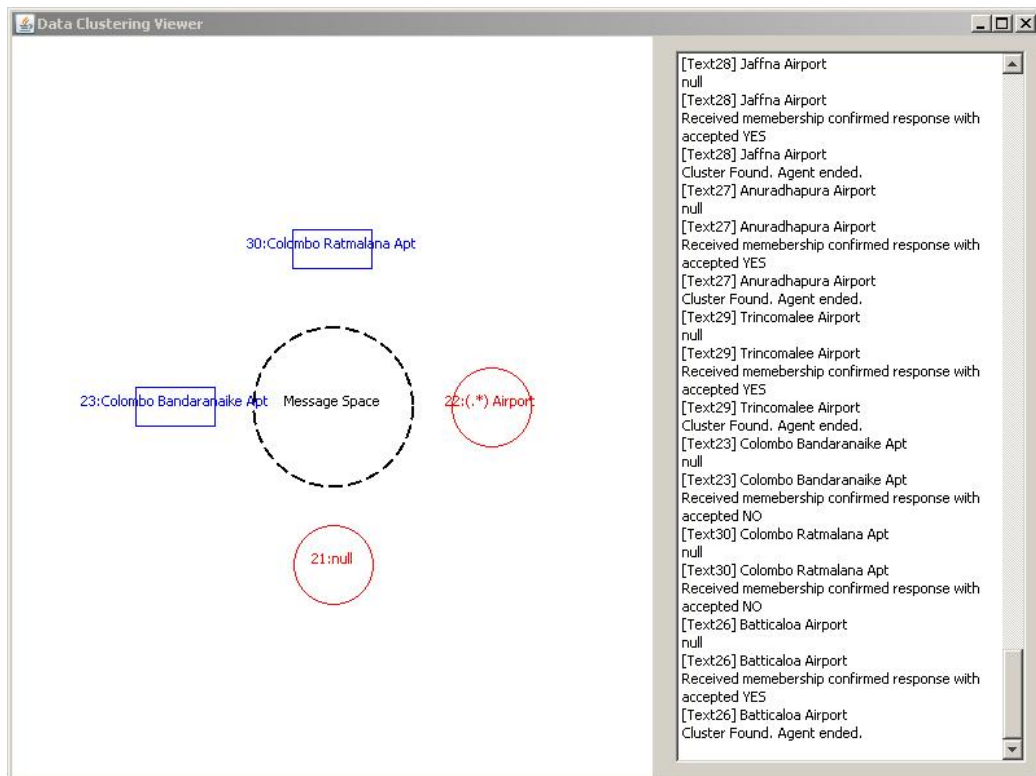


Figure B.3 : Data clustering – One cluster has been performed

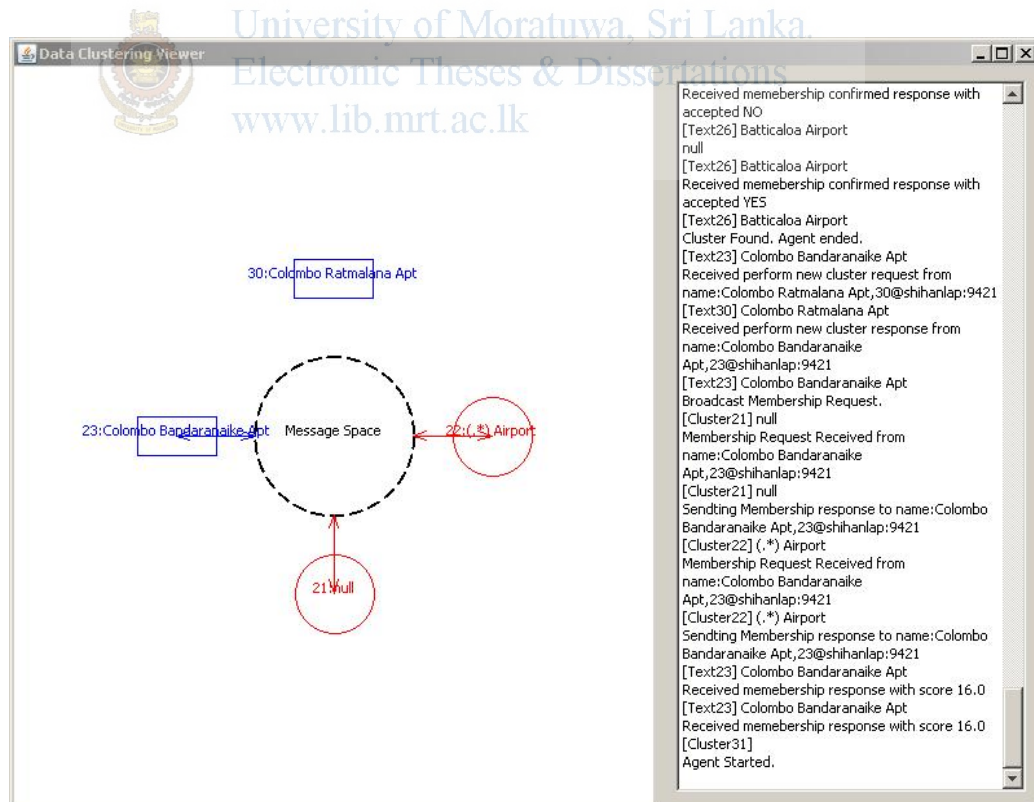


Figure B.4 : Data clustering – Cluster agents response for new requests

But there are two more Text Agents in the environment without having a cluster. Therefore those agents will resend the Membership request message and both Cluster agents in the environment will response for those messages and put their scores in the message space. The figure B.4 indicates this situation.

Finally all the Text agents are clustered into two clusters as follows. The final result is shown on figure B.5.

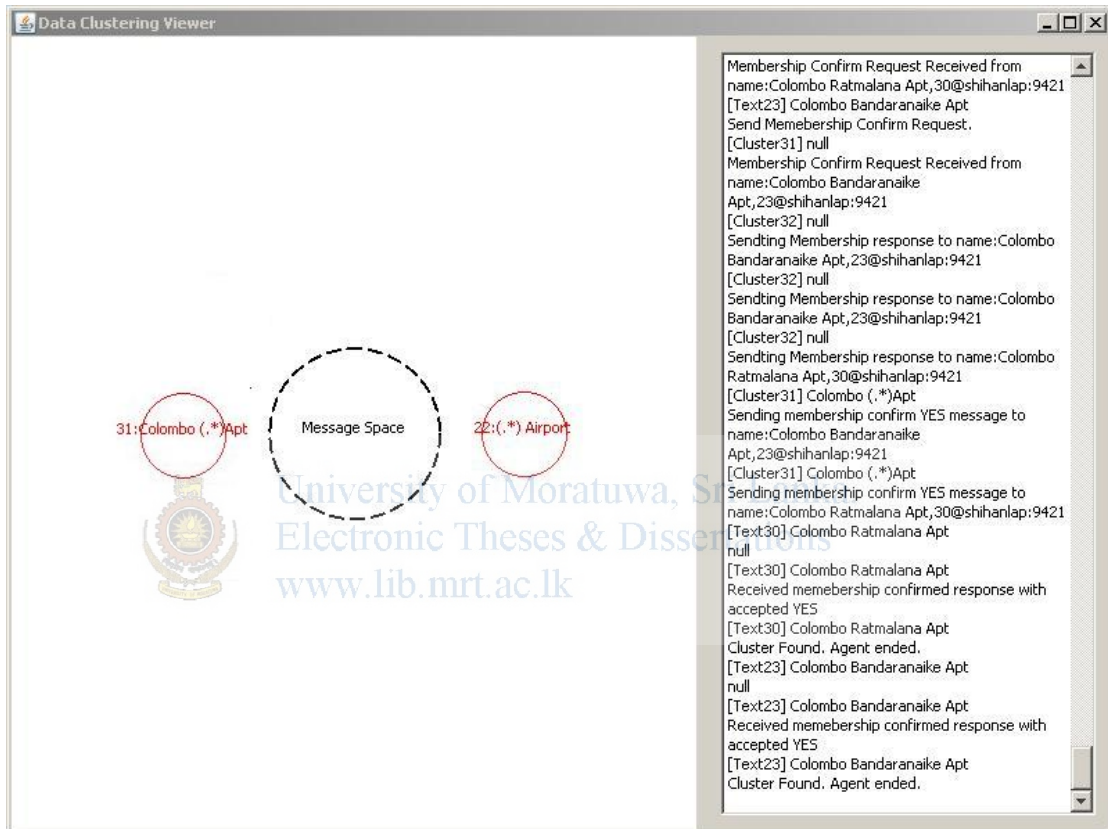


Figure B.5 : Data clustering – Clustering has finished with two clusters.

## Appendix C

### Sample Input and Output

Sample SOAP message which was taken while the evaluation process of the tool is shown in the figure C.1.

```
<ns2:getAirportsResponse xmlns:ns2="http://testservice">
  <return>
    <cityCode>ADP</cityCode>
    <code>ADP</code>
    <countryCode>LK</countryCode>
    <name>Anuradhapura Airport</name>
    <stateCode>SOA</stateCode>
  </return>
  <return>
    <cityCode>BTC</cityCode>
    <code>BTC</code>
    <countryCode>LK</countryCode>
    <name>Batticaloa Airport</name>
    <stateCode>SOA</stateCode>
  </return>
  <return>
    <cityCode>CMB</cityCode>
    <code>CMB</code>
    <countryCode>LK</countryCode>
    <name>Colombo Bandaranaike Apt</name>
    <stateCode>SOA</stateCode>
  </return>
  <return>
    <cityCode>GOY</cityCode>
    <code>GOY</code>
    <countryCode>LK</countryCode>
    <name>Gal Oya Airport</name>
    <stateCode>SOA</stateCode>
  </return>
  <return>
    <cityCode>JAF</cityCode>
    <code>JAF</code>
    <countryCode>LK</countryCode>
    <name>Jaffna Airport</name>
    <stateCode>SOA</stateCode>
  </return>
  <return>
    <cityCode>CMB</cityCode>
    <code>RML</code>
    <countryCode>LK</countryCode>
    <name>Colombo Ratmalana Apt</name>
    <stateCode>SOA</stateCode>
  </return>
  <return>
    <cityCode>TRR</cityCode>
    <code>TRR</code>
    <countryCode>LK</countryCode>
    <name>Trincomalee Airport</name>
    <stateCode>SOA</stateCode>
  </return>
</ns2:getAirportsResponse>
```

Figure C.1 : Sample SOAP Message

The optimized message generated from the system for above mentioned XML is shown on the figure C.2 below.

```
<ns2:getAirportsResponse xmlns:ns2="http://testservice">
  <![CDATA[#b@ADP@ADP@LK@Anuradhapura
Airport@SOA#b@BTC@BTC@LK@Batticaloa Airport@SOA#b@CMB@CMB@LK@Colombo
Bandaranaike Apt@SOA#b@GOY@GOY@LK@Gal Oya Airport@SOA#b@JAF@JAF@LK@Jaffna
Airport@SOA#b@CMB@RML@LK@Colombo Ratmalana Apt@SOA#b@TRR@TRR@LK@Trincomalee
Airport@SOA]]>
</ns2:getAirportsResponse>
```

Figure C.2 : Sample Optimized Message



University of Moratuwa, Sri Lanka.  
Electronic Theses & Dissertations  
[www.lib.mrt.ac.lk](http://www.lib.mrt.ac.lk)

## Appendix D

### Sample template file.

Figure D.1 shows a sample template file that is stored on the ontology.

```
01 return
02 b
03 2
04 <return><cityCode>{1}</cityCode><code>{2}</code><countryCode>
   {3}</countryCode><name>{4}</name><stateCode>{5}</stateCode>
   </return>
05 cityCode, code, countryCode, name, stateCode
06 name=01:(.*) Aitport#02:Colombo (.*) Apt
```

Figure D.1 : Sample Template File

The first line of the template file contains the tag name of the complex type. The second line is the unique key generated from the system to identify this template. The key is made as short as possible such that the transmitting message will be short. The third line contains the level of this tag in the XML document. Then on the fourth line it contains the tags corresponding to the complex type. The keyword list is saved in the fifth line. The last line of the file contains the content patterns recognized by clustering. Data in both fourth line and the sixth line are static content of that complex type of the xml document. If there is a new tag identified for the same complex type (change to line 04) then a new file should generated and both file should exists on the ontology. But if new static content identified from data clustering process, then the new pattern will be added to the end of the line six. No patterns will be deleted from the existing file.