

# Fuzzy Based Multi-Agent System Offering Cost Effective Corporate Environment

Anirban Kundu<sup>\*1</sup>, Sutirtha Kr. Guha<sup>1</sup>, Alok Ranjan Pal<sup>1</sup>, Tanay Sarkar<sup>1</sup>, Subhendu Mandal<sup>1</sup>, Rana Dattagupta<sup>2</sup> and Debajyoti Mukhopadhyay<sup>3</sup>

<sup>1</sup>Netaji Subhash Engineering College, (West Bengal University of Technology), Kolkata - 700152, West Bengal, India; <sup>2</sup>Jadavpur University, Kolkata 700032, West Bengal, India; <sup>3</sup>Calcutta Business School, Diamond Harbour Road, Bishnupur 743503, West Bengal, India

**Abstract:** We propose a general framework of a hierarchical structure, consisting of several levels of activities, for typical software related corporate hierarchy using multi-agent system. This work identifies the functionality of each level. Each and every level is considered as agent who is described further using fuzzy analysis. Our framework consists of six levels within which first five levels are considered as agents followed by the human interaction at the lowermost level. These agents interact with each other to produce a desired result for the client based on autonomous decisions which are decided through fuzzy reasoning with the help of predefined databases. A layered architecture has been proposed in this paper for showing a corporate office hierarchy in a cost effective manner. In general, management employees of the corporate system draw a huge amount of money for their activities. Our ultimate aim is to reduce the cost of the existing corporate system by observing and controlling the behavioral characteristics of each level of hierarchy by replacing typical manual operations with agents [1,2]. We have presented a case study or practical engineering example along with the description of each agent.

**Keywords:** Multi-Agent, Client Agent (CA), Project Manager Agent (PMA), Team Leader Agent (TLA), Conceptual Design Developer Agent (CDDA), Subject Expertise Agent (SEA).

## I. INTRODUCTION

Agent based systems have produced lot of excitement in recent years. This is a new paradigm for conceptualizing, designing and implementing different types of systems. Agent systems are specifically attractive for creating software which operates in distributed environment like Internet [3,4].

Multi-Agent system consists of several layers and each layer is known as 'Agent'. Research in multi-agent system is basically concerned with the study, behavioral characteristics and construction of a collection of agents which interact with each other [5-7].

The agents are considered to be autonomous entities, such as software programs or robots. In general multi-agent systems are computational systems in which a lot of agents interact together to perform a particular job. Multi-agent system distributes computational resources and capabilities across the network of interconnected agents. It allows for the interconnection and interoperation of multiple existing legacy systems. Multi-agent system efficiently retrieves, filters and globally coordinates information from sources that are spatially distributed and also provides solutions in situations where expertise is spatially and temporarily distributed. It enhances overall system performance, specifically along the dimensions of computational efficiency, reliability, extensibility, robustness, maintainability, responsiveness, flexibility and reuse [8,9].

This paper is organized as follows: Section II represents a brief preview of Related Work. The proposed system framework is discussed in Section III along with data analysis using fuzzy system. All the agents have been reported in the following sub-sections under Section III along with case study. Section IV depicts all about the Conclusion & Future Work possibilities.

## II. RELATED WORK

Researchers from around the world have suggested several applications on multi-agent systems. An application of multi-agent systems in the electronic market (e-market) is already proposed in the paper "An Agent-mediated Electronic Market of Semantic Web Services". It is basically a multi-agent system which enables the complete automation of business processes and the system is able to automate the semantic comparison of required and advertised Web services according to user preferences. The system consists of semantic and mobile agents responsible for autonomous service discovery, automated negotiation and automated purchase. Specific concepts and ontology used for creating semantic information and agent interactions, as well as mechanisms that provide the ability of semantic reasoning, are presented [10].

In [11], the authors have proposed a multi-agent based negotiation platform for improving the effectiveness and efficiency of cooperative decision-making in construction supply chain (CSC) adopting agent technology and regarding CSC as a typical multi-agent system. Negotiation is an effective and popular decision-making and coordination behavior in inter-organization systems, especially in CSC which is characterized with fragmentation, low efficiency and multiple partners. General structure of the agent based negotiation

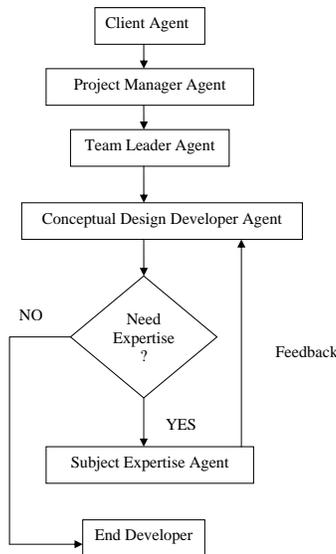
\*Address correspondence to this author at the Netaji Subhash Engineering College, (West Bengal University of Technology), Kolkata - 700152, West Bengal, India; E-mail: anik76in@yahoo.co.in

platform is designed, which includes two kinds of agent group: specialty agents and service agents. Since different members in CSC have different preferences on the decision attributes (such as cost, time, quality, safety and environment), a multi-attribute negotiation model is established by designing negotiation protocol and describing the negotiation process.

A lot of similar works have been going on in different fields of research using Multi-Agent systems for intelligent control in automation. In this paper, we are going to propose an application of Multi-Agent system using fuzzy analysis to replace the existing hierarchy structure of corporate office.

**III. PROPOSED SYSTEM FRAMEWORK**

Multiple agents have been used to realize our proposed system. We have divided our system using the following agent subsystems: Client Agent, Project Manager Agent, Team Leader Agent, Conceptual Design Developer Agent & Subject Expertise Agent. At the lowest level of hierarchy, there is a need for interaction between the agents & human programmers. These programmers are treated as ‘End Developers’ and are well connected to the agents as shown in Fig. (1).



**Fig. (1).** Proposed System Framework of Corporate Environment.

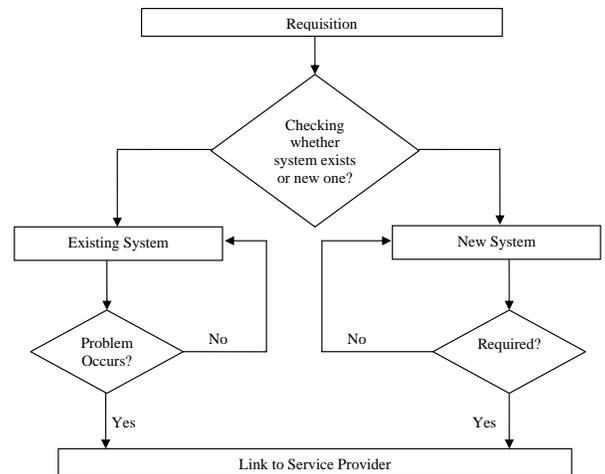
We have introduced fuzzy set theory in our paper since it provides a smooth transition between members & non-members of the concerned system. Implementation of Fuzzy is relatively simple, fast & adaptive. Fuzzy system is also less sensitive to system fluctuations. Since we are dealing with practical corporate environment, there may be some internal or system fluctuations. Thus, there is a need for using fuzzy system for getting better results. There is no need of a mathematical model due to the fact that Fuzzy set theory depends on the interval between 0 (False) & 1 (True) to describe human reasoning. In this approach, every level (agent) is treated as a distinct mathematical fuzzy set [12,13].

Practical or hypothetical set of test cases are placed with each agent in order to evaluate our proposed concept. For each and every agent, some crisp inputs have been considered for a particular time instance as a practical case study. Corresponding membership function values are found by

plotting the crisp inputs on the membership function graphs. These membership values are then put into the corresponding rule base of the particular agent. Finally, fuzzy decision index has been found using maximum / centroid method. This fuzzy decision index value is the ultimate output of each level (agent). In this paper, each agent is justified using the practical case study associated with it. Thus, experimental results have been shown along with the theoretical description of agents within each subsection of Section III.

**A. Client Agent**

CA gets overall information of a new/existing system along with the problem specification from the client-side to the server through HTTP protocol. CA checks whether the required system is already developed by our proposed system. If it is an existing system, the system highlights the problems already occurred. Otherwise, if it is a new system, requirement analysis is required. The total information with the decision given by CA is then passed on to the PMA as discussed in Algorithm 1. Detailed requirements are necessary for developing a system. So, interaction with client is the most important part in our proposed system using CA interface. A database is required for checking whether the system required by the client is a new/existing system. The detailed view is shown in Fig. (2).



**Fig. (2).** Detailed View of CA.

**Algorithm 1: Client Interaction**

*Input: Request from client with details*

*Output: Information collection by PMA*

*Step 1: Information collected from client*

*Step 2: Checking the database whether the system is new or existing*

*Step 3: If, the system is new, then specify the requirement*

*Step 4: Else If, the system is Existing, then specify the problem*

*Step 5: Information is sent to Service Provider*

*Step 6: Stop*

**Table I. Input Range FoR CA**

Inputs	Specification Range
Project Name	0-25 => Total change required 20-45 => More than Average Change required 40-65 => Average Change required 60-85 => Less than Average Change required 80-100 => No Change required
Field Specification	0-0.6 => New System 0.5-1.0 => Existing System

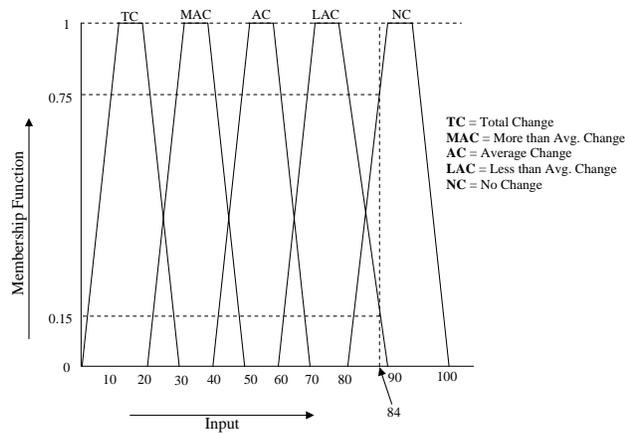


Fig. (3). Membership of Project Name.

Following are the details of CA using Fuzzy reasoning. Table I shows the suggested input ranges of the CA.

Membership functions of input project name and field specification are shown in Figs. (3 & 4) respectively.

The membership function of project name is a combination of different conditions as depicted in Fig. (3).

$$\mu(\text{Project Name}) = \{\mu(\text{TC}), \mu(\text{MAC}), \mu(\text{AC}), \mu(\text{LAC}), \mu(\text{NC})\}$$

The membership function of field specification is depicted in Fig. (4).

$$\mu(\text{Field Specification}) = \{\mu(\text{NS}), \mu(\text{ES})\}$$

Consider, crisp inputs at any particular instance are (i) Project name value = 84 & (ii) Field specification value = 0.52.

After plotting the input value of project name on the graph of Fig. (3), the membership function of project name becomes  $\mu(\text{Project Name}) = \{0, 0, 0, 0.15, 0.75\}$ . Similarly, membership function of field specification becomes  $\mu(\text{Field Specification}) = \{0.2, 0.05\}$  as shown in Fig. (3). Thus, the rule function  $f = \{N, E\}$ .

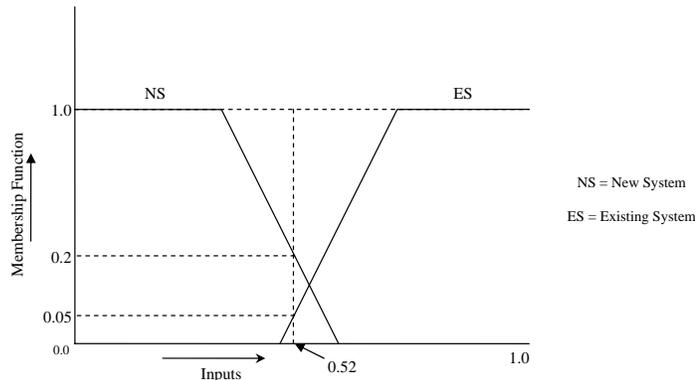


Fig. (4). Membership of Field Specification.

Table II. Rule Base For CA

	Total Change Required	More than Average Change Required	Average Change Required	Less than Average Change Required	No Change Required
New System	NEW	NEW	NEW	NEW	EXISTING
Existing System	EXISTING	EXISTING	EXISTING	EXISTING	EXISTING

Table III has been created after substituting the membership values into Table II performing min operation. At this stage, the rule function becomes  $f = \{0.2, 0.1\}$ .

Table III. Rule Base for CA After Substitution of Membership Values

	Total Change Required	More than Average Change Required	Average Change Required	0.75	0.15
0.2	NEW	NEW	NEW	0.2	0.2
0.05	EXISTING	EXISTING	EXISTING	0.05	0.05

Final Computation at CA (refer Fig. 5):

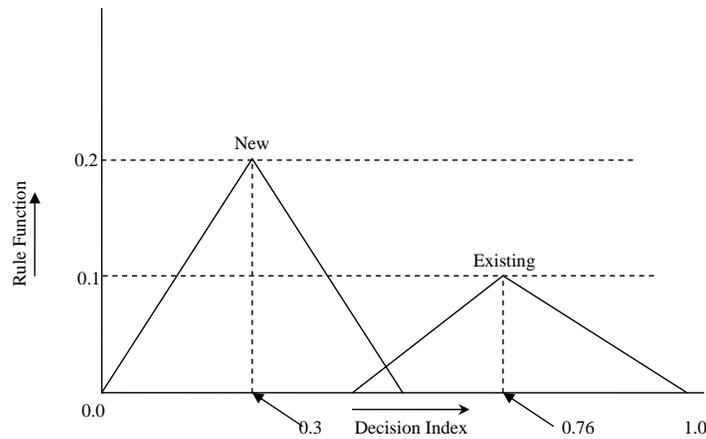


Fig. (5). Scaled Fuzzified Decision.

Two methods have been used in this context:

1. Maximum Method:

In this method fuzzy set with larger value is selected. So, final decision would be ‘New’.

2. Centroid Method:

In this method final decision is calculated based on the following formula.

$$\text{Final decision} = (\sum \mu * D) / \sum \mu = ((0.2 * 0.3) + (0.1 * 0.76)) / (0.2 + 0.1) = 0.453$$

Output: Selection of appropriate field

Step 1: Client Information is being processed by the top level management to select concerned Project Manager

Step 2: Analyze the information using predefined knowledge of Project Manager

Step 3: Characterization of analyzed data based on feasibility criteria

Step 4: If, the information matches with specific field, then the field is selected

Step 5: Else, create a new field for selection and database updated

Step 6: Stop

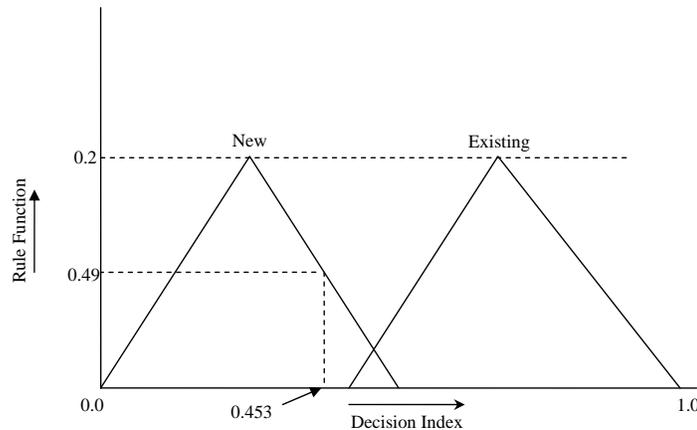


Fig. (6). Fuzzy Decision Index.

Final Decision Index would be 49% in ‘New’ criteria as referred in Fig. (6).

**B. Project Manager Agent**

The main objective of the PMA is the requirement analysis. This agent analyses the collected information using predefined knowledge database. Then, characterization of analyzed data is being done based on feasibility criteria. Further, it checks overall information matching with the specific field data of the same database. If the selected field is a new one, the creation of new field occurs within the database. Finally, the database is being updated as mentioned in Algorithm 2. The pictorial view is shown in Fig. (7).

**Algorithm 2: Project Manager Module**

Input: Collected information using Algorithm 1

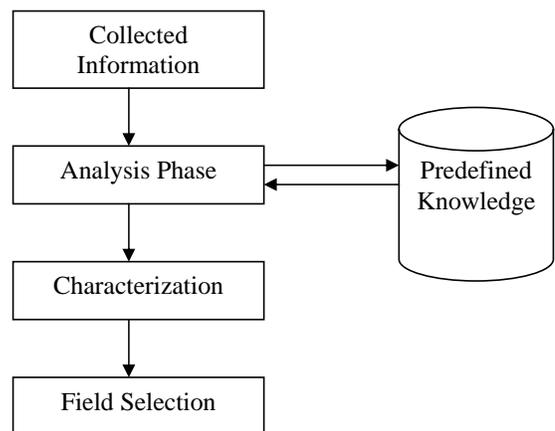


Fig. (7). Flow diagram of PMA.

Following are the details of PMA using Fuzzy reasoning. Table IV shows the details of suggested input ranges of PMA.

**Table IV. Input Range For PMA**

Inputs	Specification Range
Front-end	0-0.6 => Platform dependent 0.5-1.0=> Platform independent
Back-end	0-0.6=> Platform dependent 0.5-1.0=> Platform independent
Budget	0.2-0.3=> Not feasible 0.2-0.6=> Low budget .5-0.8=> Medium budget 0.7-1.0=> High budget

The membership function of the input front-end technology is as follows:

$$\mu(\text{Front-end Technology}) = \{\mu(\text{PD1}), \mu(\text{PI1})\}.$$

The membership function of the input back-end technology is as follows:

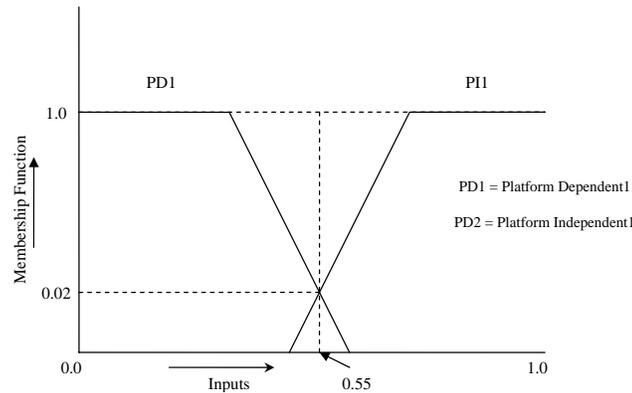
$$\mu(\text{Back end Technology}) = \{\mu(\text{PD2}), \mu(\text{PI2})\}.$$

The membership function of the input budget is defined as follows:

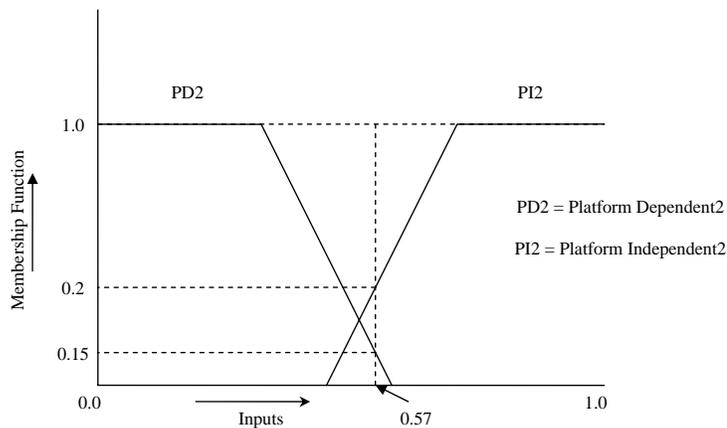
$$\mu(\text{Budget}) = \{\mu(\text{NF}), \mu(\text{LB}), \mu(\text{MB}), \mu(\text{HB})\}.$$

For example, crisp inputs at any particular instance are (i) Front-end technology - 0.55, (ii) Back-end technology - 0.57 & (iii) Budget - 0.53.

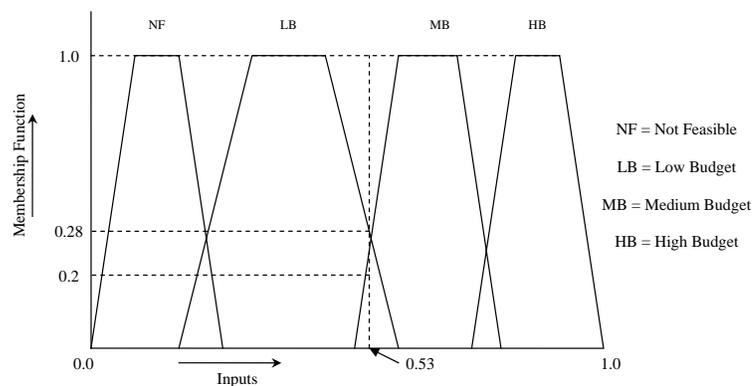
So, the membership function of Front-end Technology becomes  $\mu(\text{Front-end Technology}) = \{0.02, 0.02\}$  as shown



**Fig. (8).** Membership of Front-End Technology.



**Fig. (9).** Membership of Back-End Technology.



**Fig. (10).** Membership of Budget.

in Fig. (8). Similarly, the membership function of Back-end Technology becomes  $\mu(\text{Back-end Technology}) = \{0.15, 0.2\}$  as shown in Fig. (9). From Fig. 10, the membership function of Budget becomes  $\mu(\text{Budget}) = \{0, 0.28, 0.2, 0\}$ .

**Table V. Rule Base-1 for PMA**

	<b>Platform Dependent2</b>	<b>Platform Independent2</b>
Platform Dependent1	T1	T2
Platform Independent1	T3	T4

The rule function  $f1 = \{T1, T2, T3, T4\}$

**Table VI. Rule Base-2 for PMA**

	<b>Not Feasible</b>	<b>Low Budget</b>	<b>Medium Budget</b>	<b>High Budget</b>
T1	NA	A	A	A
T2	NA	NA	A	A
T3	NA	NA	A	A
T4	NA	NA	NA	A

The rule function  $f2 = \{NA, A\}$

**Table VII. Rule Base-1 for PMA after Substitution of Membership Values**

	0.15	0.2
0.02	0.02	0.02
0.02	0.02	0.02

Table VII has been created by substituting the membership values into the 'Rule Base 1' table (Table V) by performing min operation.

**Table VIII. Rule Base-2 for PMA After Substitution of Membership Values**

	<b>Not Feasible</b>	<b>0.28</b>	<b>0.2</b>	<b>High Budget</b>
0.02	NA	0.02	0.02	A
0.02	NA	0.02	0.02	A
0.02	NA	0.02	0.02	A
0.02	NA	0.02	0.02	A

At this stage, the rule function becomes  $f1 = \{0.02, 0.02, 0.02, 0.02\}$ .

Table VIII has been created by substituting the membership values into the 'Rule Base 2' table (Table VI) by performing min operation.

At this stage, the rule function becomes  $f2 = \{0.02, 0.02\}$ .

Final Computation at PMA (refer Fig. 11):

Two methods have been used in this context:

1. Maximum Method:

So, final decision would be '0.02'.

2. Centroid Method:

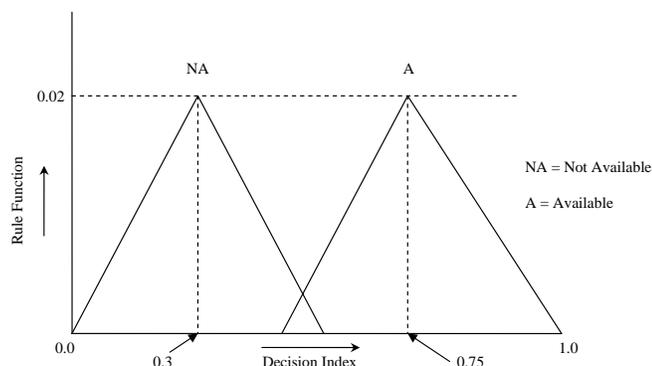
In this method final decision is calculated based on the following formula.

$$\text{Final Decision} = (\sum \mu * D) / \sum \mu = ((0.02 * 0.3) + (0.02 * 0.75)) / (0.02 + 0.02) = 0.525$$

Final Decision Index would be 12% in 'Not Available (NA)' criteria & 8% in 'Available' criteria as referred in Fig. (12).

**C. Team Leader Agent**

TLA is responsible for checking environmental setup and classification of jobs of the assigned problem. Firstly, it



**Fig. (11). Scaled Fuzzified Decision.**

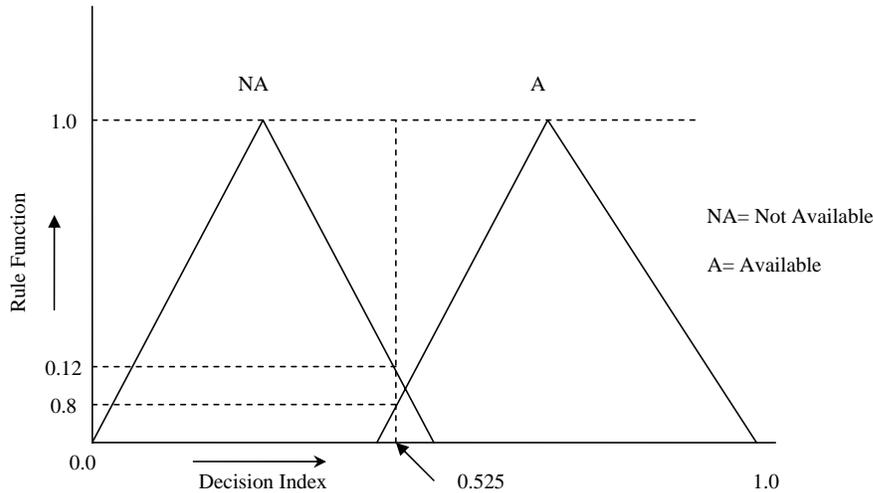


Fig. (12). Fuzzy Decision Index.

thesis of the problem is done for classifying the jobs. This classification process is helpful for distribution of distinct

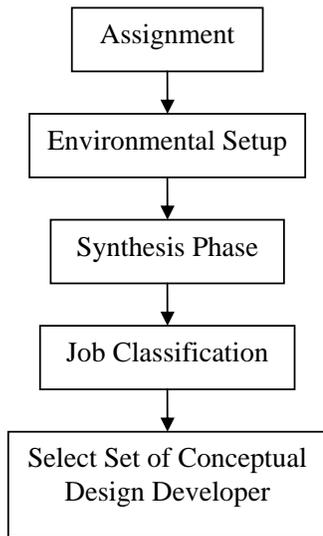


Fig. (13). TLA Activity.

(classified) jobs among the conceptual design developers as referred to Algorithm 3. Fig. (13) shows the detailed view of TLA.

**Algorithm 3: Team Leader**

Input: Information from PMA

Output: Selection of appropriate design developers

Step 1: Information taken from PMA as an assignment

Step 2: Checking existing environmental setup with respect to the assignment

Step 3: If, setup is okay, then goto Step 5

Step 4: Else, introduce a new setup

Step 5: Synthesis phase is being done for the development of design

Step 6: Job classification has been done based on the design requirements

Step 7: Select the appropriate Design Developer for each classified job

Step 8: Stop

Following are the details of TLA using Fuzzy reasoning. Table IX refers the input ranges of TLA. The membership function of the input ‘Front-End’ (Fig. 14) is as follows:

Table IX. Input Range For TLA

Inputs	Specification Range
Front end	0-0.3 => VB
	0.2-0.5=> VC++
	0.4-0.8=> .NET
	0.7-1.0=> J2EE
Back end	0-0.3=> MS Access
	0.2-0.5=> MS SQL
	0.4-0.7=> MySQL
	0.6-0.9=> Oracle 8i
	0.7-1.0=> Oracle10

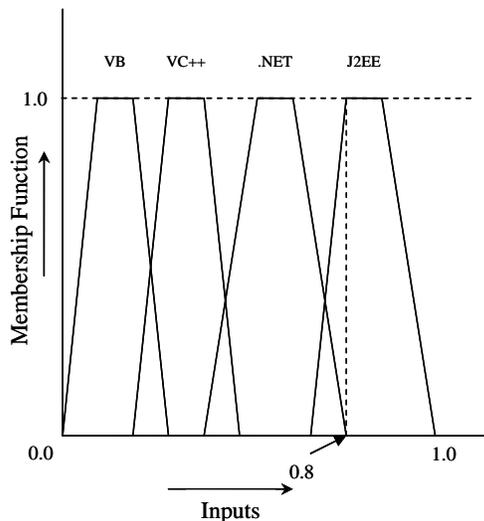


Fig. (14). Membership of Front-End.

$$\mu(\text{Front-End}) = \{\mu(\text{VB}), \mu(\text{VC++}), \mu(\text{.NET}), \mu(\text{J2EE})\}$$

The membership function of the input ‘Back-End’ (Fig. 15) is as follows:

$$\mu(\text{Back-End}) = \{\mu(\text{M1}), \mu(\text{M2}), \mu(\text{M3}), \mu(\text{M4}), \mu(\text{M5})\}$$

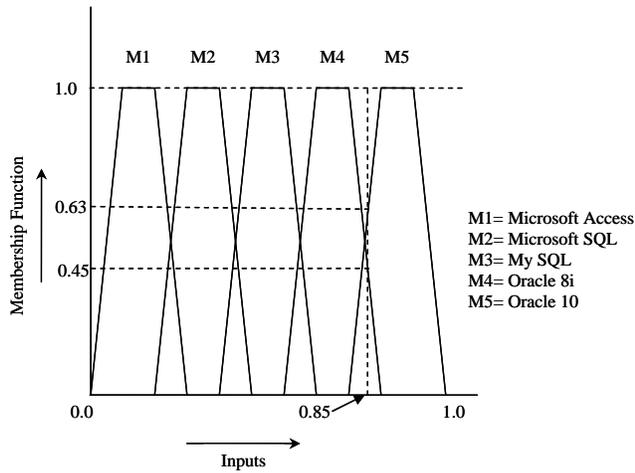


Fig. (15). Membership of Back-End.

For example, crisp inputs at any particular instance are (i) Front-end = 0.8 & (ii) Back-end = 0.85.

So, the membership function of Front-end Technology is  $\mu(\text{Front-end}) = \{0, 0, 0, 1\}$

and the membership function of Back-end Technology is  $\mu(\text{Back-end}) = \{0, 0, 0, 0.45, 0.63\}$

$G^i = i^{\text{th}}$  Developer Group.

The rule function  $f1 = \{G1, G2, \dots, G19, G20\}$  (refer Table X)

The rule function  $f2 = \{NM, M\}$  (refer Table XI)

where, M = Modification in Environmental setup is required; and,

Table X. Rule Base-1 For TLA

	MS Access	MSSQL	MySQL	Oracle 8i	Oracle 10g
VB	G1	G2	G3	G4	G5
VC++	G6	G7	G8	G9	G10
.NET	G11	G12	G13	G14	G15
J2EE	G16	G17	G18	G19	G20

Table XI. Rule Base-2 for TLA

	MS Access	MSSQL	MySQL	Oracle 8i	Oracle 10g
VB	NM	NM	NM	NM	M
VC++	NM	NM	NM	NM	M
.NET	M	M	M	M	M
J2EE	NM	NM	NM	NM	M

NM = No Modification in Environmental setup is required;

Now, substitute the membership values into the 'Rule Base 1' table (Table X) by performing min operation for creating Table XII.

The rule function becomes  $f1 = \{0, 0, \dots, 0.45, 0.63\}$

Now, substitute the membership values into the 'Rule Base 2' table (Table XI) by performing min operation for creating Table XIII.

The rule function becomes  $f2 = \{0.45, 0.21\}$

Final Computation at TLA (refer Figs. 16 & 17):

Table XII. Rule Base-1 for TLA after Substitution of Membership Values

	MS Access	MSSQL	MySQL	0.45	0.63
VB	G1	G2	G3	G4	G5
VC++	G6	G7	G8	G9	G10
0	G11	G12	G13	0	0
1	G16	G17	G18	0.45	0.63

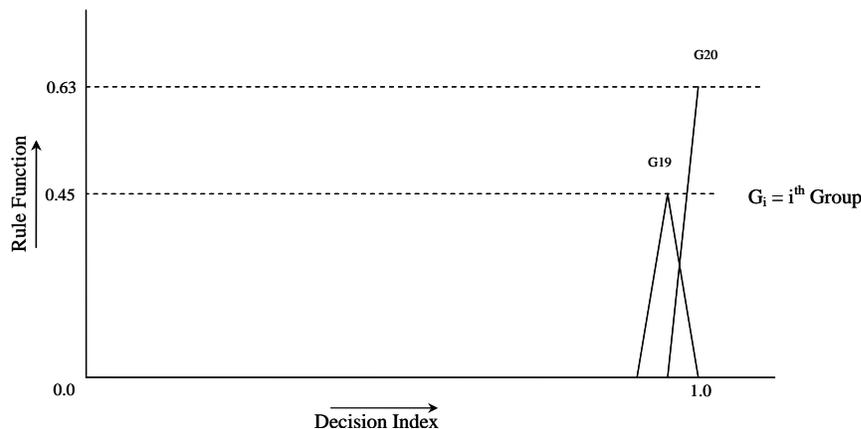


Fig. (16). Scaled Fuzzified Decision for Developers' Group.

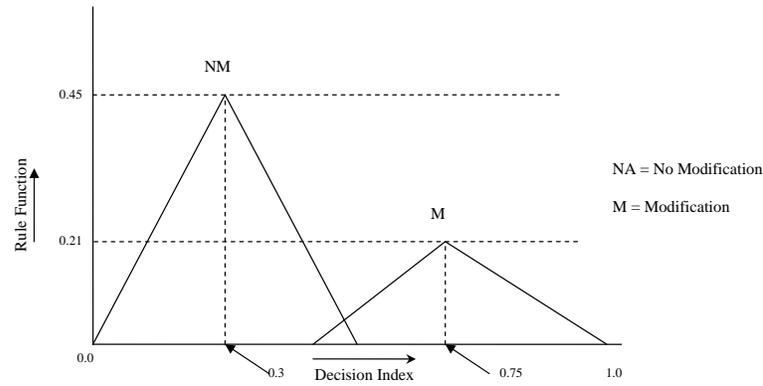


Fig. (17). Scaled Fuzzified Decision for Environmental Setup.

Table XIII. Rule Base-2 for TLA After Substitution of Membership Values

	MS Access	MSSQL	MySQL	0.45	0.63
VB	NM	NM	NM	NM	M
VC++	NM	NM	NM	NM	M
0	M	M	M	0	0
1	NM	NM	NM	0.45	0.63

Two methods have been used in this context:

1. Maximum Method:

So, final decision would be 'G20' with 'NM'. That means 'Group 20' would be assigned for the given task with 'No Modification' required.

2. Centroid Method:

In this method final decision is calculated based on the following formula.

$$\text{Final Decision for Developers' Group} = (\sum \mu * D) / \sum \mu = ((0.63 * 1) + (0.45 * 0.95)) / (0.63 + 0.45) = 0.98$$

$$\text{Final Decision for Environmental Setup} = (\sum \mu * D) / \sum \mu = ((0.45 * 0.3) + (0.21 * 0.75)) / (0.45 + 0.21) = 0.44$$

Final Decision Index would be 75% in 'Group 19' & 25% in 'Group 20' as referred in Fig. (18).

Final Decision Index would be 45% in 'No Modification' as referred in Fig. (19).

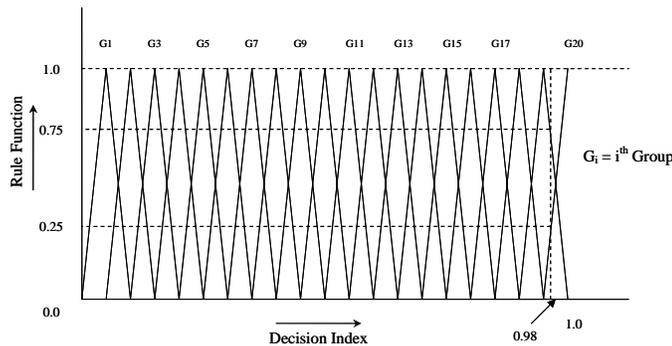


Fig. (18). Fuzzy Decision Index for Developers' Group.

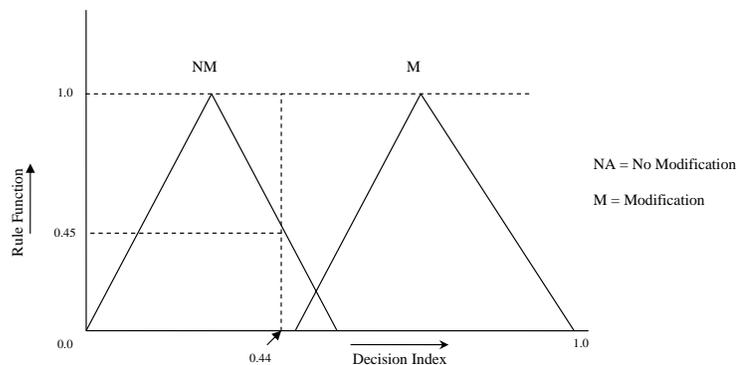
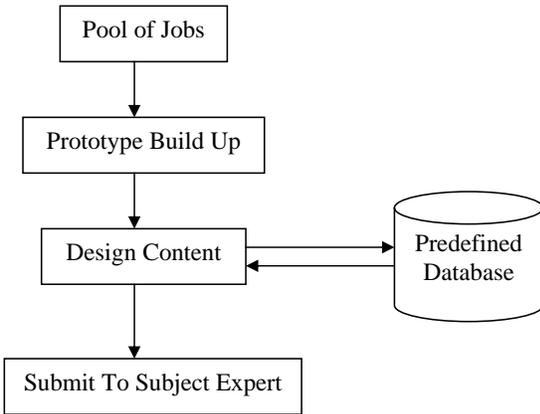


Fig. (19). Fuzzy Decision Index for Environmental Setup.

**D. Conceptual Design Developer Agent**

The conceptual design of each classified part of the previous level is developed using this agent. It builds-up the functional prototypes using predefined database step by step as depicted in Algorithm 4. Generation of design content is totally based on the database. The main objective of this agent is to submit the model of the actual project to the concerned expert as shown in Fig. (20).



**Fig. (20).** Flow diagram of CDDA.

**Algorithm 4: Conceptual Design Developer**

*Input: Job from TLA*

*Output: Design completion*

*Step 1: Assigned job taken from the pool*

*Step 2: Build-up the prototype for the specified job*

*Step 3: Create the content of design using pre-defined database*

*Step 4: Submission of completed design to subject expert and waiting for approval*

*Step 5: Stop*

Following are the details of CDDA using Fuzzy reasoning. The information of requirements, collected by CA, is carried forward to CDDA. The output of TLA (i.e., Developers Group) is also taken into consideration. In Table XIV, the input ranges of Conceptual Design Developer Agent are shown.

The membership function of the input ‘Output from TLA’ (Fig. 21) is as follows:

**Table XIV. Input Range For CDDA**

Inputs	Specification Range
Requirements	0-0.3 => Pattern1
	0.2-0.5 => Pattern2
	0.4-0.8 => Pattern3
	0.7-1.0 => Pattern4
Output from TLA (i.e., Group of Developers)	0.00-0.10 => Group1
	0.05-0.15 => Group2
	0.10-0.20 => Group3
	0.15-0.25 => Group4
	0.20-0.30 => Group5
	0.25-0.35 => Group6
	0.30-0.40 => Group7
	0.35-0.45 => Group8
	0.40-0.50 => Group9
	0.45-0.55 => Group10
	0.50-0.60 => Group11
	0.55-0.65 => Group12
	0.60-0.70 => Group13
	0.65-0.75 => Group14
	0.70-0.80 => Group15
	0.75-0.85 => Group16
	0.80-0.90 => Group17
	0.85-0.95 => Group18
	0.90-1.00 => Group19
	>=0.95 => Group20

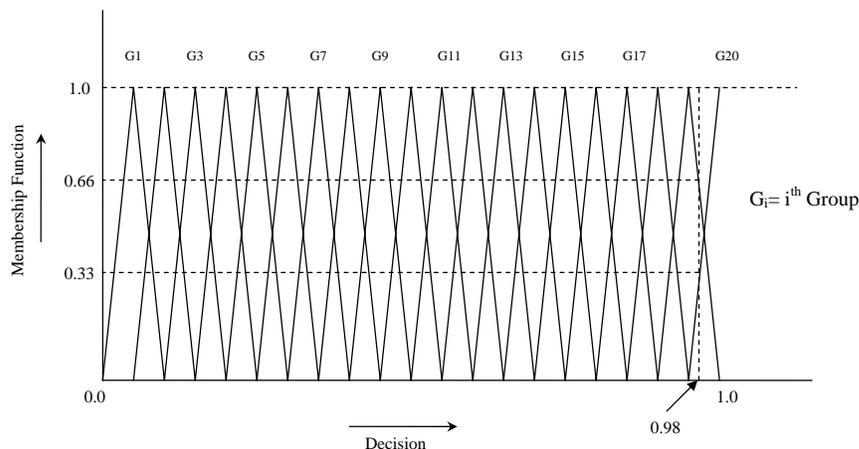
$$\mu(\text{Output from TLA}) = \{\mu(G1), \mu(G2), \dots, \mu(G19), \mu(G20)\}$$

The membership function of the input ‘Requirements’ (Fig. 22) is as follows:

$$\mu(\text{Requirements}) = \{\mu(P1), \mu(P2), \mu(P3), \mu(P4)\}$$

For example, crisp input (output of TLA) at any particular instance is 0.98.

So, the membership function of ‘Output from TLA’ is  $\mu(\text{Output from TLA}) = \{0, 0, \dots, 0.33, 0.66\}$  and the



**Fig. (21).** Membership of Output from TLA.

Table XV. Rule Base for CDDA

	P1	P2	P3	P4
G1	D1	D2	D3	D4
G2	D5	D6	D7	D8
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
G19	D73	D74	D75	D76
G20	D77	D78	D79	D80

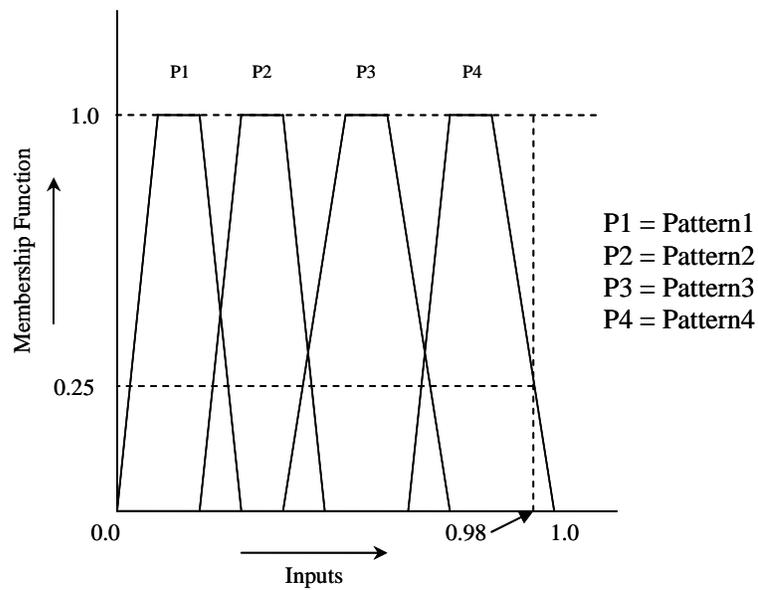


Fig. (22). Membership of Requirements.

Table XVI. Rule Base For CDDA after Substitution of Membership Values

	P1	P2	P3	0.25
G1	D1	D2	D3	D4
G2	D5	D6	D7	D8
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
G18	D69	D70	D71	D72
0.33	D73	D74	D75	0.25
0.66	D77	D78	D79	0.25

membership function of 'Requirements' is  $\mu(\text{Requirements}) = \{0, 0, 0, 0.25\}$

From Table XV, the rule function  $f = \{D1, D2, \dots, D79, D80\}$

Now, substitute the membership values into the ‘Rule Base’ table (Table XV) by performing min operation for creating Table XVI.

Final Computation at CDDA (refer Fig. 23):

Two methods have been used in this context:

1. Maximum Method:

So, final decision would be ‘D76’ or ‘D80’. That means ‘Design 76’ or ‘Design 80’ would be assigned for the given task at this level.

2. Centroid Method:

In this method final decision is calculated based on the following formula.

$$\text{Final Decision for selection of 'Design'} = (\sum \mu * D) / \sum \mu = ((0.25 * 0.94) + (0.25 * 0.99)) / (0.25 + 0.25) = 0.96$$

Final Decision Index would be 70% in ‘D77’ as referred in Fig. (24). Although, according to the Rule base, ‘D77’ is not matching the requirement criteria. So, the result obtained from ‘Maximum Method’ is considered as more appropriate at this stage. Thus, ‘D76’ would be the final decision.

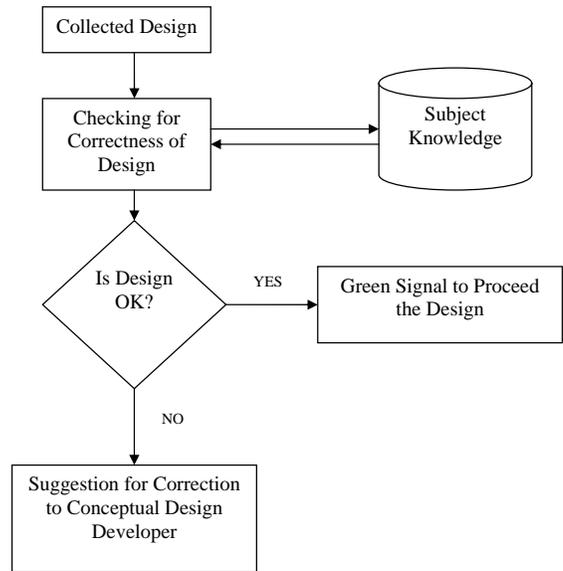


Fig. (25). Illustrative view of SEA.

Algorithm5: Subject Expertise

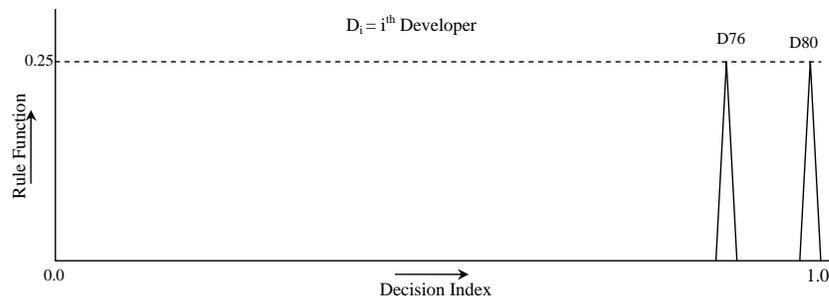


Fig. (23). Scaled Fuzzified Decision.

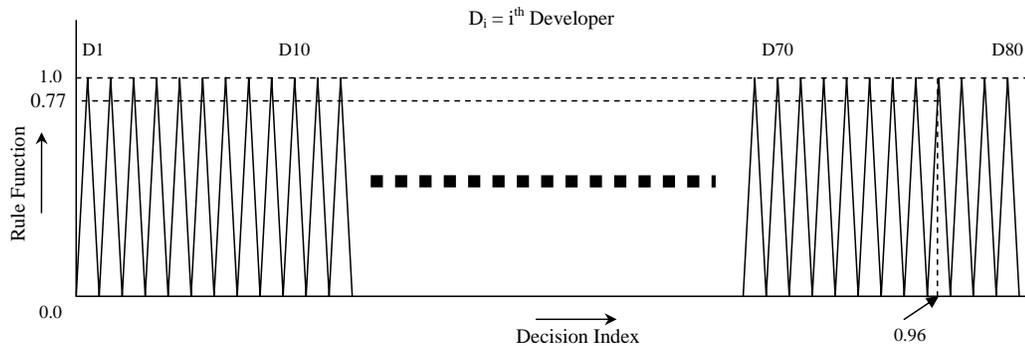


Fig. (24). Fuzzy Decision Index.

**E. Subject Expertise Agent**

SEA coordinates with the CDDA. The main objective of this agent is to check for correctness of the submitted design by the previous level (CDDA). The required decision is taken using Subject Knowledge database. If the design is successful as per standard, it is accepted for approval. Otherwise, design is rejected and suggestion message is being sent to CDDA as feedback as per Algorithm 5. The pictorial representation of SEA activity is shown in Fig. (25).

Input: Collected design

Output: Comments on the design

Step 1: The design is collected from CDDA

Step 2: Checking the design with the help of subject knowledge database

Step 3: If, the design is okay, then green signal is given to proceed with the design

Step 4: Else, the design is rejected and specific suggestions are being sent for correction of the design

Step 5: Stop

Table XVII. Input Ranges for Sea

Inputs	Specification Range
Predefined Database	0-0.25 => Design1 from Database
	0.2-0.45=> Design2 from Database
	0.4-0.6=> Design3 from Database
	0.55-0.75=> Design4 from Database
	0.7-1.0=> Design5 from Database
Output from CDDA i.e., Design type	0.00-0.0125=> Design1
	0.0125-0.025=> Design2
	0.025-0.0375=> Design3
	0.0375-0.05=> Design4
	.
	.
	.
	.
	.
	.
	0.9625-0.975=> Design78
	0.975-0.9875=> Design79
	0.9875-1.0=> Design80

Following are the details of SEA using Fuzzy reasoning. In this agent, information about the predefined knowledge base of the concerned design of system and the output of CDDA (i.e., Design type) are taken into consideration.

Table XVII shows the input ranges of the SEA. Membership functions of 'Predefined Database' & 'Design type' collected from CDDA are shown in Figs. (26 & 27) respectively.

The membership function of the 'Predefined Database' is defined as  $\mu(\text{Requirements}) = \{\mu(\text{DD1}), \mu(\text{DD2}), \mu(\text{DD3}), \mu(\text{DD4}), \mu(\text{DD5})\}$

The membership function of the 'Design type' collected from CDDA is defined as  $\mu(\text{Design type}) = \{\mu(\text{D1}), \mu(\text{D2}), \dots, \mu(\text{D79}), \mu(\text{D80})\}$

Consider, the crisp input (output from Conceptual Design Developer Agent) at any particular instance at this level is 0.94.

From Fig. (26), membership function becomes  $\mu(\text{Requirements}) = \{0, 0, 0, 0, 0.8\}$ ; and from Fig. (27),  $\mu(\text{Design type}) = \{\mu(\text{D1}), \mu(\text{D2}), \dots, \mu(\text{D76}), \dots, \mu(\text{D79}), \mu(\text{D80})\} = \{0, 0, \dots, 0.7, \dots, 0\}$

From Table XVIII, the rule function  $f = \{O, M\}$ .

Now, substitute the membership values into the 'Rule Base' table (Table XVIII) by performing min operation for creating Table XIX.

Now, the rule function becomes  $f = \{0.7, 0\}$ .

Final Computation at SEA (refer Fig. 28):

Two methods have been used in this context:

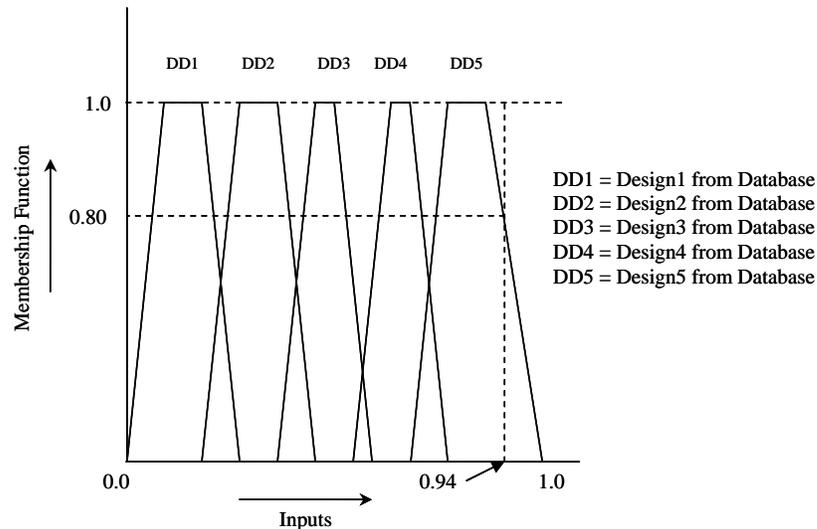


Fig. (26). Membership of Predefined Database.

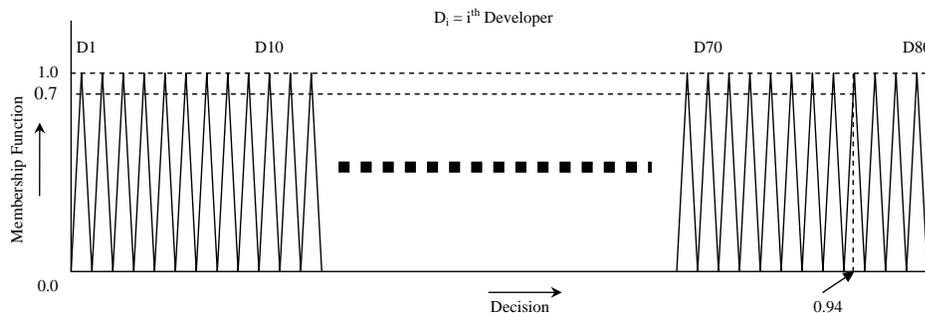


Fig. (27). Membership of Design type collected from CDDA.

Table XVIII. Rule Base for Sea

	DD1	DD2	DD3	DD4	DD5
D1	M	O	O	O	O
D2	O	M	O	O	O
D3	O	O	M	O	O
D4	O	O	O	M	O
D5	O	O	O	O	M
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
D76	M	O	O	O	O
D77	O	M	O	O	O
D78	O	O	M	O	O
D79	O	O	O	M	O
D80	O	O	O	O	M

1. Maximum Method:

So, final decision would be ‘O’ (Design is ok).

2. Centroid Method:

In this method final decision is calculated based on the following formula.

$$\text{Final Decision for } f = (\sum \mu * D) / \sum \mu = (0.7 * 0.25) / 0.7 = 0.25.$$

Final decision would be 100% in ‘O’ group as referred to Fig. (29). That means the design is ok. So, no modification is required.

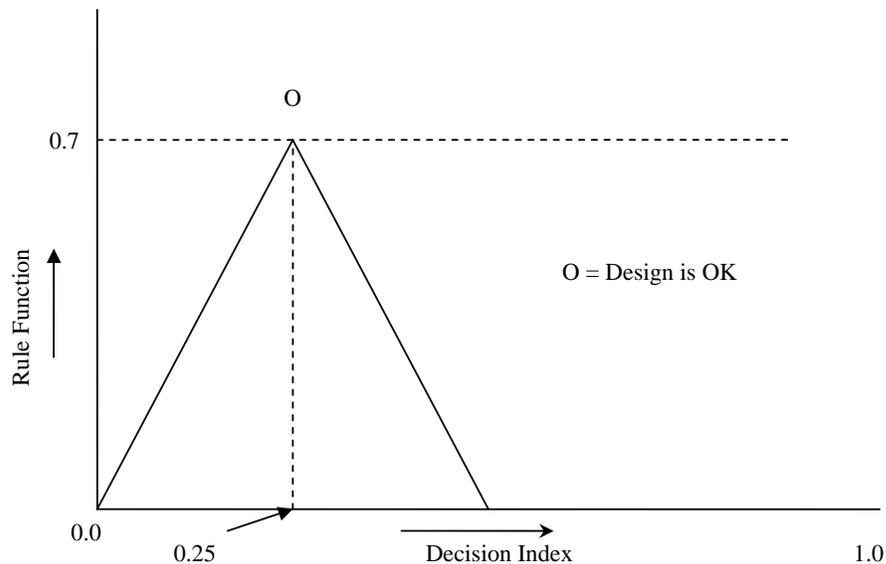


Fig. (28). Scaled Fuzzified Decision.

Table XIX. Rule Base For Sea After Substitution of Membership Values

	DD1	DD2	DD3	DD4	0.8
D1	M	O	O	O	O
D2	O	M	O	O	O
.	.	.	.	.	.
.	.	.	.	.	.
D75	O	O	O	O	M
0.7	M	O	O	O	0.7
D77	O	M	O	O	O
.	.	.	.	.	.
.	.	.	.	.	.
D79	O	O	O	M	O
D80	O	O	O	O	M

F. End Developer

End Developer stage is a manual operational level. After receiving the design from CDDA, the programmers are responsible for developing the new system or modifying the existing system as required. This part is the ultimate development phase. After developing the system, End Developers also test the system accuracy with the actual data given by the user /client (software testing phase). If the test is perfect, the solution would be submitted to the concerned client; otherwise, there should be some modification (refer Fig. 30). Algorithm 6 describes the overall activities of End Developers.

Algorithm 6: End Developer

Input: Approved design

Output: Solution of the problem

Step 1: Approved design is taken from CDDA

Step 2: If, the system is new, then create the system

Step 3: Else, modify the existing system

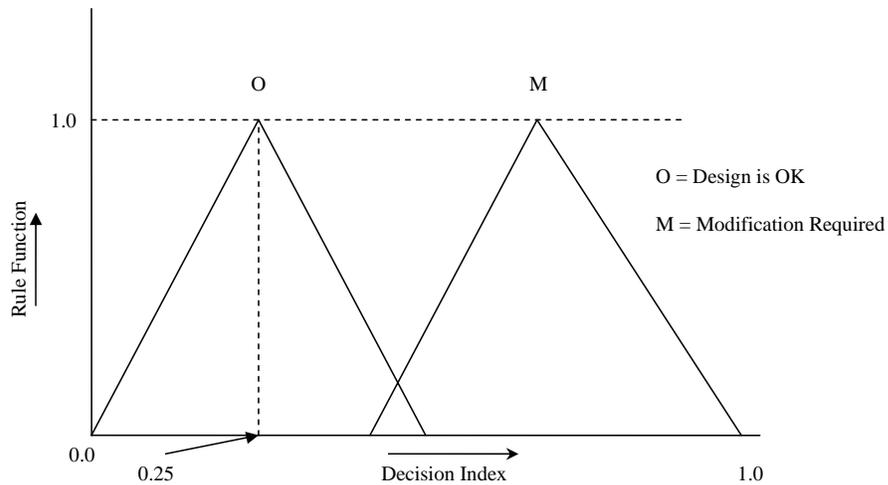


Fig. (29). Fuzzy Decision Index.

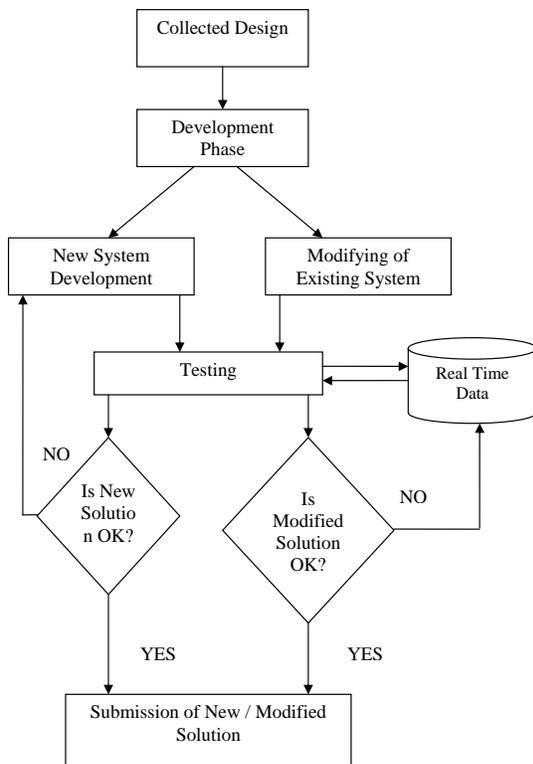


Fig. (30). Illustrative view of End Developers' Activity.

- Step 4: Test the developed system with real time data
- Step 5: If, the solution is okay, then submission done
- Step 6: Else, goto Step 2
- Step 7: Stop

**G. Cost Reduction Calculation**

In a typical software corporate hierarchy,

$$Cost_{Total} = \{Cost_{CA} + Cost_{PMA} + Cost_{TLA} + Cost_{CDDA} + Cost_{SEA} + Cost_{ED}\}$$

where,  $Cost_{CA}$  = Cost or expense required for managing interfacing between client & project manager;

$Cost_{PMA}$  = Cost or expense required for managing project managers;

$Cost_{TLA}$  = Cost or expense required for managing team leaders;

$Cost_{CDDA}$  = Cost or expense required for managing design developers;

$Cost_{SEA}$  = Cost or expense required for managing the subject experts;

$Cost_{ED}$  = Cost or expense required for managing end-developers or programmers;

In our approach,

$$Cost_{Total} = \{X + Cost_{ED}\}$$

where, X = One time investment for making the software;

$Cost_{ED}$  = Cost or expense required for managing end-developers or programmers;

So, in the long run,  $X \ll \{Cost_{CA} + Cost_{PMA} + Cost_{TLA} + Cost_{CDDA} + Cost_{SEA}\}$ .

Thus, our method is cost effective.

**IV. CONCLUSION & FUTURE WORK**

In this paper, we presented the overall structure of a corporate office hierarchy in a cost effective manner by replacing the levels of the existing system through agents. It is a novel approach for building Multi-Agent system. The agents are organized in accordance with five levels of problem solving model followed by the 'End Developers'. The main objective of our system is to focus on the autonomous mode of each level resulting in cost effectiveness. We have illustrated our approach in a corporate environment for the cost reduction purpose using our methodology. Since, agents have been utilized at different levels of the hierarchy, so there are no recurring expenses at these levels as compared to manual operations. Only the 'End Developers' level is required for manual operations. Thus, automatically the total cost is reduced in our approach.

In the proposed work, the overall corporate hierarchy is not fully agent based due to the fact that the correctness checking of the software coding with real-time data at End Developer stage requires a huge predefined knowledge base of program functions, routines, sub-routines, etc.. At the

same time, software companies throughout the world are not following the same convention about the coding structures. So, automated development of coding is highly infeasible at this moment. Our future strategy will be the development of fully agent based corporate environment system design without using human intervention.

## REFERENCES

- [1] U. Deshpande, A. Gupta and A. Basu, "Multi-agent modeling and fuzzy task assignment for real time operations in a supply chain," in *Multiagent Based Supply Chain Management*, B. Chaib-draa and J. Muller Ed., Springer series in Computational Intelligence, Vol. 28, 2006.
- [2] M. J. Wooldridge and N. R. Jennings, "Software Engineering with agents: pitfalls and pratfalls," *IEEE Internet Computing*, May-June 1999.
- [3] A. Tveit, "A survey of Agent-Oriented Software Engineering," in Proceedings of First NTNU Computer Science Graduate Student Conference, Norwegian University of Science and Technology, May 2001.
- [4] A. S. Rao and M. P. Georgeff, "BDI agents: from theory to practice," in Proceedings of First International Conference on Multi-Agent Systems, San Francisco, CA, USA, 1995.
- [5] D. Luo, L. Cao, J. Ni and L. Liu, "Building Agent Service Oriented Multi-Agent Systems," KES AMSTA 2007, Wroclaw, Poland, May-June 2007.
- [6] J. Debenham and S. Simoff, "A Model for Informed Negotiating Agents," KES AMSTA 2007, Wroclaw, Poland, May-June 2007.
- [7] Z. Xiao-fan, L.O. Zhi-cheng and L. Jian-xiong, "The Study of Multi-Agent Network Flow Architecture for Application Performance Evaluation," KES AMSTA 2007, Wroclaw, Poland, May-June 2007.
- [8] S. Saha and S. Sen, "Towards Efficient Outcomes in Multi-issue Negotiation," BAsEWEB'06 Proceedings, Hakodate, Japan, May 2006.
- [9] G. Fernandes, L. Ayres, J. Carvalho and V. Furtado, "An Agent-based Approach for Explaining Web Service Composition via Problem Solving Methods in OWL-S," BAsEWEB'06 Proceedings, Hakodate, Japan, May 2006.
- [10] V. Podobnik, G. Jezic and K. Trzec, "An Agent-mediated Electronic Market of Semantic Web Services," BAsEWEB'06 Proceedings, Hakodate, Japan, May 2006.
- [11] X. Xue, J. Lu, Y. Wang and Q. Shen, "Towards an Agent-based Negotiation Platform for Cooperative Decision-Making in Construction Supply Chain," KES AMSTA 2007, Wroclaw, Poland, May-June 2007.
- [12] [www.iau.dtu.dk/~jj/pubs/logic.pdf](http://www.iau.dtu.dk/~jj/pubs/logic.pdf)
- [13] [www.fuzzy-logic.com/Ch1.htm](http://www.fuzzy-logic.com/Ch1.htm)

---

Received: September 01, 2008

Revised: September 20, 2008

Accepted: September 26, 2008

© Anirban Kundu; Licensee *Bentham Open*.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.