CHAPTER 6

FUZZY SLA AGREEMENT GENERATION FOR MULTI PARTY USERS

6.1 USER TRANSACTIONS AND SERVICE LEVEL AGREEMENT GENERATION

In service marketing environment, based on the customer requirements service offers are published with multiple service providers. During agreement generation, different optional services are initially registered with UDDI repository. Storage area has the collection of all type of services that are registered, either in public or private registry. Among the two registries, public registry is viewed and accessed with unregistered users and private registry is accessed with registered users. Normally, in service agreement generation, services are published through repository and displayed in the form of yellow pages. From these pages, the published services are viewed by other service providers who buy the mandatory services. Along with the agreement generation, the providers can transfer an acknowledgement. If the service acknowledgement is satisfied with the service provider, Service Level Agreement is generated between these two providers.

The service agreement is usually generated, if both service providers are satisfied with their requirements. Using agreement generation, services are accessed easily and frequently from the required services. Moreover, the Service Level Agreement (SLA) is written in metadata language, because each company has their own requirements and their

requirement functionalities are easily defined. While designing the agreement, certain methods of processing are noted and used for generation of the agreement.

6.1.1 Service History Information Verification

In SLA generation, the two parties are defined. They are service providers and consumers. The service providers may provide multiple offers to the customers and the customers consume the offered services. Before making an agreement to the provider, customers should review and do a clear clarification about their service priorities. The discussion is conducted to check service provider commitments in the previous and present level of history information. Such pre-analysis study does only improve the customer understanding and verification about their needed requirements. In one way, service provider designs the service agreements with respect to customer feedbacks and complaints. This formation is designed before provider writes the service agreement. And if any, modifications are monitored during SLA design and corrected during agreement generation.

6.1.2 Fixing Agreement Confirmation

The oral collaborative discussion of two party confirmation is a fixed problem in service agreement generation. In review meeting discussions, each provider explains about their service view points. Particular provider has his own service plans, quality parameters and service costs. During discussions or review meetings, once the final satisfied goal is reached, then SLA is generated between two providers. Also, the finalized fixed agreement is based on the confirmation from both parties. Different service boundaries, data descriptions and functionalities are described in detail to form a successful SLA generation. The final formation of an agreement must satisfy both parties and be useful to access dissimilar

services. If the written agreement does not define the specific purpose, then the grounding of agreement is a waste of time between two parties.

6.1.3 Agreement Generation

After fixing the agreement generation processing, the structure of the agreement should be drawn. The structure depicts the contents, development tasks and the major constraints of services. Contents may be based on both parties agreements. If the requesting customer is satisfied with buying service, then the contents are defined with that service provider's service. The development tasks will represent the scheduling of activities that are purchased or agreed with the customers. The generated tasks are based on the chosen service and list the selected service functionalities. The constraints of the agreement are written with the agreed conditions of both parties. The satisfied conditions are clearly discussed and framed as conditions. During agreement generation, constraints play active role and state the price values and quality factors of the service.

6.1.4 Completion of Service Agreement

With the detailed structural design view, both service parties may get a clear idea about the service. Each provider has their own service structural representations that are discussed in a detailed way. Discussed structures vary with each party, among which the satisfied conditions are selected and framed as an agreement. The fixing of such conditions finally initiates the implementation of the agreement. The processed implementation may preferably choose any selected language. The meta data description is probably defined to represent the service constraints and functionalities of the other way, SLA is established with SLA developers, who achieve prior experience in the elected field. Established SLA is further discussed with

custom service manager, business system manager, quality manager, service level manager and project manager.

Above discussed key points are the major processing elements of the service agreement. Once the generated agreement focuses on these criteria, the agreement is considered to be valuable and offers more services to the customers.

6.2 ESTABLISHING EFFECTIVE SLA AGREEMENTS

The best Service Level Agreement (SLA) is generated with the right understanding and communication between service provider and customer. Right agreement is proposed to the customer at the right time. Clear functionalities and many factors are considered while establishing the agreement. During agreement generation, the final agreement is generated after face-to-face negotiation. Such communication only improves the short and frequent distribution of agreement document between two parties. In other way, establishment of agreement is region-wise, based on either national or international boundaries.

During agreement generation, customer and provider should clearly represent their dealings. The trust worthiness and quality standards of the agreed services are represented in the design architecture. Once the statements are clearly considered, the proper view is defined and executed. If the company launches their first SLA for the first time, it has to put forth multiple views by the project manager, business manager, service level manager, support service manager etc. With these persons the design document is molded and executed multiple times. Finalized document is sent for customer approval. Normally, the agreement publishing takes 3 to 6 months of time. For exact information gathering, analysis and collaboration implementation, it

takes three months. If the proper information is not gathered by the project teams, then the Service Level Agreement (SLA) gets delayed.

If the above mentioned points are considered, when generating the SLA agreement, it will provide the clear view of understanding between customer and service provider. So, the customers can easily chose the selected services from UDDI repository. And the services are listed under public and private repositories to store the group of services. From the list, customers can access their requirements-suited service and provide Service Level Agreement (SLA). In the proposed HOAG system, the agreement is written using Fuzzy based language to define the conditional constraints. Here, the agreement is finalized and generated with the Decision Manager, User Requirements and Expert Advice members. After the final marking of agreement is done, it is sent for customer approval. Also, the agreements provide multiple choices to choose the selected services for required customers.

6.3 DESIGN ARCHITECTURE OF FUZZY BASED OFFER AGREEMENT GENERATOR

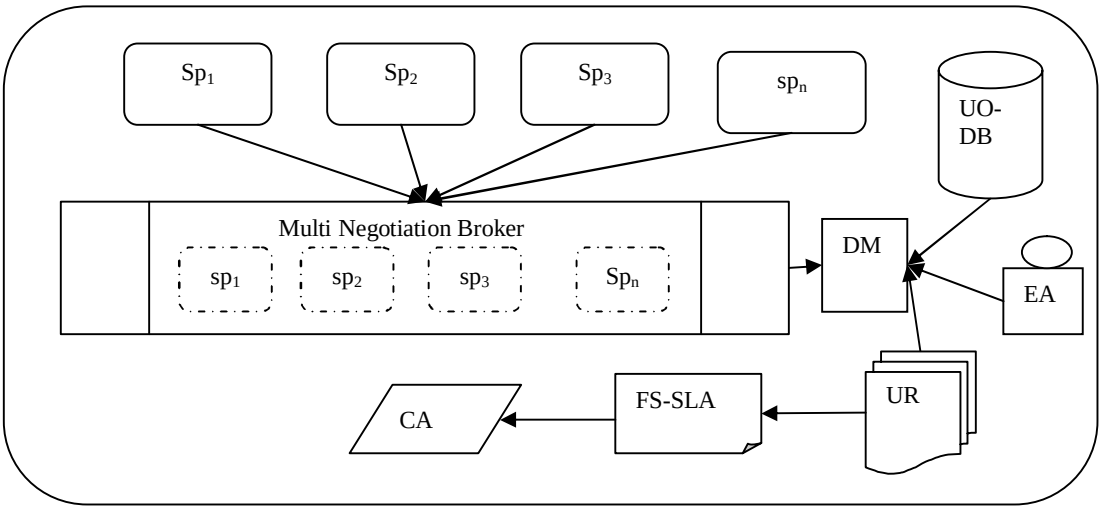

Figure 6.1 Fuzzy Based Offer Agreement Generator System

The fuzzy agreement generator act as layer 3 and from this layer, the agreement is generated with the components of Decision Manager, User-Oriented Database, Expert Advice and User Requirements. The functional generation is displayed in the pictorial view of figure 6.1. and this will be the final solution executor, where it is based on the pre-execution of layer 2. After the classification of bulk requirements, the authenticated data is passed from Classified Filter to Fuzzy Based Agreement Generator.

The layer 3 execution starts with multiple service providers. The customer requirements are matched with the service provider's services and stored in the Multi Negotiation Broker (MNB). Here, the broker is the multi-collective storage and has the collection of preferred services. The collected data from the Negotiation Broker are further input into the Decision Manager. The manager will take decisions by using the User Oriented Database, Expert Advice and User Requirements. From the concluded decision, fuzzy agreements are generated and sent for Customer Approval. If the customer is satisfied with the agreement, they buy the services. Otherwise, they reject the services. The overall functionality ends with this layer. Also it acts as the stop functionality to produce agreements to the customers.

6.3.1 Decision Manager

Decision manager is the deciding authority to make decisions from the valuable conclusions obtained from the User Oriented Database, User Requirements and Expert Advice. The final documented service agreement is generated based on the Decision Manager. The prime agreement is designed with the parameters of service Cost, Duration and Quality. From the listed parameters, customers define their required functionalities. Then, the given options are matched with the service provider services and analyzed with the User Oriented Database and Expert Advice. Finally, weights are allotted for the requirements so that they are represented as 'High', 'Medium' and 'Low'.

From the preferred priority, the customer selected parameters are executed by Decision Manager using the Decision Manager algorithm shown in Figure 6.2

1. Accept the User Requirements.
2. Evaluate the User Requirement parameters with the User Oriented Database and Expert Advice.
3. Execute the collected data with respect to cost, duration and quality parameters.
4. If matching was found set the weightage.
5. The generated weightage to be any one of the options of “High”, “Medium” and “Low”.

The DM algorithm was displayed with the coding..

```
namespace Decision Manager
{
public partial class UserRequriement : System.Web.UI.Page
{
        dbaccess obj = new dbaccess();
protected void Page_Load(object sender, EventArgs e)
{
        if (IsPostBack == High)
{
obj.GridLoad(GridView1, obj.DataadapterQuery("select parameter,1 'from_value',1
'to_value' from input_field_parameters"));
elseif
if (IsPostBack == Medium)
obj.GridLoad(GridView2, obj.DataadapterQuery("select parameter,1 'from_value',1
'to_value' from input_field_parameters"));
else
obj.GridLoad(GridView3, obj.DataadapterQuery("select parameter,1 'from_value',1
'to_value' from input_field_parameters"));
}
```

Figure 6.2 Decision Manager Algorithm

6.3.2 User-Oriented Database System

This database will act as the active database, where it collects more than 2000 set of service provider services. The database collects the present services and the existing services. Present services are the services collected for 5 years from the present year, and the existing services are the services of more than 10 years to be enrolled in the business field. The User Oriented

Database (UO-DB) system is analyzed and matched with the received requirements from the User Oriented Database. If the non-matched service data are found, they are discarded and resent to the customers. So, this database only collects the user registered details and user feedbacks.

6.3.3 User Requirements

These are the requirements which are collected from end users. The types of users are external users and they have various choices of selecting services. The proposed HOAG system provides the service selection option with the parameters of Cost, Duration and Quality. With respect to the parameters, customers can choose the listed services with the options of ‘High’, ‘Medium’ and ‘Low’. In HOAG system, the collected requirements also act as feedback to execute and choose the service providers. To select the required service, multiple choices are given to the customers. They are

1. High Cost with High Duration and High Quality.
2. High Cost with Low Duration and Low Quality.
3. Low Cost with Medium Duration and Medium Quality.
4. Medium Cost with High Duration and High Quality.
5. Medium Cost with Medium Duration and Medium Quality.
6. Low Cost with Low Duration and Low Quality.

With the multiple choice customers can choose the required options. In the above, priority selection ‘From’ and ‘To’ criteria is listed to the customers. From that, they can submit their optional criteria. According to the chosen service, weights are assigned randomly to calculate the requirement options. A formula is proposed in this research work to calculate the requirement specification.

X_1 = Cost X w_1

X_2 = Duration X w_2

X_3 = Quality X w_3

where… w_1, w_2 and w_3 are the weights provided for user requirements

For each service, an individual weightage is set and is multiplied with the individual parameters of Cost, Duration and Quality. Figure 6.3(a) shows the User Requirements algorithm and 6.3 (b) shows the functionalities of User Requirements algorithm

1. Initiate the service offers with various service providers.
2. User Requirements are evaluated based on the Service Cost, Duration and Quality.
3. Analyze the requirements with Decision Manager.
4. Execute the requirement analysis with the given range of "From" and "To" criteria.
5. Compare the analyzed requirement with Decision Manager.
6. Examine the user requirements with the formula of X_1, X_2, X_3.

X_1 = CostX w_1

X_2 = DurationX w_2

X_3 = Quality X w_3

Figure 6.3 (a). User Requirement Algorithm

```
public partial class UserRequriement : System.Web.UI.Page
{       dbaccess obj = new dbaccess();
protected void Page_Load(object sender, EventArgs e)
{       if (IsPostBack == false)
        obj.GridLoad(GridView1,obj.DataadapterQuery("selectparameter,1
        'from_value',1  'to_value' from input_field_parameters"));
        HP.HP objwebHP = new HomogeneousApp.HP.HP();
        objwebHP.PreAuthenticate = true;
        objwebHP.Credentials=
        System.Net.CredentialCache.DefaultCredentials;
 (txt_product_name.Text.Trim(),  XMLString); }
```

```
DataSet ds1= objwebHP.ResultOutput(txt_product_name.Text
.Trim ( ),XMLString);
Samsung.Samsung objwebSamsung = new HomogeneousApp.
                                    Samsung.Samsung();
objwebSamsung.PreAuthenticate = true;
objwebSamsung.Credentials = system.Net.CredentialCache.
                                    DefaultCredentials;
DataSetds2 = objwebSamsung.ResultOutput
```

Figure 6.3(b) User Requirement Function

6.3.4 Expert Advice

Expert Advice (EA) is the advice token from a knowledgeable person or expert who has years of experience in the relavant field. The expert gives suggestions and forwards the review commands to the Decision Manager. Based on the valuable suggestions provided by the expert ,the manager does the analysis and prepares service agreement. During the agreement generation, the collected user requirements are analyzed with the Expert Advice, and suggestions are finalized with the flag set up. Two types of flags are set up for the suggestions namely flag ‘1’ and flag ‘0’. If the analyzed requirement is accepted, the flag to be set up with ‘1’ and if not it assigned with flag ‘0’. From this flag analysis, customer requirements are easily estimated. The steps of the Expert Advice algorithm are shown in Figure 6.4 (a) and its functionality represented in 6.4 (b).

1. Retrieve data from User Requirements.
2. Match the data with User Oriented Database.
3. If matching was satisfied review process started.
4. Based on the weightage, a flag is set to state the status of each data.
5. If the flag state is ‘1’ , the filtered accepted with best accuracy, if the flag set to ‘0’ data get rejected.
6. Collect the finalized flag and put for review process.

Figure 6.4 (a) Expert Advice Algorithm

```
public partial class Expert_Advise : System.Web.UI.Page
{       dbaccess obj = new dbaccess();
protected void Page_Load(object sender, EventArgs e)
{       if (IsPostBack == false)
{       obj.GridLoad(GridView1, obj.DataadapterQuery("select
        parameter,1 'From_value',1 'To_value' from
        input_field_parameters"));
}}
protected void btn_request_Click(object sender, EventArgs e)
{       string XMLString =
        obj.ConvertDataSetToXMLString(Grid1_DatatableUpdatation());
        XMLString = XMLString.Replace(" ", "");
HP.HP objwebHP = new HomogeneousApp.HP.HP();

objwebHP.PreAuthenticate = true;
        objwebHP.Credentials = System.Net.CredentialCache.
        DefaultCredentials;
        DataSet ds1 = objwebHP.ResultOutput
        (txt_product_name.Text.Trim(), XMLString);Samsung.Samsung
        objwebSamsung = new HomogeneousApp.Samsung.Samsung();
        objwebSamsung.PreAuthenticate = true;
        objwebSamsung.Credentials = System.Net.CredentialCache.
                                    DefaultCredentials;
        DataSet ds2 = objwebSamsung.ResultOutput

        (txt_product_name.Text.Trim(), XMLString);
        ds1.Merge(ds2);
if (ds1 != null)
        {       if (ds1.Tables[0].Rows.Count > 0)

        {       GridView2.DataSource = ds1.Tables[0];
GridView2.DataBind();
GridView2.Visible = true;}
else
{
GridView2.Visible = false; }}
else
{

Response.Write("ds is null"); }}
```

Figure 6.4 (b) Expert Advice Algorithm Function

6.3.5 Fuzzy Support-Service Level Agreement (FS-SLA)

After estimating the decision parameters, the Decision Manager generates the final document based on fuzzy inference. In the proposed HOAG system, the agreement is prepared using fuzzy rules because, the most predicted conditional constraints are easily defined. Two types of logic used in fuzzy logic namely the conditional logic and computational logic, where the conditional logic is based on the conditions stated with 'true' or 'false' and the computational logic uses algorithmic methods. The proposed agreement is based on the conditional logic and it states the customer requirements and provider conditions. Also, the individual provider can update, delete and easily modify the meta data during agreement generation. The proposed agreement has been designed using the IF-THEN condition in rules also it states all user preferred criteria in a single statement. Fuzzy SLA is calculated using the formula

$$\text{Service Selection} \quad x = \bar{x} = \sum_{i=1}^{n} \frac{w_i x_i}{\sum w_i} \tag{6.1}$$

where the x_i – is the input data of Cost, Duration, Quality

w_i – is the weight of Cost, Duration ,Quality

Figure 6.5 (a) and 6.5 (b) shows the steps of the proposed FS-SLA algorithm for multi party provider

1. User Requirements are retrieved with the parameters of Cost, Duration and Quality (x_1, x_2,…x_n).
2. Assign a weight w_i for the stated requirements.
3. Extract the value for various ranges of Cost, Duration and Quality.
4. According to the range data categorize the User Requirement as 'High', 'Medium' and 'Low'.
5. Match the collected data with the Decision Manager.
6. Finalize the fuzzy logic with the stated requirements and sent for customer approval.

Figure 6.5 (a) FS-SLA Algorithm for Multi Party Providers

```
            declare @cost numeric (18,2), @Duration numeric (18,2),@Quality
numeric(18,2)
            select @cost = set_value   from @Temp_ds where parameter='cost'
and
            product_code= @product_code
            select @Duration = set_value   from @Temp_ds where
            parameter='Duration' and product_code= @product_code
            select @Quality = set_value   from @Temp_ds where
            parameter='Quality' and product_code= @product_code

            --select @cost ,@Duration ,@Quality
-- to find the cost parameter
            declare @m_cost nvarchar(50)
            select @cost =  isnull (@cost,0)
            if  ((@cost < 1000) or ( @cost between 1000 and 2999))
            begin
                  select @m_cost = 'Low'
            end
            else if ( ( @cost between 3000 and 5999) or ( @cost = 6000) or(
            @cost between 6000 and 7999))
            begin
                  select @m_cost = 'Medium'
            end
            else if ((@cost between 8000 and 25999) or (@cost >= 150000))
            begin
                  select @m_cost = 'High'
            end
            insert into @Temp_decision
            select  @product_code,'Cost',@cost, @m_cost
 -- to find the duration
            declare @m_duration nvarchar(50)
            select @duration = isnull (@service,0)
            if ((@duration <2) or (@Duration between 2 and 3))
            begin
                  select  @m_duration ='Low'
            end
            else if ( (@duration  =4) or (@Duration between 4 and 6)  )
            begin
                  select @m_duration='Mediumn'

            end
            else if ( (@Duration between 6 and 9) or (@Duration=10) or
           (@Duration between 10 and 12))
            begin
                  select @m_duration = 'High'
            end
            insert into @Temp_decision
            select  @product_code,'Duration',@Duration, @m_duration
```

```
-- to find the quality
declare @m_quality nvarchar(50)
select @Quality = isnull (@Quality,0)
 if ((@Quality <2) or (@Quality between 1 and 2))

 begin
              select  @m_quality ='Low'
              end
              else if ( (@Quality=3) or (@Quality between 3 and 5)  )
              begin
                    select @m_quality='Mediumn'
              end
              else if ( (@Quality between 6 and 8) or (@Quality=9) or (@Quality
             between 9 and 10))
              begin
           select @m_quality = 'High'
              end
           insert into @Temp_decision
           select  @product_code,'Quality',@Quality, @m_quality
           fetch next from cur into @product_code
            end
            select distinct  b.ventor,
             a.product_code,b.product_name,a.parameter,a.set_value,a.fuzzy_set
from
@Temp_decision a , @Temp_decision b
where a.product_code=b.product_code
CLOSE cur
DEALLOCATE cur
```

Figure 6.5 (b) FS-SLA Algorithm for Multi Party providers Function

The user requirement weightage is calculated from the fuzzy formula. Each weightage is calculated for the Cost, Duration and Quality. The above algorithm displays the value ranges assigned for ‘High’, ‘Medium’ and ‘Low’. If the cost ranges from <1000 or above 1000 to 2999, it will be assigned as the ‘Low’ category. Similarly, if the cost ranges from 3000 to 5999 or above 6000 to 7999, it is assigned as ‘Medium’ category. when cost ranges from 8000 to 25,999 or above 1, 50,000, they are assigned as ‘High’ category. Like the cost calculation, the duration is also calculated as ‘High’, ‘Medium’ and ‘Low’. If the standard duration ranges from 2 to 3 it is assigned

as ‘Low’. For ‘Medium’, it is ‘>=4’ to ‘6’ and ‘6’ to ‘>=9’ or ‘=>10’ for ‘High’ range. The service product quality range is also defined as ‘Low’, ‘Medium’ and ‘High’. The low product quality ranges between ‘>=3’ to 5, for ‘Medium’ ranges from ‘6’ to ‘8’ or ‘9’ to ‘10’, and ‘>10’lies for the ‘High’ range. Based on the defined ranges, the fuzzy agreement is designed and sent for customer approval.

6.3.6 Customer Approval

The generated fuzzy based agreement is sent for Customer Approval. If the customers are satisfied, they can choose the opted category so that an agreement will be generated between service providers and customers. The main functionality of the Customer Approval module is to send the approval document to the service customers and to receive the acknowledgement from the customers. In the proposed HOAG system, the fuzzy service agreement provides all required requirements stated by the customers. The algorithm shown in Figure 6.6 depicts the customer approval generation algorithm.

```
protected void btn_Consumer_Approval_Click(object sender, EventArgs e)
    {  Response.ContentType = "application/pdf";
       Response.AddHeader("content-disposition",
                                  "attachment;filename=Export.pdf");
       Response.Cache.SetCacheability(HttpCacheability.NoCache);
       StringWriter sw = new StringWriter();
       HtmlTextWriter hw = new HtmlTextWriter(sw);
       HtmlForm frm = new HtmlForm();
       GridView2.Parent.Controls.Add(frm);
       frm.Attributes["runat"] = "server";
       frm.Controls.Add(GridView2);
       frm.RenderControl(hw);
       StringReader sr = new StringReader(sw.ToString());
       Document pdfDoc = new Document(PageSize.A4, 10f, 10f, 10f, 0f);
       HTMLWorker htmlparser = new HTMLWorker(pdfDoc);
       PdfWriter.GetInstance(pdfDoc, Response.OutputStream);
       pdfDoc.Open();
```

```
htmlparser.Parse(sr);
pdfDoc.Close();
Response.Write(pdfDoc);
Response.End(); }
```

Figure 6.6 Customer Approval Algorithm

6.4 SAMPLE VALUE ANALYSIS OF LAYER 3

This layer provides final agreement generation using the fuzzy language. Each authenticated credentials are executed in the Classified Filter and moves to the agreement generator.

- In the last layer the service selection (X) was executed with the formula of

$$x = x = \sum_{i=1}^{n} \frac{w_i x_i}{\sum w_i} \qquad (6.2)$$

- where the service Id ‘HP200’,Transaction Id ‘2325’, password ‘12#77gg’and purchase amount ‘8000’ are collected for the third layer execution.
- Here the xi are calculated as x1= Cost 8000*(HP200) * 23

  x2= Duration (3) * 13

  x3= Quality (5) * 16
- For cost parameter service Id is recognized , to choose the prompt service.
- After the xi calculation, the sum of weight is calculated as

  Σ wi = 23+13+16 = 52
- The value of xi = ((HP200)*23) *(3*13)*(5*16)/52

  = (HP200)23*39*80/52

=(HP200)*(71760)/52

X =(HP200)1380

- where 1380 is the hidden key values passed to the preferred customers. Figure 6.7 and 6.8 displays fuzzy based agreement generation process.

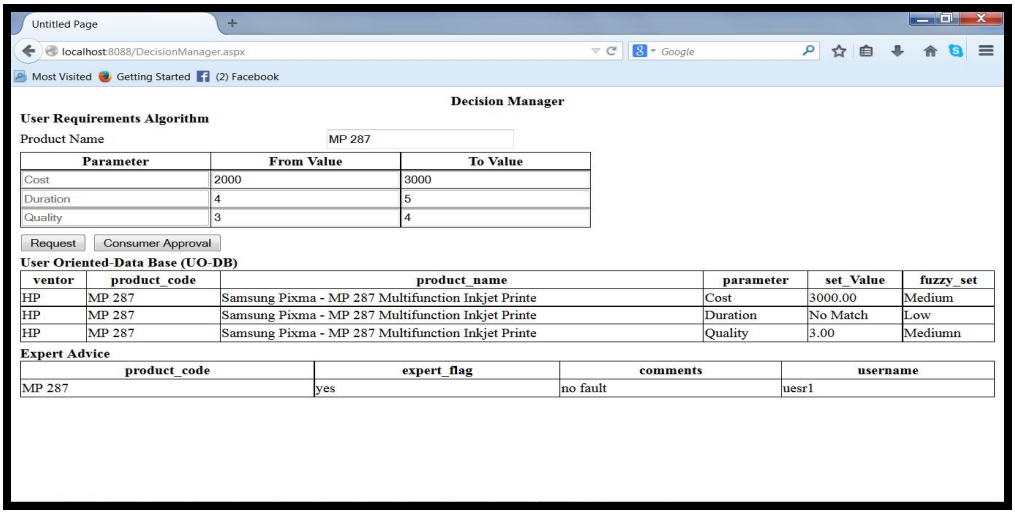

Figure 6.7 Sample Data Execution in Fuzzy Based Agreement Generation

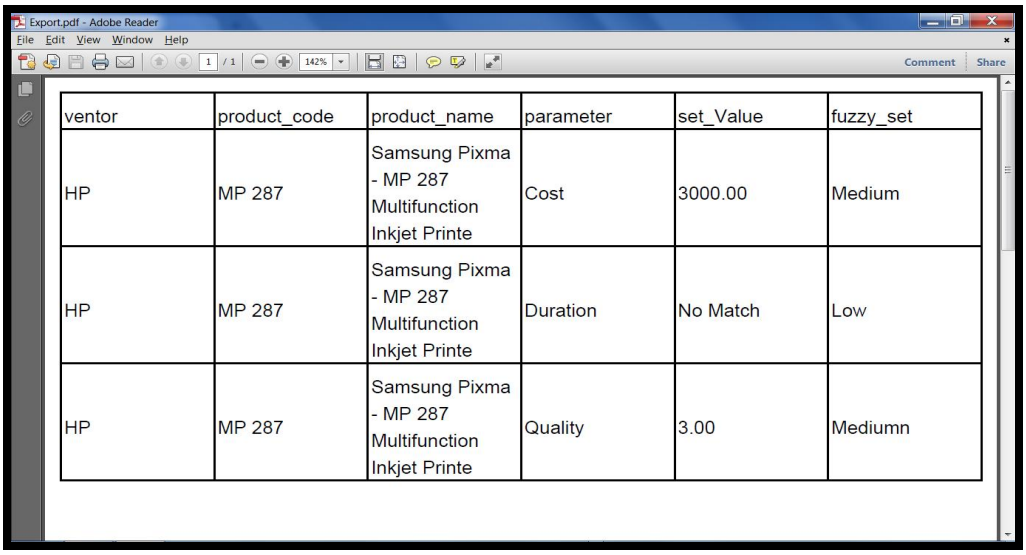

Figure 6.8 Customer Required Fuzzy Based SLA Agreement Generation

6.5 EXPERIMENTAL RESULTS OF LAYER 3

The classified requirements from layer 2 credentials are fed to the Fuzzy Based Agreement Generator. This is the base layer from which the requirements are matched with the service providers and generate the agreement. The experiment was tested with the 10,000 instance data from layer 2. Based on the customer requirements the data verified or matched with the service providers. The selected service provider services are taken as a copy and stored in the Multi Negotiation Broker. This Broker only stores the matched set of services. From this layer, testing analysis happens with the major components of Decision Manager, Fuzzy Support–Service Level Agreement (FS-SLA) and bilateral comparison.

6.5.1 Decision Manager

Decision Manager execution is processed with Expert Advice, User Requirements and User Oriented Database. The proposed HOAG Decision Manager was compared with existing experiment. The existing experiment was tested with agent based approach that makes the long time analysis to execute the results. The experimental testing values are shown in table 6.1, where the number of users taken range from 2200, 1000,800,300 and 100.

Experiments are carried out with the sec from 0.5 sec, 10.11 sec, 20.5 sec,25 sec and 30.66 sec. For the first 0.5 sec timing, 2200 total number of users has been taken for testing among that the proposed Decision Manager Produce 250 users and experiment produces 200 users. Compared to the Experimental the proposed Decision Manager tested with 2200 users from that 250 users are found out to be the matched requirement users. The analysis was tested with the 0.5 ns with 250 users for the proposed Decision Manager, during the graph analysis figure 6.9 represented as 0.5 to 250 as a blue graph.

In second 10.11 sec timing, 1000 users has been taken among that 200 users filtered for Decision Manager and 200 for experimental. In third test 20.5 sec 800 users has been taken among that 300 users for Decision Manager and 150 users for experimental purpose. For the fourth test 300 users requests are tested from those 220 users negotiated for Decision Manager and 100 users for experimental with the timing of 25 sec. Finally 30.66 sec 88 users negotiated for proposed Decision Manager and 50 users for experimentation.

Table 6.1 Experimental Results of Proposed DM

Time (sec)	Decision Manager (prop)	Experimental Results	Total Number of Users
0.5	250	200	2200
10.11	200	200	1000
20.5	300	150	800
25	220	100	300
30.66	88	50	100

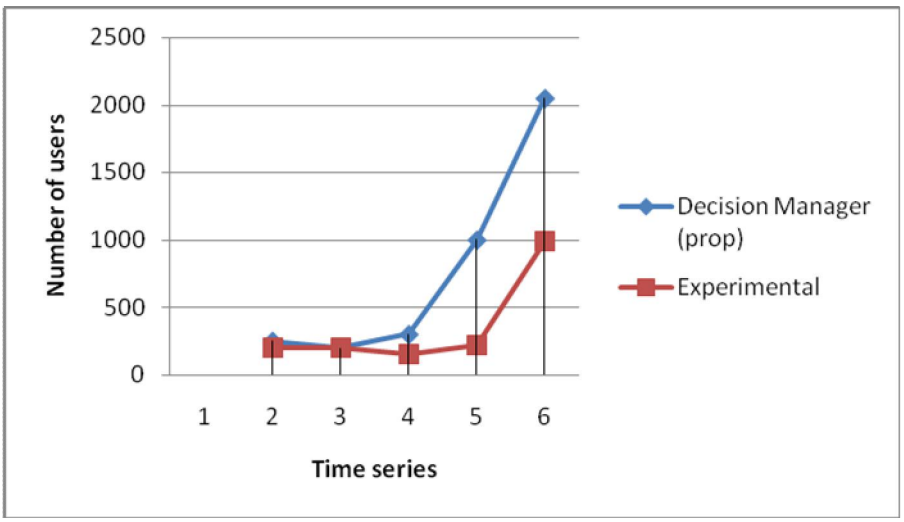

Figure 6.9 Comparative Results of Proposed DM with Experimental

The proposed HOAG Decision Manager was compared with the existing experimentation. In the graph shown in Figure 6.9 analysis was designed between the time series and number of users. From the analysis ‘X’ axis represents time series and ‘Y’ axis represents number of users. In graphical representation, the blue graph shows the increasing curve of Decision Manager and red graph shows decreasing curve of experimentation. The proposed blue graph increases high that represents that it has the more user frequency accessibility with the short time duration. The red graph shows the less time accessibility than the blue graph. When compared to the existing experimental works, the proposed Decision Manager produces more reasonable user calculation.

6.5.2 Fuzzy Support- Service Level Agreement (FS-SLA)

In business environment service providers offers multiple services to their customers. From more number of services, the customers can search the best offered services within a frequent time. Where, the proposed HOAG system FS-SLA was tested with respect to the parameters of Cost, Duration and Quality.

6.5.2.1 Comparative Result with Respect to Cost

In Table 6.2 the proposed work FS-SLA was tested with more than 7000 users. For the test samples it takes 0.5 sec, 10.11 sec, 20.5 sec, 25 sec and 30.66 sec. In the first test 0.5 sec was taken, from that cost ranges from 1000 to 2999 and the customers accumulated by FS-SLA are 127 users and BSLA (Bilateral Service Level Agreement) is 100 users. Second test is carried for 10.11 sec from that 100 users accumulated for FS-SLA and 80 users for BSLA with the cost of 3000 to 5999. Third test was carried out with the cost of 6000 to 7999 with the timing of 20.5 sec from that 150 users accumulated for FS-SLA and 93 users for BSLA. The fourth test was taken with the timing

of 25 sec with the cost of 8000 to 25999 from that 100 users taken for FS-SLA and 74 users filtered for BSLA.

Table 6.2 Comparative Analysis of FS-SLA with BSLA Agreement with Respect to Cost

TIME (sec)	Cost	FS-SLA (users)	BSLA (users)
0.5	1000 – 2999	127	100
10.11	3000-5999	100	80
20.5	6000-7999	150	93
25	8000-25999	100	74
30.66	>150000	66	20

The last test was processed for the cost of >1, 50,000 with the timing of 30.66 sec from that FS-SLA was taken 66 users and BSLA was taken for 20 users. From the overall result, it shows that FS-SLA produce more reasonable filtered users than the BSLA.

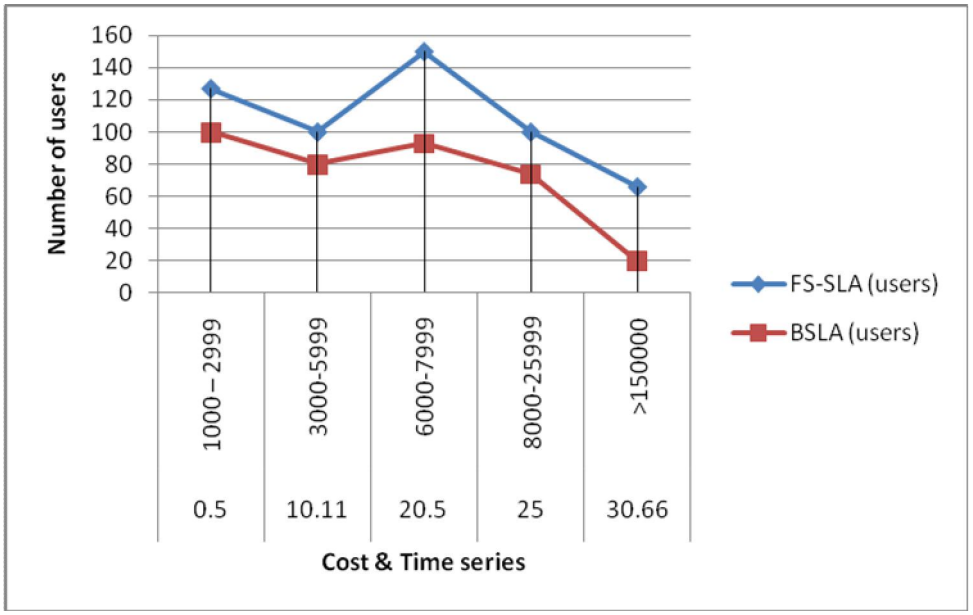

Figure 6.10 FS-SLA Vs BSLA with Respect to Cost

The graphical representation Figure 6.10 shows the performance analysis of FS-SLA and BSLA. In graph ‘X’ axis takes cost and time series and ‘Y’ axis takes number of users. Where the red curve shows the sliding curve of BSLA and blue curve shows the increasing curve of FS-SLA.According to the price value, the red graph increased with the number of users with the frequent time period and BSLA agreement less users can access the service provider services. From the curve range, it shows the accurate result of users can be filtered for FS-SLA than with BSLA.

6.5.2.2 Comparitive Result with Respect to Duration

The duration was depicted with the table 6.3 where the first test processed with the duration of 0 to 2 months for 0.5 seconds, 127 users tested for FS-SLA and 100 users tested for BSLA. Second test processed with the duration of 2 to 3 months for 10.11 seconds 100 users tested for FS-SLA and 80 users from BSLA, from this test it depicts that the FS-SLA is accepted with high number of users than BSLA. Third test use 150 users of FS-SLA and 93 users of BSLA for the time of 20.55 seconds tested with the duration of 4 to 6 months, during this testing the FS-SLA accepted with more number of users. Fourth test conducted with the duration of 7 to 9 months with the listed users of 100 from FS-SLA and 74 users from BSLA, during this test results it depicts that FS-SLA produce high ratio users. The last test conducted with 10 to 12 months with 66 users form FS-SLA and 20 from BSLA during this test it presents the high covered user ratio with FS-SLA

Table 6.3 Comparative Analysis of FS-SLA with BSLA Agreement with Respect to Duration

TIME (sec)	Duration(months)	FS-SLA (users)	BSLA (users)
0.5	0 to 2	127	100
10.11	2 to 3	100	80
20.5	4 to 6	150	93
25	7 to 9	100	74
30.66	10 to 12	66	20

The graph analysis shown in Figure 6.11 displays the performance analysis of FS-SLA and BSLA. The 'X' axis displays the duration and time series where 'Y' axis represents number of users. The blue curve depicts the increasing ratio of the FS-SLA and red curve depicts the sliding ratio of BSLA. Compared to the red graph the blue graph achieves 66 users accessing ratio with the standard of 10 to 12.From the value ranges it shows that FS-SLA has produce high priority range value than BSLA.

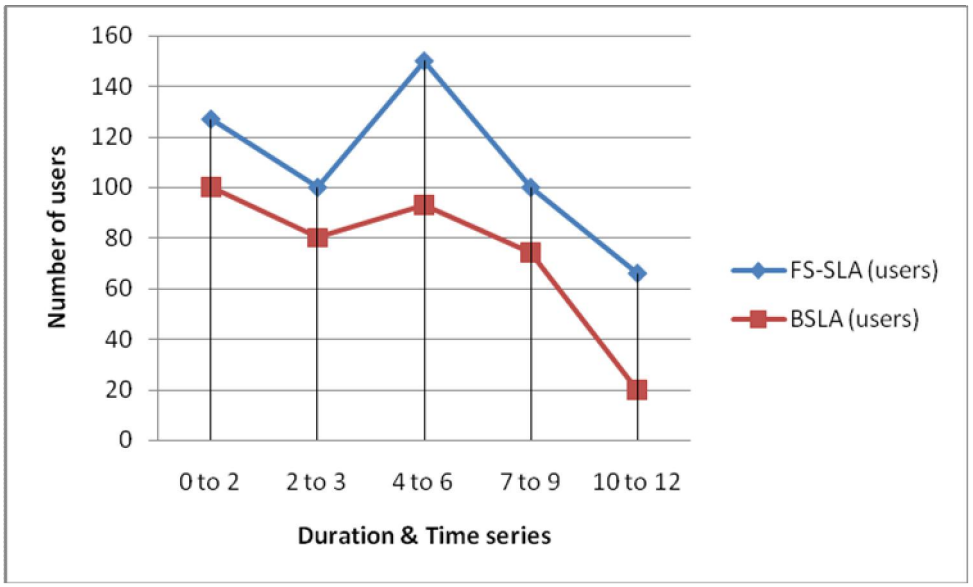

Figure 6.11 FS-SLA Vs BSLA with Respect to Duration

6.5.2.3 Comparative Result with Respect to Quality

Quality is the one of the factor to predict the verifiable services. The quality factor is fixed with the standard range of 1 to 2, 3 to 5, 6 to 8 and 9 to 10 and 10 to 12. The quality standard tested with the number of users and showed in Table.6.4. with the standard of 1 to 127 users are accepted with the FS-SLA and 100 users received with BSLA in the time series of 0.5 sec. The range of 3 to 5 standard, 100 users accepted with FS-SLA and 80 users accepted with BSLA in the time series of 10.11 sec. In third range of 6 to 8 tested with the time series of 20.55 sec with 150 users are received with FS-SLA and 93 users received with BSLA. Fourth test done with 100 users of FS-SLA and 74 users of BSLA with the standard of 9 to 10 with the time duration of 25 sec. Fifth test done with 66 users for FS-SLA and 20 users for BSLA with the standard of 10 to 12 with the time duration of 30.66 sec.

Table 6.4 Comparative Analysis of FS-SLA with BSLA Agreement with Respect to Quality

TIME (sec)	Quality	FS-SLA (users)	BSLA (users)
0.5	1 to 2	127	100
10.11	3 to 5	100	80
20.5	6 to 8	150	93
25	9 to 10	100	74
30.66	10 to 12	66	20

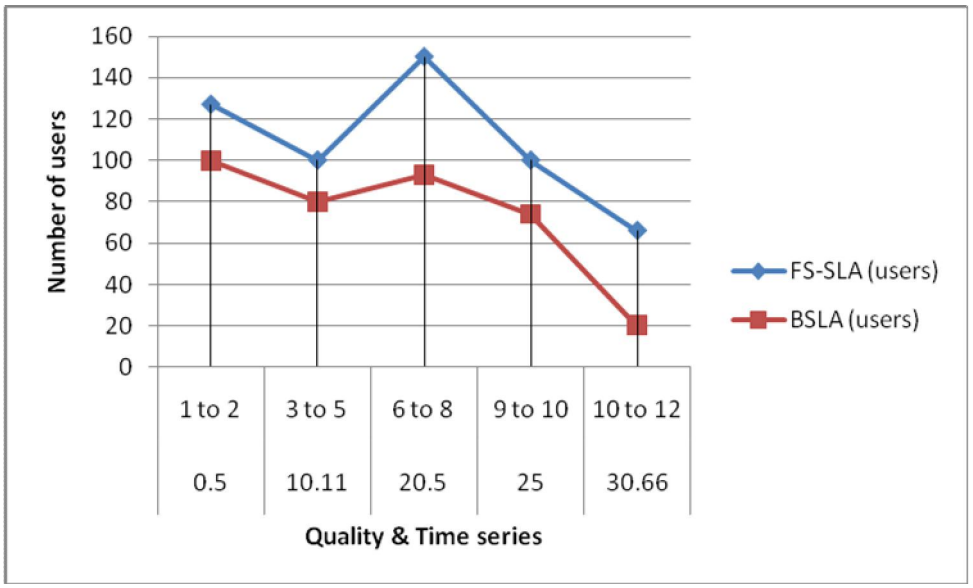

Figure 6.12 FS-SLA Vs BSLA with Respect to Quality

Figure 6.12 shows the Quality standards analysis graph from quality standards and time series and number of users. The red wave shows the BSLA and blue wave shows FS-SLA. Time series indicates the standards from 1 to 2 represents 0.5 sec, 3 to 5 represents 10.11 sec, 6 to 8 represents 20.5 sec and 9 to 10 represents 25 sec and 10 to 12 represents 30.66 sec. From the analysis, it is observed that the proposed work produce high analysis range than the existing system.

6.5.2.4 Multiparty Negotiation system vs Bilateral system

The multi party users agreement was tested with the set of users for both multi lateral users and bilateral users. For the testing analysis first sample was tested with the time of 30.66 sec from that 2500 users accessed by multi lateral agreement and 200 users for bilateral agreement. Secondly for 25 sec, multi lateral agreement accessed with 1000 users and bilateral agreement accessed with 100 users. In third, 20.5 sec multi lateral agreement accessed with 800 users and bilateral agreement accessed with 150 users. For 10.11 sec multi lateral agreement accessed by 300 users and bilateral agreement accessed with 100 users. In 0.5 sec, the multi lateral agreement was accessed

by 100 users and bilateral agreement accessed by 50 users. Table 6.5 shows the comparative execution of multiparty system with bilateral system.

Table 6.5 Comparative Analysis of Multiparty System with Bilateral System

Time Duration (sec)	Multi lateral users	Bilateral users
30.66	2500	200
25	1000	200
20.5	800	150
10.11	300	100
0.5	100	50

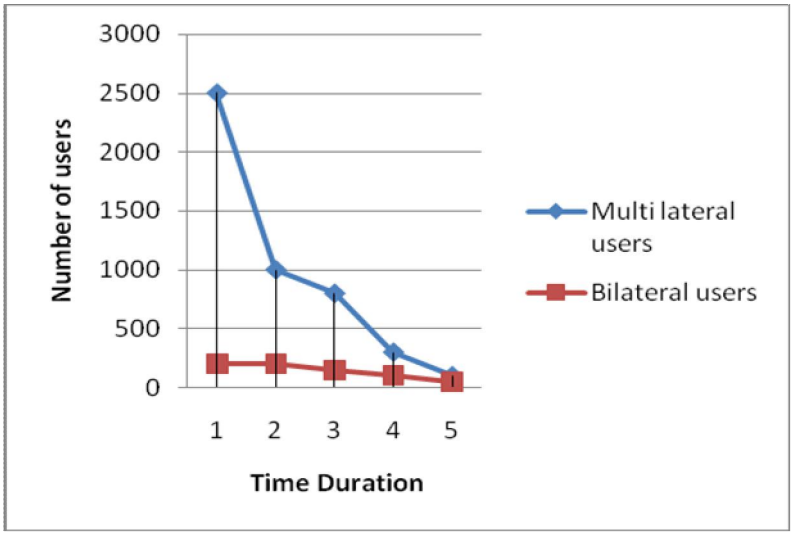

Figure 6.13 Multiparty Agreement System Vs Bilateral Agreement System

Graph shown in Figure 6.13 displays the overall performance analysis for the multiparty agreement and bilateral agreement. 'X' axis represents the time duration and 'Y' axis represents the number of users, where the red graph denotes the increasing ratio of bilateral users and blue graph represents for multi lateral users. From both these comparisons, it can

be observed that the accessing ratio of users for the proposed system will be high than the existing system.

6.6 SUMMARY

Thus, the above experimental of layer 3 provides the generation of fuzzy level agreement for multiple providers. Here, the service providers are collected based on the user requirements and matched with their services. The fuzzy algorithm was designed with the Cost, Duration and Quality. The design of algorithm will reduce the searching time of customers to access the required service providers.

Compared to the existing research works, the proposed HOAG system produces the available services to the customers with the quick period of time. As for each defined parameter of Cost, Duration and Quality weights are managed and according to the user accessing requirements the weights get fixed.