

**Knowledge Discovery Based Simulation System in  
Construction**

**Emad Elwakil**

A Thesis

In the Department

of

Building, Civil and Environmental Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Building Engineering) at

Concordia University

Montreal, Quebec, Canada

June, 2011

© Emad Elwakil, 2011

**CONCORDIA UNIVERSITY**  
**SCHOOL OF GRADUATE STUDIES**

This is to certify that the thesis prepared

By:                   Emad Elwakil

Entitled:           Knowledge Discovery Based Simulation System in Construction

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Building Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

\_\_\_\_\_ Chair  
Dr. C. W. Trueman

\_\_\_\_\_ External Examiner  
Dr. A.R. Fayek

\_\_\_\_\_ External to Program  
Dr. L. Kadem

\_\_\_\_\_ Examiner  
Dr. O. Moselhi

\_\_\_\_\_ Examiner  
Dr. A. Bagchi

\_\_\_\_\_ Supervisor  
Dr. T. Zayed

Approved by \_\_\_\_\_  
Dr. K. Ha-Huy, Graduate Program Director

June 9, 2011

\_\_\_\_\_  
Dr. Robin A.L. Drew, Dean  
Faculty of Engineering & Computer Science

# **ABSTRACT**

## **Knowledge Discovery Based Simulation System in Construction**

**Emad Elwakil, Ph.D.**

**Concordia University, 2011**

Uncertainty is an entrenched characteristic of most construction projects. Typically, probability distributions are utilized to accommodate uncertainty when estimating duration of project's activities. Distributions are fitted, based on the collected data from construction projects, to estimate activity durations, to assess productivity and cost, and to identify resource bottlenecks using simulation. The subjectivity in selecting these fitted probability distributions is an imprecise process and may significantly affect simulation outputs. Most research works in simulating construction operations has focused predominantly on modeling and has neglected to study the effect of subjective variables on simulation process. Therefore, there is a need for a system, which: (1) handles uncertainty, fuzziness, missing data, and outliers in input data, (2) effectively utilizes historical data, (3) models the effect of qualitative and quantitative variables on the simulation process, (4) enhances simulation modeling capabilities, and (5) optimizes simulation system output(s).

The main objective of this research is to develop a knowledge discovery based simulation system for construction operations, which achieves the abovementioned necessities. This system comprises three stages: (i) a Knowledge Discovery Stage (KDS), (ii) a Simulation Stage (SS), and (iii) an Optimization Stage (OS). In the KDS, raw data are prepared for

the SS where patterns, which represent knowledge implicitly stored or captured in large databases, are extracted using Fuzzy K-means technique. During the KDS, the effect of qualitative and quantitative variables on construction operation(s) is modelled using Fuzzy Clustering technique. This stage improves the efficiency of data modeling by 10% closer to real data. The movement of units is modeled in the SS where the interaction between flow units and idle times of resources can be examined to discover any bottlenecks and estimate the operation's productivity and cost. The OS, using Pareto ranking technique, assists in selecting and ranking feasible productivity-cost solution(s) for diverse resource combinations under different conditions.

An automated general purpose construction simulation language (KEYSTONE) is developed using C#. The developed system is validated and verified using several case studies with sound and satisfactory results, i.e. 4% - 11% digression. The developed research/system benefits both researchers and practitioners because it provides robust simulation modeling tool(s) and optimum resources allocation for construction operations.

## **ACKNOWLEDGEMENT**

First and foremost, I am very grateful to God for keeping me blessed and granting me the ability to follow through and achieve my goal. I am also thankful to the almighty for allowing me the opportunity to complete my research and obtain my Doctorate degree. Secondly, I would like to thank my parents, Abd Elmoeti Elwakil and Hekmat Elwakil, for their great support and encouragement throughout my life. They have played an instrumental part in my development as a person and I owe them a debt of gratitude.

I would like to express special gratitude to my research advisor Dr. Tarek Zayed for his instructive advice and guidance throughout my research. I would not have been able to conclude my work in isolation from his mentoring, suggestions, and timely comments. Dr. Zayed has been a great source of inspiration throughout this process. He taught me to be helpful and collaborative by setting an excellent example. He also encouraged me to smile at the face of adversities and to maintain a positive attitude in times of frustration.

I would also like extend my gratitude to Dr. Osama Moselhi and Dr. Sabah Alkass for their help and support. Dr. Moselhi's words "work hard and smart" has been engrained in my mind and heart; a motto for life.

I wish to thank, my wife Salma from the bottom of my life for her patient care and loving concern. She has been my source of strength and encouragement throughout this journey. I am forever indebted to her for putting up with my busy schedule and long hours of work. She graced our home with much love and compassion and gave me the treasure of having an adorable family. Last but not least, I want to express my love and adoration for my little angles, Norhan and Norjhan and my young heroes, Adel II and Emad II.

# Table of Contents

<b>List of Figures .....</b>	<b>viii</b>
<b>List of Tables .....</b>	<b>x</b>
<b>List of Abbreviations .....</b>	<b>xi</b>
<b>Chapter 1: Introduction .....</b>	<b>1</b>
1.1 PROBLEM STATEMENT AND RESEARCH MOTIVATION .....	1
1.2 THESIS OBJECTIVES.....	3
1.3 SUMMARY OF THE RESEARCH METHODOLOGY .....	4
1.4 THESIS ORGANIZATION.....	5
<b>Chapter 2: Literature Review.....</b>	<b>6</b>
2.1 CHAPTER OVERVIEW.....	6
2.2 THE STATE OF THE ART REVIEW OF CONSTRUCTION SIMULATION .....	6
2.2.1 <i>Construction Simulation Systems</i> .....	8
2.3 DATA MINING AND KNOWLEDGE DISCOVERY .....	10
2.3.1 <i>Data Mining &amp; Knowledge Discovery systems and models in construction</i> .....	15
2.4 APPROACHES TO MODEL BUILDING AND KNOWLEDGE DISCOVERY.....	20
2.4.1 <i>Ordinary Statistics</i> .....	21
2.4.2 <i>Nonparametric Statistics</i> .....	21
2.4.3 <i>Linear Regression in Statistical Models</i> .....	22
2.4.4 <i>Cluster Analysis</i> .....	23
2.4.5 <i>Artificial Neural Networks (ANN)</i> .....	24
2.4.6 <i>Fuzzy SQL System</i> .....	26
2.5 FUZZY KNOWLEDGE BASE BUILDING AND DESIGN.....	27
2.5.1 <i>Membership Value Assignments</i> .....	28
2.5.2 <i>Fuzzy Rule Induction and Fuzzy Models</i> .....	30
2.5.2.1 <i>The Vocabulary of Rule Induction</i> .....	30
2.5.3 <i>Knowledge Base Generation Cycle</i> .....	32
2.5.4 <i>The Rule Induction Algorithm</i> .....	32
2.6 MULTI OBJECTIVE OPTIMIZATION .....	33
2.6.1 <i>Pareto Ranking</i> .....	35
2.6.2 <i>Fitness Sharing</i> .....	35
2.6.3 <i>Elitism</i> .....	36
2.7 SUMMARY AND LIMITATIONS OF LITERATURE REVIEW .....	37
<b>Chapter 3: Research Methodology .....</b>	<b>39</b>
3.1 CHAPTER OVERVIEW.....	39
3.2 LITERATURE REVIEW .....	39
3.3 SYSTEM DEVELOPMENT.....	41
3.3.1 <i>Knowledge Discovery Stage (KDS)</i> .....	41
3.3.1.1 System Definition Phase.....	42
3.3.1.2 Data Identification Phase .....	42
3.3.1.3 Data Preprocessing Phase .....	44
3.3.1.4 Data Mining Engine Phase.....	48
3.3.2 <i>Simulation Stage (SS)</i> .....	57

3.3.2.1	Simulation Engine Design Phase .....	58
3.3.2.2	Validation Phase.....	58
3.3.3	<i>Optimization Stage (OS)</i> .....	59
3.3.3.1	Sensitivity Analysis on the Resource Level Phase.....	59
3.3.3.2	Sensitivity Analysis on the Project Level Phase .....	60
3.3.3.3	Selection and Ranking of the Optimal Solutions Phase.....	60
3.4	DATA COLLECTION .....	64
3.5	VERIFY AND VALIDATE THE SYSTEM USING REAL DATA .....	64
3.6	THE KEYSTONE LANGUAGE AND SYSTEM AUTOMATION .....	64
3.7	VERIFY AND VALIDATE THE SYSTEM USING A CASE STUDY.....	65
<b>Chapter 4: Data Collection.....</b>		<b>66</b>
4.1	CHAPTER OVERVIEW .....	66
4.2	DATA COLLECTION .....	66
4.2.1	CASE STUDY PROJECT .....	66
4.2.2	DATA COLLECTION PROCESS .....	66
4.2.3	THE COLLECTED DATA .....	67
<b>Chapter 5: Results and Analysis.....</b>		<b>74</b>
5.1	CHAPTER OVERVIEW.....	74
5.2	BUILDING THE KNOWLEDGE DISCOVERY STAGE (KDS).....	74
5.2.1	DATA CLEANING .....	75
5.2.1.1	DATA CLUSTERING.....	75
5.2.1.2	MISSING DATA.....	77
5.2.2	COMPARISON BETWEEN DATA MODELS BEFORE AND AFTER CLEANING .....	77
5.2.2.1	CASE I: DATA MODELING AFTER REMOVING MISSING DATA AND OUTLIERS .....	78
5.2.2.2	CASE II: DATA MODELING AFTER FILLING MISSING DATA AND KEEPING OUTLIERS.....	79
5.2.3	BUILDING DATA MINING ENGINE .....	80
5.2.3.1	SELECTION OF VARIABLES .....	81
5.2.3.2	FUZZY SETS (FUZZIFICATION) .....	86
5.2.3.3	FUZZY RULES INDUCTION .....	89
5.2.3.4	FUZZY KNOWLEDGE BASE (FKB) .....	91
5.2.3.5	VALIDATION .....	91
5.3	SIMULATION STAGE (SS) .....	95
5.3.1	SIMULATION ENGINE DESIGN AND BUILDING PHASE .....	96
5.3.1.1	DURING THE PLANNING STAGE .....	96
5.3.1.2	DURING THE CONSTRUCTION STAGE .....	97
5.3.2	VALIDATION PHASE.....	101
5.4	OPTIMIZATION STAGE (OS) .....	103
5.4.1	SENSITIVITY ANALYSIS ON PROJECT LEVEL PHASE .....	104
5.4.2	SELECTION AND RANKING OF THE OPTIMAL SOLUTIONS PHASE.....	109
<b>Chapter 6: KEYSTONE Language and System Automation.....</b>		<b>113</b>
6.1	CHAPTER OVERVIEW.....	113
6.2	KEYSTONE LANGUAGE'S CORE .....	113
6.2.1	PROJECT CLASS .....	115
6.2.1.1	METHODS.....	117
6.2.2	OPERATION CLASS.....	128
6.2.2.1	METHODS.....	128
6.2.3	PROCESS CLASS.....	133
6.2.3.1	METHODS.....	134
6.2.4	AUXILIARY INSTANCE CLASS.....	141

6.2.4.1	METHODS .....	141
6.3	CRITERIA FOR THE SELECTION OF PROGRAMMING DEVELOPMENT TOOLS .....	144
6.4	SYSTEM'S ARCHITECTURE .....	145
6.5	DATA FLOW OF THE DEVELOPED SYSTEM .....	147
6.6	GRAPHICAL USER INTERFACE (GUI) .....	148
6.7	APPLYING THE DEVELOPED SYSTEM ON CASE STUDY .....	148
6.7.1	INPUT THE PROJECT DATA .....	148
6.7.2	TRAINING MODULE .....	149
6.7.3	SIMULATION MODULE .....	152
6.7.4	OPTIMIZATION MODULE .....	152
6.7.5	REPORTING MODULE .....	155
<b>Chapter 7: Conclusions and Recommendations .....</b>		<b>160</b>
7.1	SUMMARY AND CONCLUSIONS .....	160
7.2	RESEARCH CONTRIBUTIONS .....	162
7.3	RESEARCH LIMITATIONS .....	163
7.4	RECOMMENDATIONS FOR FUTURE RESEARCH .....	164
<b>References .....</b>		<b>167</b>
<b>Appendix A .....</b>		<b>176</b>
<b>Appendix B .....</b>		<b>181</b>



## List of Figures

Figure 2-1 Literature review chapter’s organization chart .....	7
Figure 2-2 Procedure of traditional construction simulation study (Adopted from Wang, 2005) .....	8
Figure 2-3 Event generation using a fuzzy expert system (Shaheen et al. 2005) .....	11
Figure 2-4 Data mining as a step in the process of knowledge discovery (Han and Kamber, 2006) .....	13
Figure 2-5 Architecture of a typical data mining system (Han and Kamber, 2006).....	15
Figure 2-6 Knowledge discovery in databases approach (Soibelman and Hyunjoo, 2002) .....	17
Figure 2-7 Data fusion and modeling methodology for construction management knowledge discovery (Soibelman et al. 2004).....	18
Figure 2-8 Design and schedule data integration system architecture (Shen et al. 2004)	19
Figure 2-9 Conceptual frameworks for capturing knowledge in construction projects (Kivrak et al. 2008) .....	20
Figure 2-10 Basic architecture of ANN (Cho et al. 1999).....	26
Figure 2-11 Rule induction process (Cox, 2005).....	34
Figure 2-12 The basic knowledge-base generation cycle (Cox, 2005).....	34
Figure 2-13 A set of points and the first non-dominated solution (Branke et al. 2008) ...	36
Figure 3-1 Overview of the research methodology .....	40
Figure 3-2 Overview of the system development .....	43
Figure 3-3 The architecture of the data collection phase .....	44
Figure 3-4 Architecture of the data preprocessing phase.....	46
Figure 3-5 Architecture of the data mining engine phase.....	51
Figure 3-6 Neural network as a tool to determine membership functions.....	52
Figure 3-7 Framework of the simulation engine.....	61
Figure 3-8 The selection and ranking procedure .....	63
Figure 4-1 The considered variables in the present study.....	67
Figure 4-2 Model of the concrete pouring operation .....	69
Figure 4-3 The data sets of concrete pouring operation .....	72
Figure 4-4 Statistical parameters and probability density functions.....	72
Figure 4-5 Statistical parameters and probability density functions for temperature.....	73
Figure 5-1 (4-D) Plot of data clusters .....	76
Figure 5-2 Fuzzy curve for temperature .....	83
Figure 5-3 Estimated vs. actual task duration .....	95
Figure 5-4 Concrete pouring operation model.....	101
Figure 5-5 Simulation model code.....	103
Figure 5-6 Simulation process results.....	104
Figure 5-7 Productivity comparison between proposed system and other systems.....	106
Figure 5-8 Temperature sensitivity analysis .....	111
Figure 5-9 Floor level sensitivity analysis .....	112
Figure 5-10 Labor percentage sensitivity analysis.....	112
Figure 6-1 KEYSTONE language and system automation chapter’s organization chart	113

Figure 6-2 KEYSTONE core language framework .....	115
Figure 6-3 System component interaction .....	116
Figure 6-4 Project class components .....	118
Figure 6-5 Operation class components.....	129
Figure 6-6 Process class components .....	136
Figure 6-7 Auxiliary class components .....	142
Figure 6-8 System architecture and data flow .....	146
Figure 6-9 Graphical user interface (GUI).....	150
Figure 6-10 Create and define a server .....	151
Figure 6-11 Create and define an auxiliary.....	152
Figure 6-12 Graphical representation of the project .....	153
Figure 6-13 Actual vs. predicted concrete pouring durations.....	154
Figure 6-14 Simulation module in the planning stage .....	156
Figure 6-15 Sensitivity analysis results .....	157
Figure 6-16 Ranking and selecting the optimum solution(s).....	158
Figure 6-17 Simulation results report (Template) .....	159
Figure A-1 Fuzzy curve for humidity .....	177
Figure A-2 Fuzzy curve for precipitation .....	177
Figure A-3 Fuzzy curve for wind speed .....	178
Figure A-4 Fuzzy curve for gang size.....	178
Figure A-5 Fuzzy curve for labor percentage .....	179
Figure A-6 Fuzzy curve for floor level .....	179
Figure A-7 Fuzzy curve for method.....	180

## List of Tables

Table 3-1 Sensitivity analysis matrix.....	62
Table 4-1 Variables description.....	68
Table 4-2 Variables affecting concrete pouring process.....	69
Table 4-3 Variables affecting the hauling process.....	69
Table 4-4 Variables affecting loading process.....	70
Table 4-5 Variables affecting return process.....	71
Table 5-1 Fuzzy membership values.....	76
Table 5-2 The calculated centroids of data sets.....	78
Table 5-3 Comparison with filling in the missing values.....	78
Table 5-4 Best subset analysis case I.....	80
Table 5-5 Best subset analysis case II.....	82
Table 5-6 Fuzzy curve and mean square error calculations.....	83
Table 5-7 Variables ranking using fuzzy average method.....	84
Table 5-8 Variables ranking using ANN.....	85
Table 5-9 Variables ranking using regression method.....	85
Table 5-10 Variables describing data points.....	87
Table 5-11 Fuzzy cluster membership values and task durations.....	90
Table 5-12 A sample of candidate rules.....	92
Table 5-13 A sample of the rules' effectiveness degrees.....	93
Table 5-14 A sample of a fuzzy knowledge base.....	94
Table 5-15 A sample of fuzzy knowledge base validation results.....	95
Table 5-16 The range of variables affecting the concrete pouring process.....	97
Table 5-17 Predicted task durations.....	97
Table 5-18 Simulation event list.....	98
Table 5-19 Concrete pouring process states.....	98
Table 5-20 Predicted task duration during construction.....	99
Table 5-21 Simulation event list.....	99
Table 5-22 Process duration information.....	101
Table 5-23 Resources information.....	102
Table 5-24 Sensitivity analysis for Web cyclone and Ezstrobe.....	105
Table 5-25 Sensitivity analysis on the project level.....	107
Table 5-26 Comparison between the sensitivity analysis of the proposed system and that of web cyclone.....	108
Table 5-27 Candidate solutions.....	111
Table 5-28 Candidate solutions comparison (Rank 1).....	111
Table 5-29 Classification of solutions (Rank 2).....	111
Table 5-30 Classification of solutions (Rank 3).....	111
Table B-1 Fuzzy curve and mean square error calculations for humidity.....	182
Table B-2 Fuzzy curve and mean square error calculations for precipitation.....	183
Table B-3 Fuzzy curve and mean square error calculations for wind speed.....	184
Table B-4 Fuzzy curve and mean square error calculations for gang size.....	185
Table B-5 Fuzzy curve and mean square error calculations for labor percentage.....	186
Table B-6 Fuzzy curve and mean square error calculations for floor level.....	187
Table B-7 Fuzzy curve and mean square error calculations for method.....	188

## List of Abbreviations

C#	C sharp programming language
ANN	Artificial Neural Network
SQL	Structured Query Language
KDD	Knowledge Discovery in Database
OLTP	On-Line Transaction Processing
DSS	Decision Support System
CMDSS	Construction Management Decision Support System
$Y_i$	Value of response variable
$\beta_0$	Regression parameter
$\beta_1$	Regression parameter
$\varepsilon_i$	Random error
BPNN	Back propagation neural networks
FAM	Fuzzy Associative Memory
E	Degree of effectiveness
$\mu_X$	Fuzzy set membership value
$v_k$	Cluster fuzzy membership value
m	The fuzzifier
k	Number of clusters
d	The distance between centroid $v_k$ and object $x_i$
M	Data points
b	The width of the influence interval
$c_i$	Fuzzy curve point
MSE	Mean Square error
$C_i$	Cluster's number
$R_{ij}$	Assumed membership function values
AIP	Average invalidity percent
AVP	Average Validity Percent
RMSE	Root Mean Squared Error

MAE	Mean Absolute Error
EET	End of event time
GUI	Graphical User Interface (GUI)

# **Chapter 1: Introduction**

## **1.1 Problem Statement and Research Motivation**

Simulation has been considered an effective analytical tool to handle uncertainty with probabilistic characteristics. It is a dominant tool that can be used by construction companies in several tasks, such as productivity measurement, risk analysis, resource planning, and design/analysis of construction methods (Law and Kelton, 2000; Sawhney et al. 1998). Complex construction processes are difficult to analyze and optimize using standard mathematical methods because of their uncertain characteristics. Simulation is a sound potential alternative to evaluate these complex systems with multiple potential benefits (Ioannou and Martinez, 1996). Several construction simulation tools have been developed for the construction industry. However, the use of simulation in planning and designing construction operations is limited (Hajjar and AbouRizk, 2002; Halpin and Martinez, 1999, Tucker et al., 1998, Huang and Halpin, 1994) because simulation models and their results cannot be presented effectively enough to be understood by typical decision makers in the construction industry (Kamat and Martinez, 2003).

Simulation systems usually deal with random variables represented by probability distributions, which are fitted based on construction data, to estimate productivity and cost as well as identify resource bottlenecks of construction operations (Wang, 2005). Experts typically estimate an approximate probability distribution for tasks based on linguistic terms without carefully reflecting on the probability values (Kim and Fishwick, 1997). The subjectivity involved in selecting these fitted probability distributions, which

represent the collected data, significantly affects simulation outputs and does not lead to the superlative solution or representation of such data.

Unforeseen subjective factors, such as weather changes, equipment accidents, labor inefficiency and resource delivery, may result in a great deal of uncertainty that affects construction processes and may lead to project/activity failure (Zhang et al. 2003). For example, weather can adversely affect a construction process in many ways. In severe weather conditions, certain activities and sometimes the entire project can be halted. Poor weather conditions can slow down a construction process by lowering the productivity of construction crews and equipment (Moselhi et al. 1997). Most research works in simulating construction processes have primarily focused on simulation modeling with little emphases on studying the qualitative variables that affect the simulation process. Several examples are shown in: Halpin 1973, Paulson et al. 1987, Liu 1991, Odeh, et al. 1992, Martinez and Ioannou 1994, Huang et al. 1994, Martinez 1998, and AbouRizk and Mohamed 2000. Therefore, these subjective variables should be carefully considered when designing simulation models/tools for construction operations.

It has been estimated that the amount of data stored in world's databases doubled every 20 months in which it was reported that (Han and Kamber, 2006): *"If users believe data are dirty, they are unlikely to trust the results of any data mining that has been applied to it. Furthermore, dirty data can cause confusion for the mining procedure which leads to unreliable output"*. The explosive growth of many business, government, and scientific databases has begun to far outpace our ability to interpret data. Such amount of data clearly overwhelms traditional methods of data analysis, such as spreadsheets. Knowledge is a valuable asset of any organization, particularly in construction.

Corporations must incorporate both continuous improvement and organizational learning to improve construction business results (Fisher, 1997). Missing data and outliers, which make data often incomplete, noisy, and inconsistent, cause exigent problems to construction organizations. Missing values in construction databases should be filled and inconsistencies in data should be rectified while identifying outliers. It is clear that there is a significant need for a new generation of techniques and tools with the ability to automatically assist humans in analyzing the mountains of available construction data searching for useful knowledge (Soibelman and Hyunjoo, 2002). There is also an urgent need for interactive simulation systems that provide the user with capabilities to interact with running models and control the simulation process (Bishop and Balci, 1990).

The outputs of existing simulation systems, which are obtained by performing sensitivity analysis for possible resource combinations, are manually manipulated to identify the optimal resource combinations. Therefore, obtaining the optimal solution becomes a lengthy and difficult process (Lee et al, 2010). Decision makers often do not have the means, training and/or time to verify and validate the outputs of simulation models (Ioannou and Martinez, 1996). The validation of simulation results makes it credible, acceptable and usable as a decision making tool (Law and Kelton, 2000). Therefore, developing model(s)/tool(s) to optimize resource combinations is paramount for construction decision makers.

## **1.2 Thesis Objectives**

The overall objective of this research is to develop a knowledge discovery based construction simulation system for construction operations. This main objective can be broken down to the following sub-objectives:



- Develop a strategy and methodology to treat simulation input data problems, i.e. uncertainty, fuzziness, missing, and outliers, and to generate more precise and valuable knowledge from databases of construction companies.
- Model the uncertainty due to the fuzziness of input variables and model the effect of qualitative and quantitative variables on the simulation process.
- Develop a model to optimize simulation results due to diverse resource combinations.
- Design and build interactive simulation software to implement the developed models and methodology.
- Verify and validate the developed system/models using real case study (ies).

### **1.3 Summary of the Research Methodology**

The methodology of this research comprises several steps as follows:

- 1- The literature review covers the state of the art in construction simulation, data mining and knowledge discovery, approaches to model building and knowledge discovery, building and design of Fuzzy knowledge bases as well as multi objective optimization.
- 2- System development, which includes the design and building of the Knowledge Discovery Stage (KDS), Simulation Stage (SS), and Optimization Stage (OS). The KDS cleans data, prepares data to be modeled, models the hidden pattern in data, models the affect of qualitative and quantitative variables on the process's duration, builds the data mining engine, and builds the Fuzzy knowledge base. The SS builds simulation models for construction operations including operations, processes, auxiliaries and servers, and models the interaction of the KDS and the

OS. The OS develops simulation analysis for various resource combinations and selects the optimum solution(s) of resource distribution.

- 3- Data are collected from a multi-story building case study in order to effectively verify and validate the developed system/models.
- 4- A general purpose simulation language (KEYSTONE) is developed to build simulation environment platform for modeling construction operations. The KEYSTONE simulation language is developed using the C# package.
- 5- The developed models, methodology, and the KEYSTONE are verified and validated using the case study.

#### **1.4 Thesis Organization**

This thesis consists of seven chapters and two appendices. Chapter one contains the problem statement and research motivations, thesis objectives, and summary of the research methodology. Chapter two provides a detailed literature review that describes: the previous work done in this field including the different models, methodologies and systems currently being used. Chapter three discusses a detailed description of the current developed methodology. Chapter four presents the data collection procedure and analysis. Chapter five is a case study and analysis of results that show the application of the proposed system on a real world case study and its validation. Chapter six focuses on the design and build a general purpose construction simulation language, the computer implementation of the developed system as a software package and validation on a case study. Chapter seven summarizes the research and contributions, and presents suggestions for future work. Appendix A contains the fuzzy curves and mean square error calculations. Appendix B gives the developed fuzzy curves.

# **Chapter 2: Literature Review**

## **2.1 Chapter Overview**

The main aim of this chapter is to provide a comprehensive literature review. Figure 2.1 illustrates the organization chart of this literature review chapter, indicating that it contains six main sections. The first section contains a state of the art review in construction simulation, including a construction simulation overview and the construction simulation systems currently in use. The second section introduces data mining and a knowledge discovery overview and includes the currently-used data mining and knowledge discovery systems and models. The third section contains a variety of approaches to model building and knowledge discovery, among them ordinary statistics, nonparametric statistics, linear regression, cluster analysis, artificial neural networks (ANN), and fuzzy SQL systems. Fuzzy knowledge base building and design; including membership functions value assignments, fuzzy rule induction, fuzzy models, and the knowledge-base generation cycle are presented in the fourth section. The fifth section introduces multi-objective optimization including Pareto ranking and selecting optimum solution(s). The sixth section summarizes the limitations of the literature.

## **2.2 The State of the Art Review of Construction Simulation**

Simulation is one of the most widely used operations research and management science techniques (Law and Kelton, 2000). Construction simulation is a powerful tool that can be used by a construction company for several tasks, such as productivity measurement, risk analysis, resource planning, and the design and analysis of construction methods (Sawhney et al. 1998). Traditionally, as shown in Figure 2.2, research studies on

simulation include field data collection, process modelling, process simulation and sensitivity analysis to estimate productivity and suggest alternatives (Wang, 2005). Computer simulation is the process of designing a mathematical-logical model to present a real system and then experimenting with the model on a computer system (Pritsker, 1986).

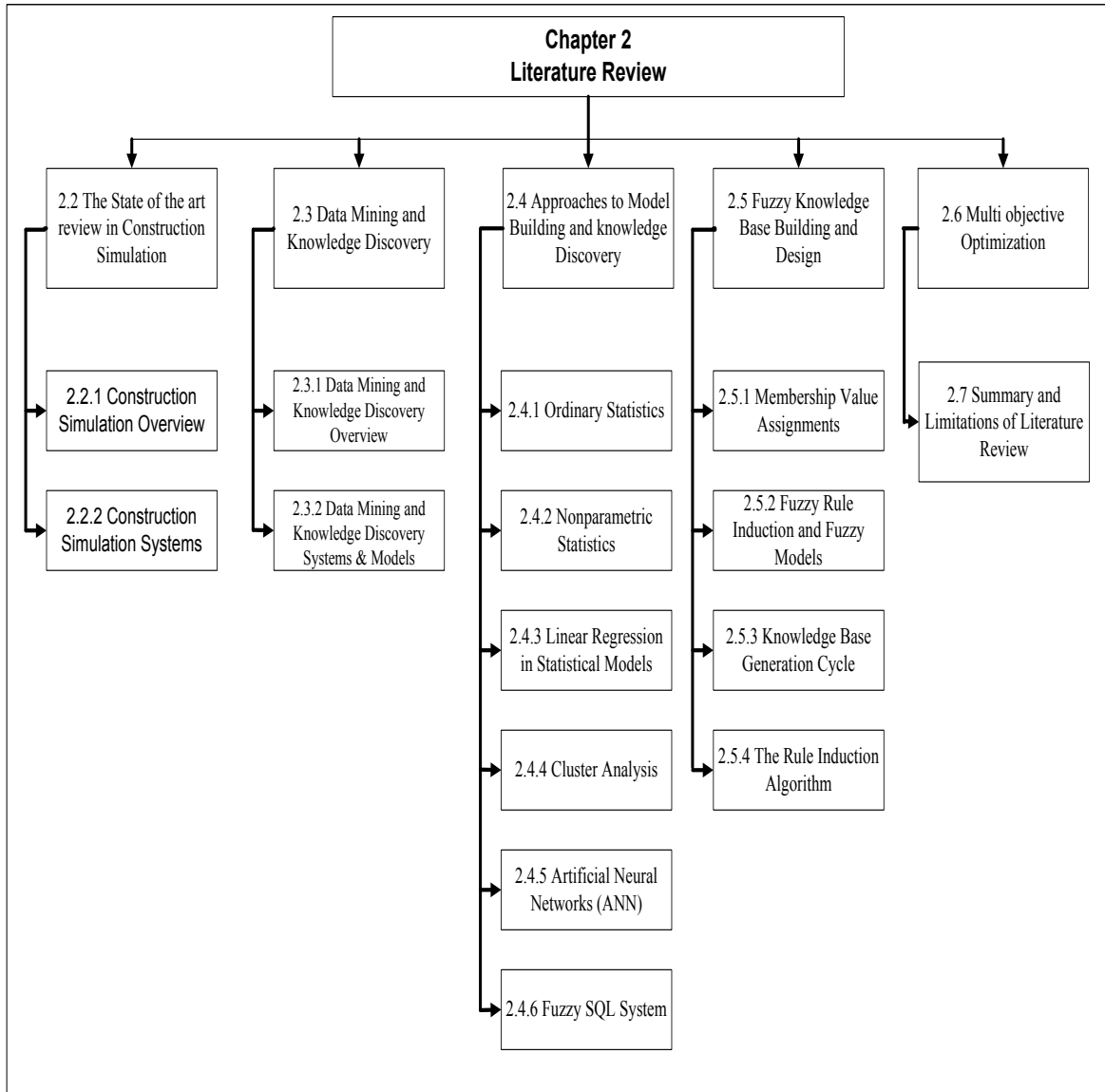


Figure 2-1 Literature review chapter's organization chart

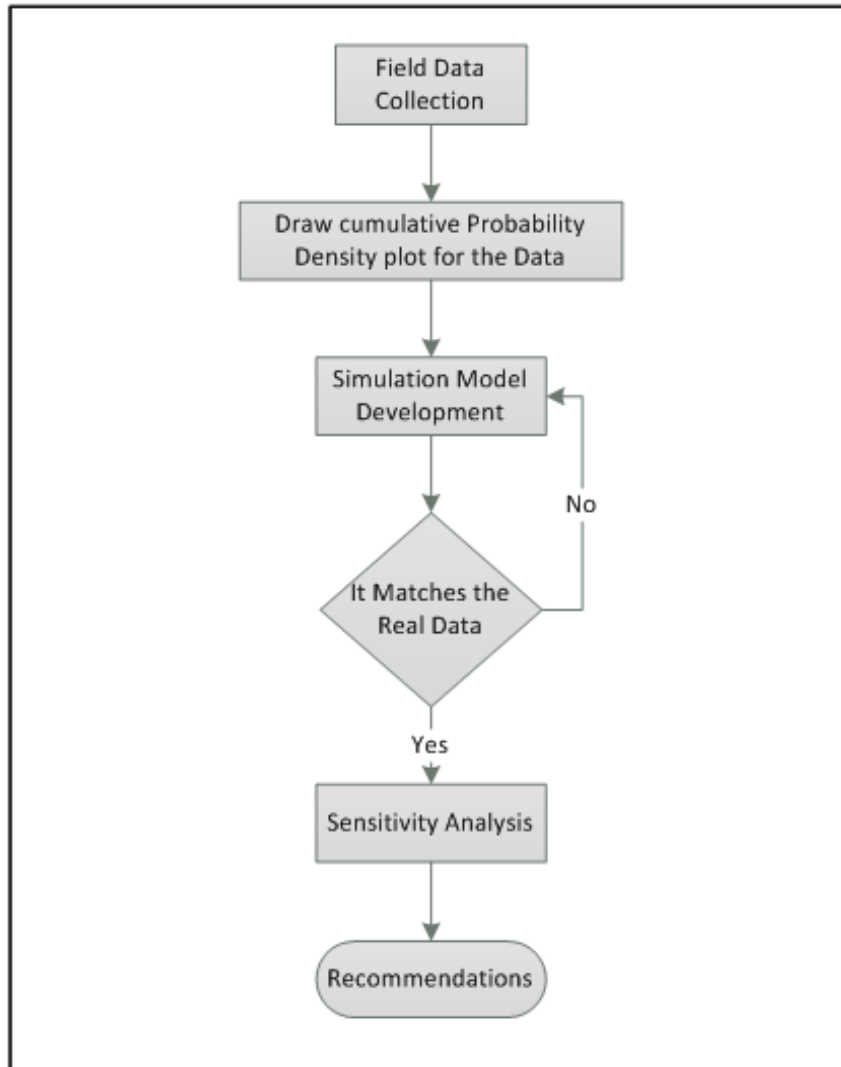


Figure 2-2 Procedure of traditional construction simulation study (Adopted from Wang, 2005)

### 2.2.1 Construction Simulation Systems

Simulation is a valuable tool used to apply project scenarios, establish a feasible work plan and assess resource allocation on an operation (Halpin, 1977). Early simulation systems required practitioners to build the code using a programming language such as FORTRAN, Pascal, Basic, etc (Ruwanpura, 2001). In the 1970s, Halpin applied simulation science on construction operations. Halpin (1973) developed CYCLONE modeling elements that simplified the simulation and modeling process for practitioners

with limited simulation background (Swahney et al. 2003). Many simulation systems were built on CYCLONE; the progression as follows: Paulson et al. (1987) used the time data collected from video tapes in the construction field to develop their system (**Insight**). Insight was developed make powerful simulation analysis and design techniques available on microcomputers at the field office level. Liu (1991) developed **COOPS** to deal with direct-event simulation. It is similar in design and function to CYCLONE but with more graphical features. Odeh et al. (1992) developed **CIPROS** to make it possible for users to integrate construction plans, project design drawings, and specifications to the network of construction processes. CIPROS uses the knowledge of common construction resources and design features in the form of class hierarchies of the resources and components.

Martinez and Ioannou (1994) developed **STROBOSCOPE** to deal with the uncertainty of any aspect, such as quantities of resources produced or consumed (and not just time). Huang et al. (1994) developed **DISCO** to deal with the complexity of modeling and simulating construction processes by providing an interactive graphical environment. Martinez (1998) developed **EZSTROBE** to fill in the complexity of model development that was lacking in existing simulation systems, especially **STROBOSCOPE**. EZSTROBE is based on Activity Cycle Diagrams, and employs the three-phase activity scanning paradigm. AbouRizk and Mohamed (2000) developed **Simphony** as an integrated environment for building special purpose simulation tools for modeling construction systems. It provides various services that enable a developer to easily control different behaviors in the developed tool, such as simulation behaviors, graphical representation, statistics, and animation. It allows the user to build flexible and user

friendly tools in a relatively short time. **Simphony** provides the modeler with graphical elements that abstract the real world system and supports the modularity and interchangeability of components from different templates. Although **Simphony** essentially makes the process of developing simulation toolkits for construction cost-effective and efficient, it neglects the pre-simulation data processing and optimization of simulation output.

Shaheen et al. (2005) developed a framework for integrating fuzzy expert systems and discrete event simulation as shown in Figure 2.3. This framework can then be utilized to enhance the input modeling process. The framework predicts activity output (i.e. duration) using a fuzzy expert system. The proposed integration is designed to provide real-time prediction of the activity output (i.e. duration) by capturing and modeling the changes in the factors affecting the activity output whenever the simulation time advances. This system provides integration between fuzzy numbers and discrete event simulation. However; it depends only on expert opinions, which can be very scarce because of the unique nature of construction projects. Moreover, this framework was designed with the assumption that the experts are confident and they are 100 % certain about the information provided.

### **2.3 Data Mining and Knowledge Discovery**

Data mining refers to process of extracting or “mining” knowledge from large amounts of data (Han and Kamber, 2006). Data mining is the process of finding patterns that lie within large collections of data sets. Contrary to more traditional data analysis methods, which begin with a hypothesis and then test the hypothesis based upon the data, data mining deals with the problem from the opposite direction to discover the patterns. Data

mining is discovery-driven rather than assumption-driven (Radivojevic et al. 2003). Data mining has attracted a great deal of attention in the information industry and in society due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge (Han and Kamber, 2006), as well as the need of users for a type of control over how their personal data is used. In general, data mining objectives can be placed into two categories, Descriptive and Predictive.

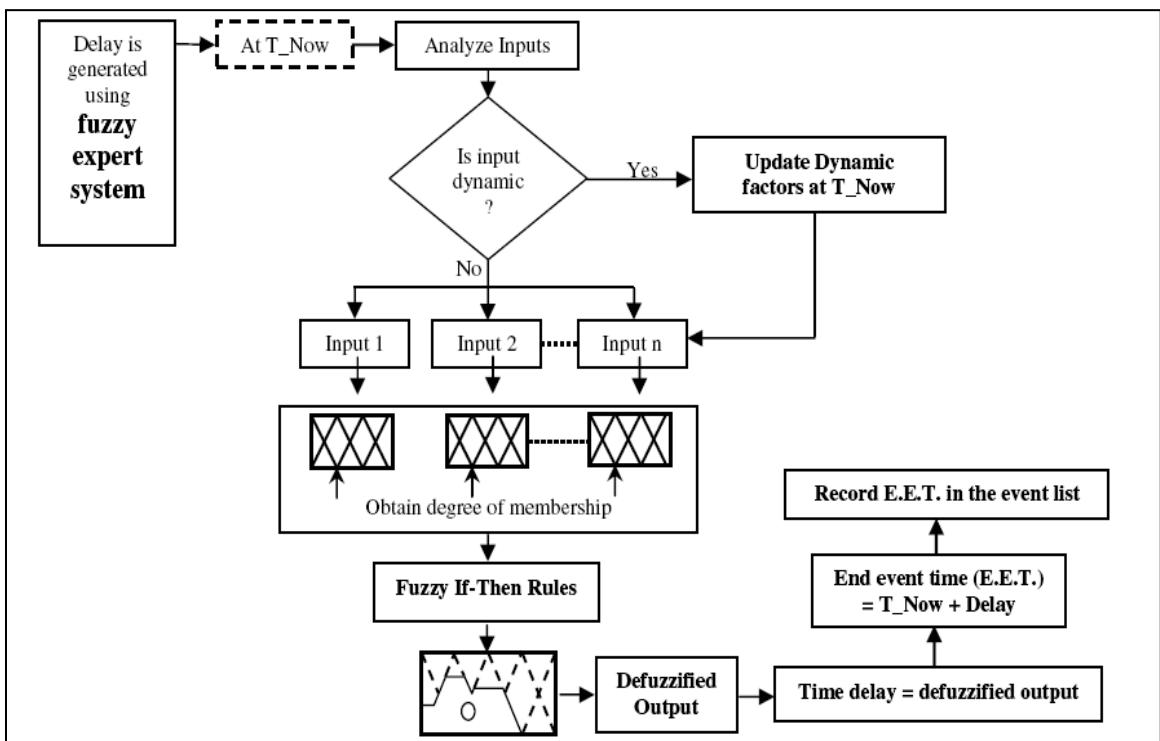


Figure 2-3 Event generation using a fuzzy expert system (Shaheen et al. 2005)

The goal of descriptive data mining is to find general patterns or properties of elements in data sets that involve aggregate functions such as mean, variance, count, sum, etc. Descriptive data mining reports patterns about the data itself. Predictive data mining attempts to infer meaning from data in order to create a model that can be used to predict



future data. This is often done by grouping data elements based on similarities in the attributes of data sets. The common attributes should be a reasonable predictor for the given result. Another important concept is the difference between supervised and unsupervised learning. Supervised learning takes place when the data has been pre-classified (De Raedt et al. 2001). Unsupervised learning, on the other hand, occurs when the data is not pre-classified. In this case, the data mining process cannot make value judgments. It can find correlations within the data, but it is not able to make any inferences about the meaning of those patterns. In other words, unsupervised learning is descriptive while supervised learning is predictive (Christopher, 2009). As shown in Figure 2.4, knowledge discovery consists of the following steps (Han and Kamber, 2006):

1. Data cleaning, where noised and inconsistent data are treated;
2. Data integration, where multiple data sources may be combined;
3. Data selection, where data relevant to the analysis task are retrieved from the database;
4. Data transformation, where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for example;
5. Data mining, where intelligent methods are applied in order to extract data patterns;
6. Pattern evaluation, where the truly interesting patterns representing knowledge, based on some measures of interestingness, are identified;

7. Knowledge presentation, where visualization and knowledge representation techniques are used to present the mined knowledge to the user.

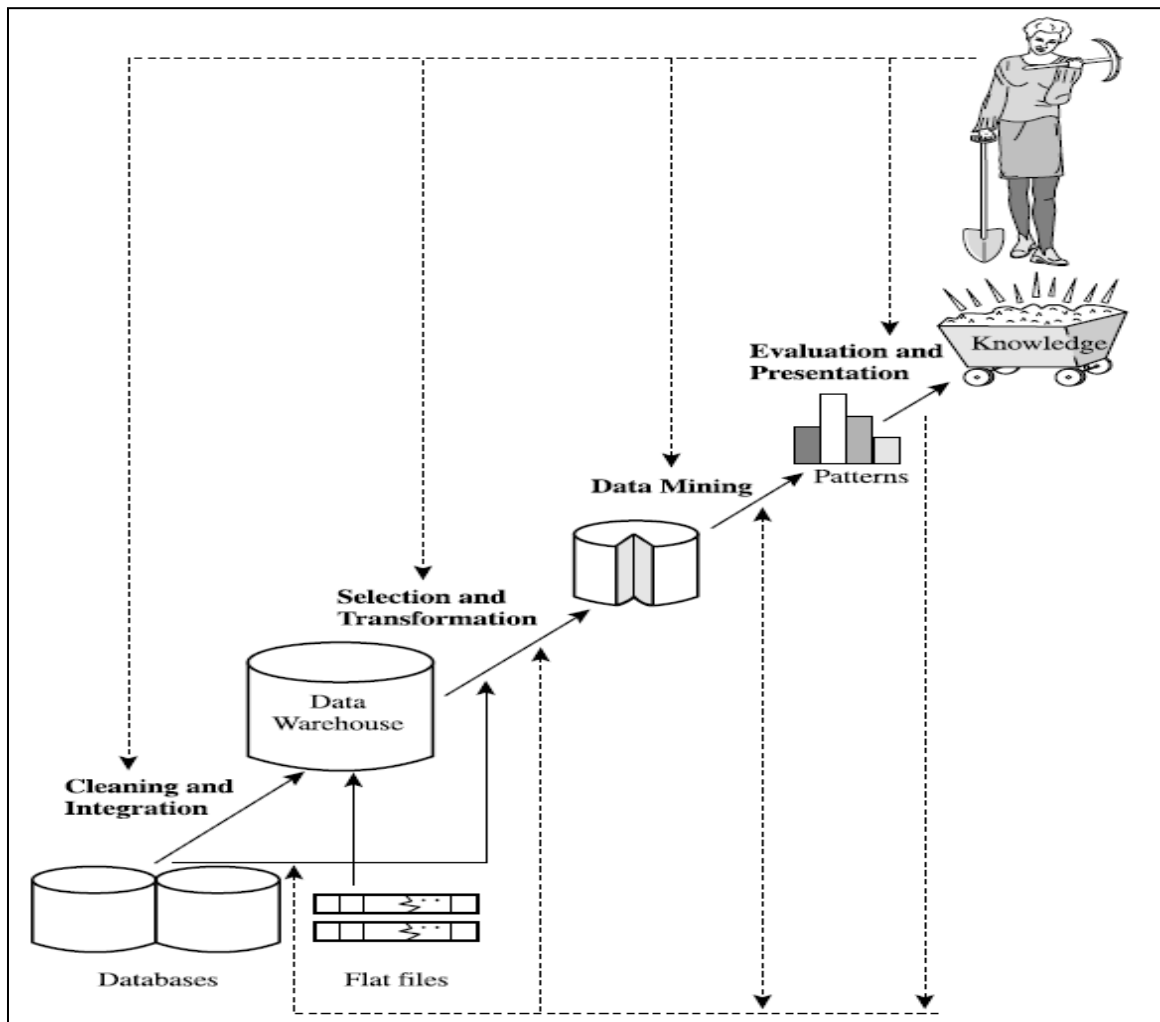


Figure 2-4 Data mining as a step in the process of knowledge discovery (Han and Kamber, 2006)

As shown in Figure 2.5, the architecture of a typical data mining system may have the following major components:

- A database system, data warehouse, World Wide Web, or other information repository which consists of databases, data warehouses, spreadsheets, or other kinds of information repositories.

- A database or data warehouse server which is responsible for fetching the relevant data, based on the user's data mining request.
- A knowledge base that is used to guide the search or evaluate the resulting patterns. The knowledge can include concept hierarchies, which are used to organize attributes or attribute values into different levels of abstraction.
- A data mining engine, which is essential to the data mining system and ideally consists of a set of functional modules for tasks such as characterization, association and correlation analysis, classification, prediction, cluster analysis, outlier analysis, and evolution analysis.
- A pattern evaluation module that employs interestingness measures and interacts with the data mining modules so as to focus the search towards interesting patterns. It may be used to filter out discovered patterns.
- A user interface module communicates between users and the data mining system. It allows the user to interact with the system by specifying a data mining query or task (Han and Kamber, 2006).

Knowledge Discovery in Database (KDD) is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data (Fayyad et al. 1996). Data mining is used to extract hidden knowledge from a data set. Hidden knowledge can be described as knowledge that is not readily obtained by traditional means such as queries or statistical analysis (Roiger and Geatz, 2003).

The KDD is an interdisciplinary field involving concepts from machine learning, database query, statistics, mathematics and visualization (Anand et al. 1998). It also consists of a series of steps including domain and data understanding, data preparation,

data mining, and finally, pattern evaluation and deployment (Chapman et al. 1999). KDD has been applied in the area of construction and facility management in recent research in response to the explosive growth of computerized historical databases (Buchheit et al. 2000; Soibelman and Kim, 2002).

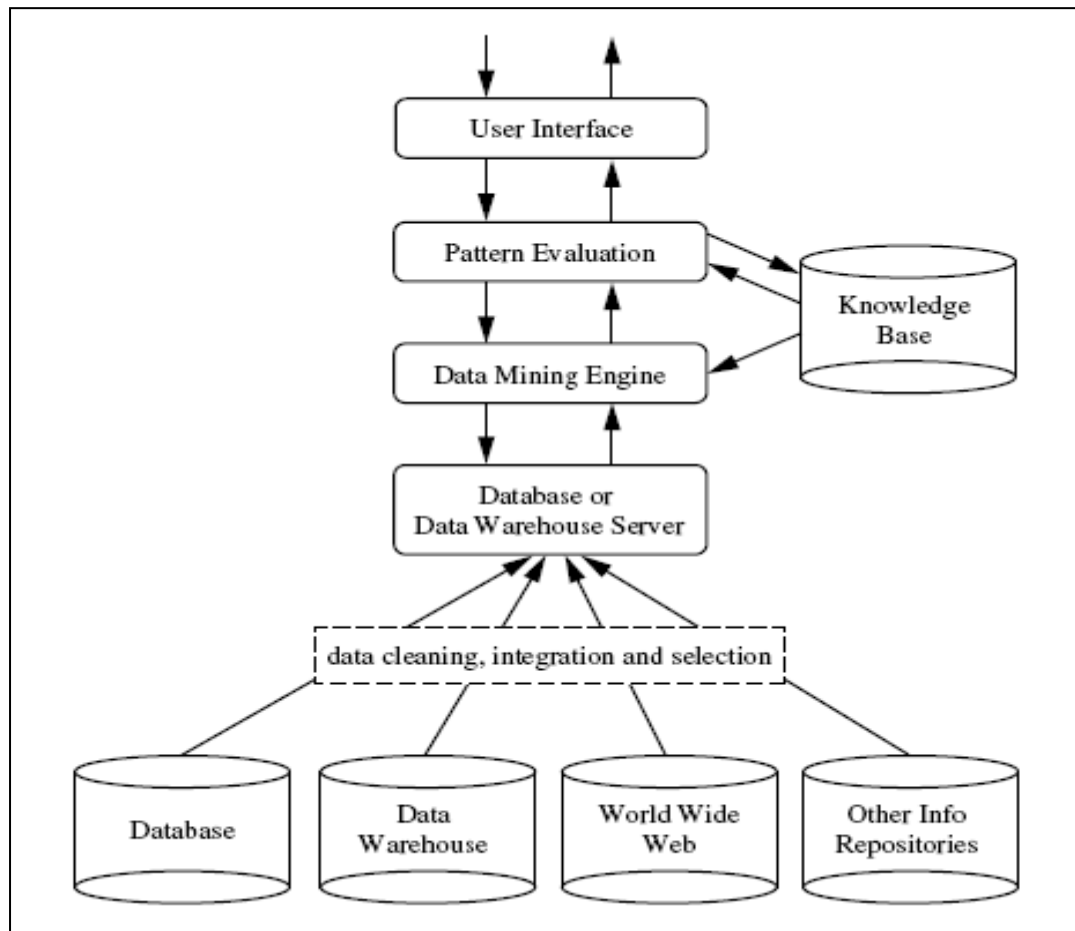


Figure 2-5 Architecture of a typical data mining system (Han and Kamber, 2006)

### 2.3.1 Data Mining & Knowledge Discovery Systems and Models in Construction

#### i. Approach of Knowledge Discovery in Databases

Soibelman and Hyunjoo, (2002) utilize knowledge discovery in databases (KDD) technologies that reveal predictable patterns in construction data that were previously

thought to be chaotic. This research presents the necessary steps for developing KDD such as 1- Identification of problems, 2- Data preparation, 3- Data mining, 4- Data analysis, and 5- Refinement process required for the implementation of KDD, as shown in Figure 2.6. The KDD process was applied to identify the causes of construction activity delays. Although this system presents the necessary steps for revealing the predictable patterns in construction data, the system does not provide any predictable models for that purpose, and so it is considered to be a methodology for KDD. Reffat et al. (2006) utilized the application of data mining techniques on building maintenance data. Applying data mining techniques and the potential benefits of their results are used to identify useful patterns of knowledge and its correlations. It supports decision making, improving the management of building life cycle. Boulicaut and Jeudy (2010) present an inductive query that specifies the desired constraints and algorithms that are used to compute the patterns satisfying the constraints in the data. This research emphasizes a real breakthrough for difficult problems concerning local pattern mining under various constraints. Even though this research presents the methodology of the data problems, it also presents a theoretical approach for dealing with data mining problems.

Soibelman et al. (2004) developed a methodology to provide timely and consistent access to historical data for efficient and effective management knowledge discovery. It is considered as a bridge between historical databases and data analysis techniques. It shields project managers from complex data preparation solutions. The methodology is called “data fusion” because it does not only involve data retrieval from construction databases, but it also reorganizes and represents historical data in a new analysis-friendly way that is different from their original data structures. It enables them to use discovered

knowledge for decision making more conveniently, as shown in Figure 2.7. This methodology does not present how these data analysis techniques work, or even the proposed analysis techniques themselves.

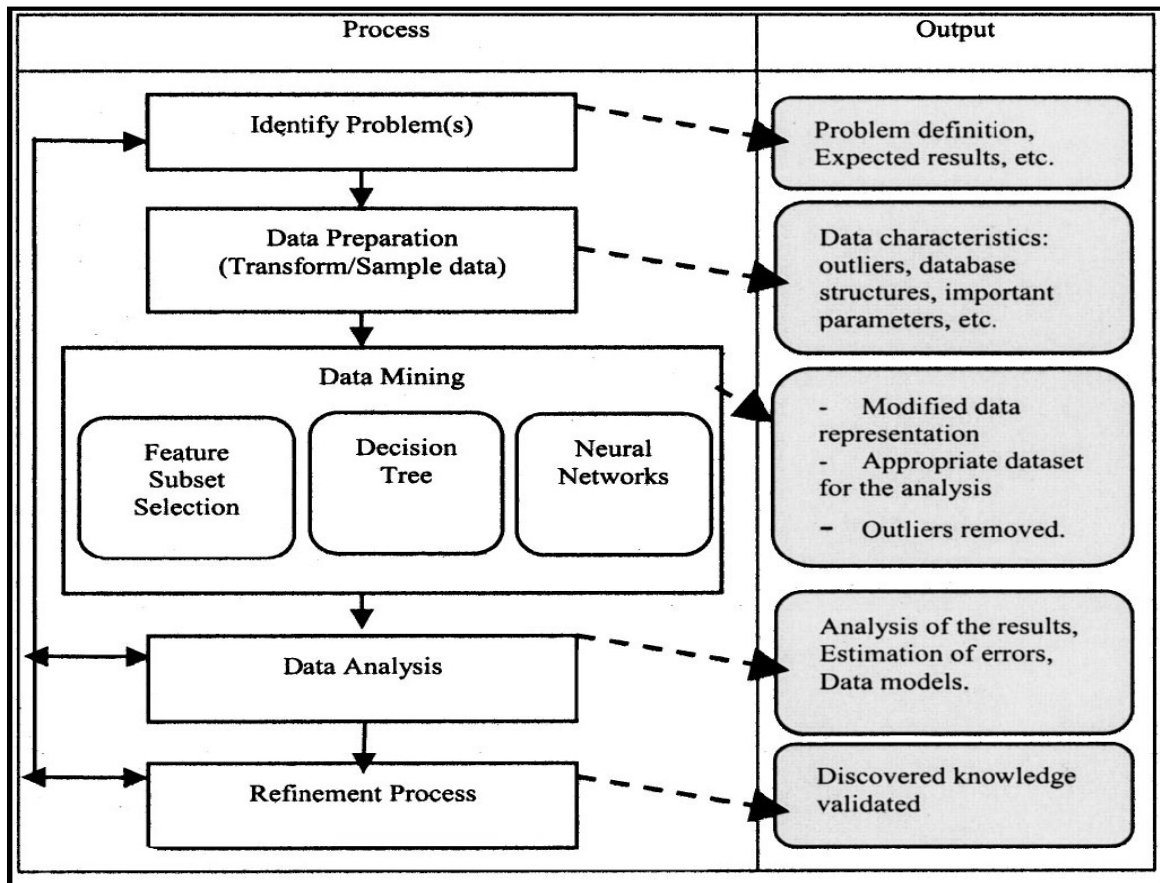


Figure 2-6 Knowledge discovery in databases approach (Soibelman and Hyunjoo, 2002)

ii. Application of data warehouse and Decision Support System in construction

Chau et al. (2003) developed this application to provide construction managers with information about and insight into existing data. It makes decisions more efficiently and without interrupting the daily work of an On-Line Transaction Processing (OLTP) system. The system solves this problem by integrating a data warehouse and a Decision Support System (DSS), changing the data in the data warehouse into a multidimensional

data cube and integrating the data warehouse with a DSS. The research is a prototype of the Construction Management Decision Support System (CMDSS), employing the integration of the ‘data warehouse’ technology with computer database platforms. It helps construction managers to view data from various perspectives with significantly reduced query time. Bao and Zhang (2010) analyzed the decision support system architecture based on data warehousing using SQL implementing, optimizing performance, and data mapping in OLAP. Although this research presented design principles and a method of data warehousing in decision support systems, it does not present a system or model to be applied. Shen et al. (2004) developed a methodology to provide a new way to handle the data-heterogeneity problems encountered in a construction project. The research proposes a tree structured product model. It binds design knowledge, cost data, and schedule data together as a feasible solution for the data integration problem in construction projects, as shown in Figure 2.8.

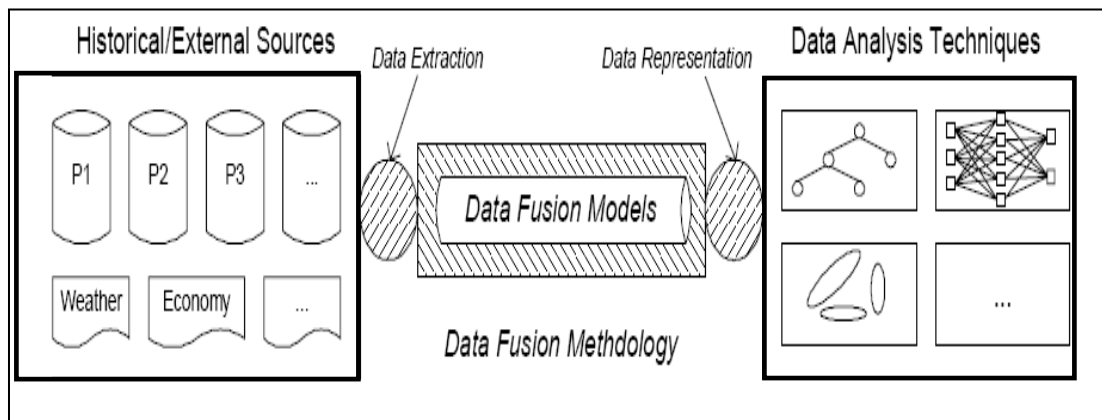


Figure 2-7 Data fusion and modeling methodology for construction management knowledge discovery (Soibelman et al. 2004)

The methodology uses the knowledge representation of construction projects based on ontology. Meta-data are used to describe the conceptual structure of the project

knowledge. The methodology supports queries about a particular construction project from different user perspectives, based on heterogeneous construction data sources. It works in a dynamic environment. The major parts of this methodology are: (1) organize the heterogeneous construction data into a tree structure; and (2) retrieve information and obtain domain views by specifying the ways of traversing the tree. The scope of this research is only to provide the fundamental methodology to handle heterogeneous and dynamic data in the construction industry.

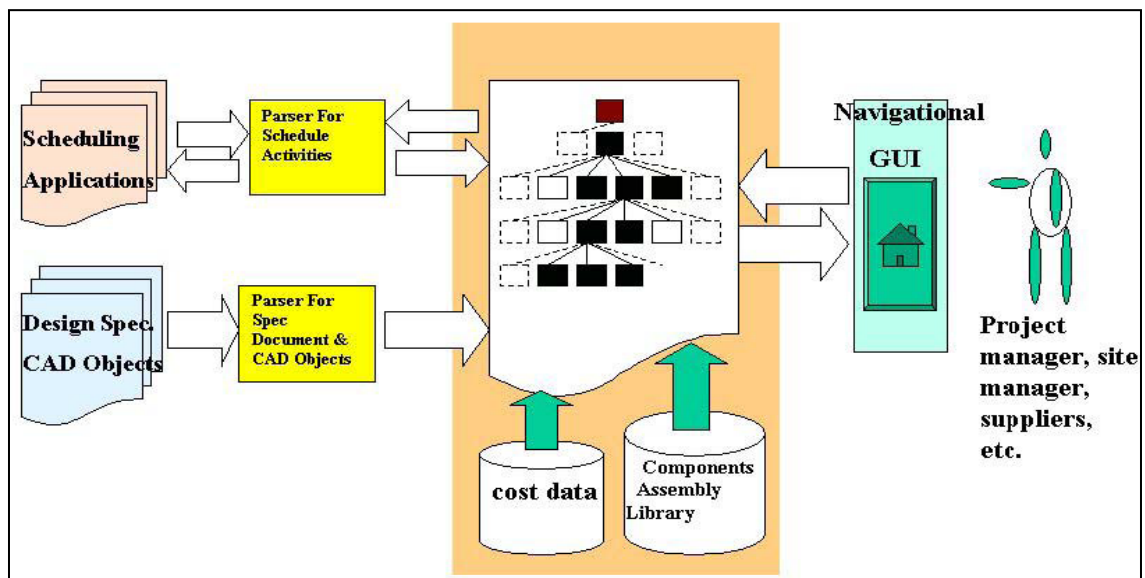


Figure 2-8 Design and schedule data integration system architecture (Shen et al. 2004)

Kivrak et al. (2008) carried out a survey among eight leading Turkish construction contractors operating within the international construction market. The objectives of this survey were to determine how the tacit and explicit knowledge are captured, stored, shared, and used in forthcoming projects, as well as major drivers and barriers for knowledge management. Based on the survey, it was determined that most of these firms do not have a knowledge management strategy and a systematic way of capturing and storing tacit knowledge. As shown in Figure 2.9, a conceptual framework is proposed to



formalize the knowledge-capturing process within construction companies. A web-based system and knowledge platform for contractors are presented. It is hypothesized that it can be used to manage both tacit and explicit knowledge effectively in construction projects.

## 2.4 Approaches to Model Building and Knowledge Discovery

Knowledge Discovery is not a single discipline but involves a combination of many techniques and technologies to support or filter services. Cox (2005) stated that the major methods of data mining and knowledge discovery are: (1) Ordinary Statistics, (2) Nonparametric Statistics, (3) Linear regression, (4) Cluster Analysis, (5) Artificial Neural Networks, and (6) Fuzzy SQL.

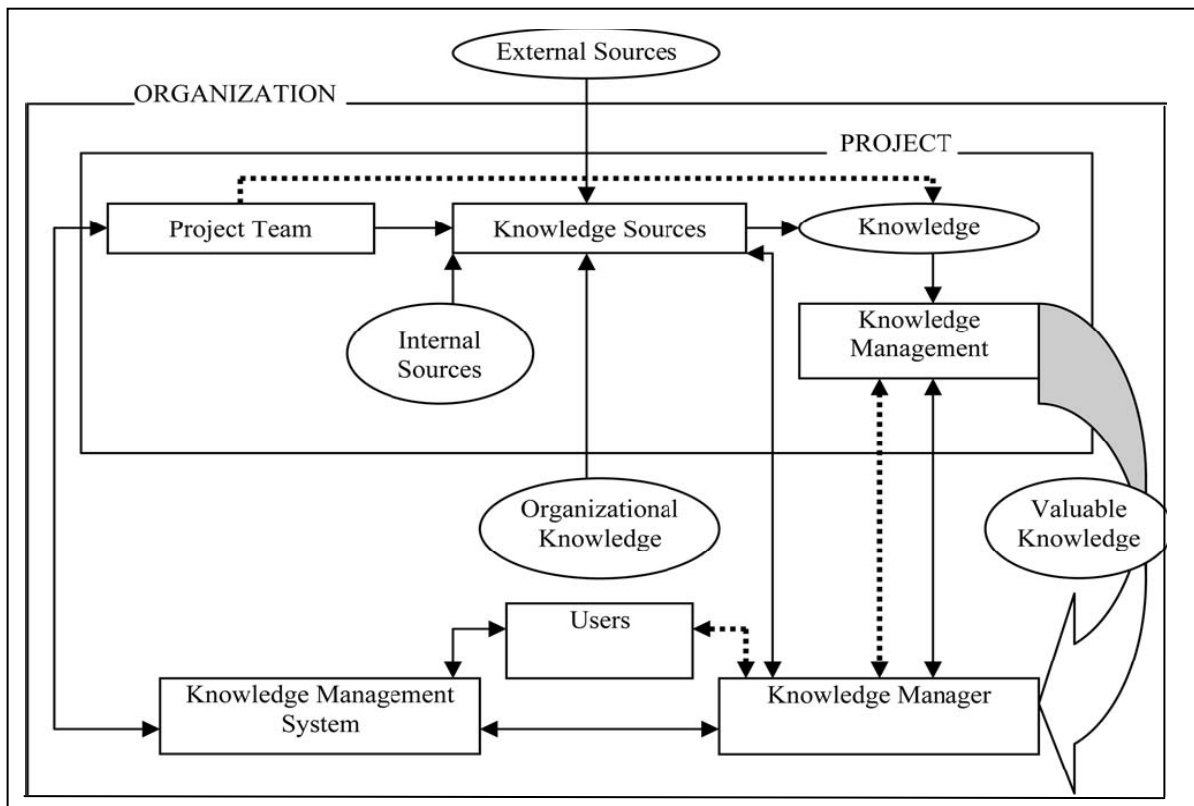


Figure 2-9 Conceptual frameworks for capturing knowledge in construction projects (Kivrak et al. 2008)

### **2.4.1 Ordinary Statistics**

This is generally the best starting point for any data mining project and often satisfies the need to understanding the mechanics of data relationships. For statistical data mining projects, a complete statistical analysis of existing data provides a keen understanding of how the data are clustered, the degree of variance, and the tightness of relationships. For any population of data, statistical properties need to be determined. These properties include the minimum, the maximum, the mean or average value, the mode, the median, and the standard deviation. Statistical analysis of large databases can often uncover many unexpected and interesting relationships. Ordinary statistics is a proven mathematical model. It can deal with large data, and find causal relationships among large numbers of variables, but the user must know what tests to apply and how to interpret the results. It is sensitive to population distribution assumptions and to the problem of outliers (Knorr and Raymond, 1996).

### **2.4.2 Nonparametric Statistics**

This term is defined as a distribution-free method which does not rely on assumptions--that data are drawn from a given probability distribution (Coder and Foreman, 2009). It means that the interpretation does not depend on the population fitting any parameterized distributions, for example, the ranks of observations (Wasserman, 2007). Nonparametric statistics is an advanced and proven mathematical method; it can describe an unknown population. It is useful in discovering the actual distribution in preparation for a more detailed analysis by ordinary parametric statistics, but it has a limited range of analytical tools. It is not often included in statistical software products (Coder and Foreman, 2009).

### 2.4.3 Linear Regression in Statistical Models

Regression analysis is a statistical methodology that utilizes the relation between two or more quantitative or qualitative variables so that the dependant variable can be predicted from the independent variables. In its simplest form the model can be stated as follows (Neter et al. 1996):

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \quad (\text{Equation 2.1})$$

Where  $Y_i$  is value of the response variable in the  $i^{\text{th}}$  trial,  $\beta_0$  &  $\beta_1$  are regression parameters,  $X_i$  is the value of the predictor variable in the  $i^{\text{th}}$  trial, and  $\epsilon_i$  is the random error. In multiple regression models, more than one variable is used to predict the behavior of the response variable. Therefore Equation 2.1 can be transformed into Equation 2.2.

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \dots + \beta_{p-1} X_{ip-1} + \epsilon_i \quad (\text{Equation 2.2})$$

The equation developed for the response variables is expected to give a best fit curve, which is anticipated to have certain variation errors based on the following assumptions:

- (1) The error around a regression line is independent for each value of predictor variable;
- (2) The errors around a regression line are assumed constant for all variable values; and
- (3) The errors around a regression line are assumed to be normally distributed at each value of  $x$  (Levine, 2008). Cox (2005) defined the strength and weakness of linear regression as follows: Linear regression is a proven mathematical technique. It is easy to use and understand. It creates a working mathematical model of the underlying system. It is easy to adjust through data fitting but it is limited to linear models. It is difficult to use for more than two variables. It is very sensitive to the degree of correlation between

variables, sensitive to outlier clustering and is a fairly simplistic method of curve fitting and often fails to find the true underlying trend line for the data.

#### **2.4.4 Cluster Analysis**

Clustering is an unsupervised learning technique that aims at decomposing a given set into subgroups or clusters based on similarity. The goal is to divide the data set based on the assumption that the sets belonging to the same cluster are as similar as possible, whereas sets belonging to different clusters are as dissimilar as possible. The motivation for finding and building classes in this way can be manifold (Bock, 1974 and Fung et. al. 2008).

Cluster analysis is primarily a tool for discovering hidden structures in a set of unordered data sets. In this case one assumes that a ‘true’ or natural grouping exists in the data. However, the assignment of data sets to the clusters and the description of these classes are unknown. By arranging similar data sets into clusters one tries to reconstruct the unknown structure in the hope that every cluster found represents an actual type or category of data sets. Clustering methods can also be used for data reduction purposes. In that case, it is aimed at finding a simplified representation of the set of objects. It allows for dealing with a manageable number of homogeneous groups instead of a vast number of single objects (Oliveira et al. 2007).

The cluster analysis technique is an unsupervised knowledge discovery technique forming the core engine in most data mining tools. It is available on the market for standalone cluster analysis. It is easy to implement using standard statistical toolkits. It works well with a wide spectrum of data types (numeric, categorical, and even textual data). It is a robust methodology that can be easily tuned for various types of analysis.

However, it is not a standalone data mining technique; it must be combined with other technologies. It is very sensitive to the choice of similarity functions, distance metrics, and variable weights. It also is very sensitive to the initial number of clusters and the initial (seed) values of the clusters. Cluster analysis has been used effectively in many studies, such as estimating haulers' travel time for estimating earthmoving production (Marzouk and Moselhi, 2004).

#### **2.4.5 Artificial Neural Networks (ANN)**

Artificial neural networks simulate the working network of the neurons in the human brain. Artificial neural networks are structured based on the functions of human brain learning mechanisms. The concept of the human brain is used to perform computations on computers (Ross, 2004). Artificial neural network (ANN) technology is a strong artificial intelligence technology that can deal with the dynamic nature of many real life systems' complexity. The ANN technique is an effective tool due to its ability to learn by example (Sawhney and Mund, 2002). The ANN technique is very useful in data modeling, when there is no clear relationship between different data elements, even when dealing with noisy data (Du, and Swamy, 2006). The ANN technique has been used effectively in many studies as a predictive tool.

Werbos developed Back propagation neural networks (BPNN) in 1974, which are considered the most used and most popular ANN algorithms. The BPNN mainly consist of three layers, usually called the input, the hidden, and the output layers, as shown in Figure 2.10. The input and output layers serve as nodes to buffer input and output for the model. The hidden layer provides a means for the input relations to be represented in the output (Werbos, 1994) and (Cho et. al 1999).

The BPNN is recommended when it is desired to correlate a large amount of data, after arranging this data into input/output data sets and when there is no clear relation among them. In BPNN the ANN needs to be trained; this training is based on pre-classified input and output data introduced to the network. The BPNN learns by examples; therefore, the user must provide a learning set that consists of some input examples with known output for certain cases (Sadiq et al. 2010).

The Artificial Neural Networks technique is based on the solid mathematics of learning in connectionist machines. It is well suited to modeling nonlinear and continuous data. It is easy to use (and many software tools are available). It can function in both supervised and unsupervised modes. It is a good data mining approach for large and unstructured databases, and it is a good choice when underlying reasons and explanations are not required. With AANs, the inputs must be well-understood. The outputs are continuous. The categorical data can be difficult to generate or to interpret. It is a “black box,” meaning it cannot explain its actions and a user cannot examine the “rules”. It requires expertise in neural networks to design and configure the system for the proper number of inputs, the correct number of hidden layers, and the suitable number of output classifiers. It requires some skill in deciding on a proper training algorithm for many neural network configurations. It can be computationally intensive in training mode, very difficult to make it handle nonnumeric data, and requires large amounts of training data. The network can be over-fit and over-trained. It produces a static model that must be retrained when new data are acquired because of fixed rules for the training data.

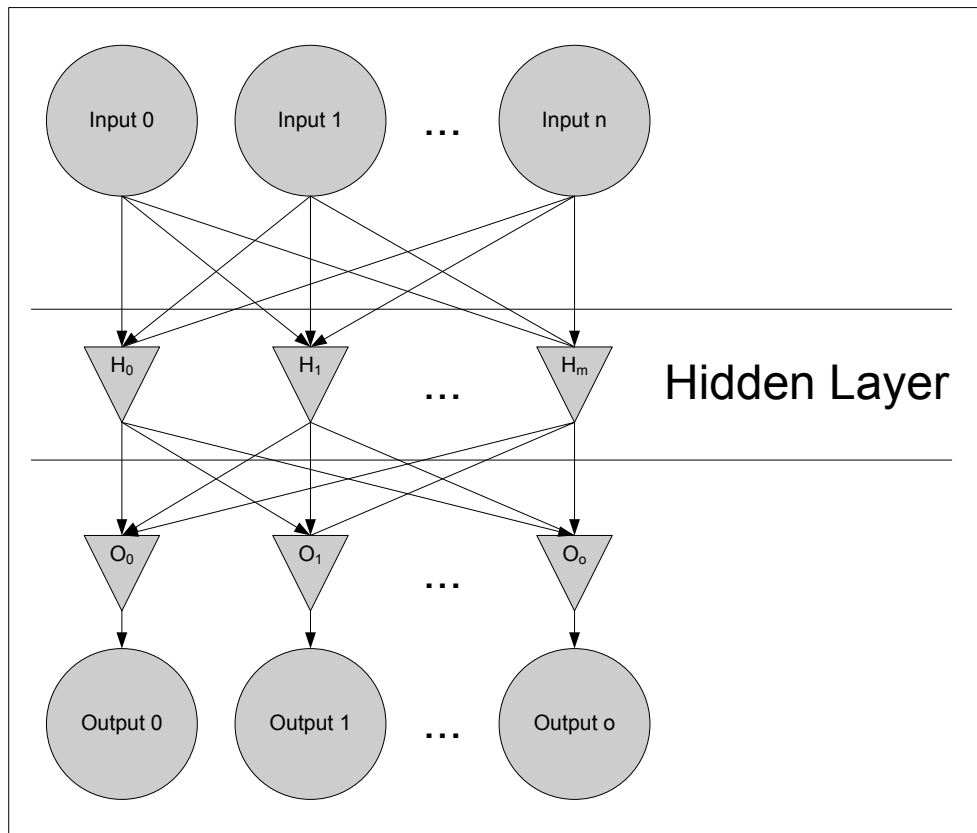


Figure 2-10 Basic architecture of ANN (Cho et al. 1999)

#### 2.4.6 Fuzzy SQL System

This kind of fuzzy query will be solved through combinations of fuzzy theory and SQL functions, which improves the fuzzy query system (Zhang et al. 2009). Database query systems provide the knowledge mining engineer with a formidable and powerful data analysis. It is a classification tool that will provide the ability to rank and classify sets of data based on the semantics of imprecision. It makes the fuzzy query system an ideal. In a fuzzy query, the database variables (fields) are broken down into collections of overlapping fuzzy sets (Cox, 2005). Although it provides a very convenient way for users to express complex queries, a nested fuzzy query may be very inefficient to process with the naive evaluation method based on its semantics (Yang et al. 2001). It is based on the

solid theory of fuzzy sets and approximate reasoning. It is easy to use and easy to understand. It works on any conventional database system as well as spreadsheets and comma-delimited files. It retrieves records based on “what I mean” not simply “what the arithmetic says”. Its results are ranked according to their compatibility. However, it is not easy to use on very large databases (due to a lack of fuzzy index support) or across multiple tables (due to a lack of fuzzy join). It requires an understanding of fuzzy logic and fuzzy set theory. It has only a few tools to support fuzzy queries. There are a few knowledge engineers trained in fuzzy logic (and those that do have some fuzzy logic experience generally have little, if any, experience with database systems and business problems. It cannot discover deep relationships. It must function with the proper definition of fuzzy sets and modifiers. It relies on the knowledge of the query user. It is applicable to numeric data only, and difficult to conceptualize complex query statements.

## **2.5 Fuzzy Knowledge Base Building and Design**

Many of the quantities that are considered to be crisp and deterministic are not really deterministic because they carry considerable uncertainty. The uncertainty occurs due to imprecision, ambiguity, or vagueness. The variable is probably fuzzy and can be represented by a membership function (Ross, 2004). Fuzzification is the process in which the crisp quantities are converted to fuzzy values. The fuzzy values are formed by identifying some of the uncertainties present in the crisp values. The conversion of fuzzy values is represented by the membership functions. Hence Fuzzification is performed. The Fuzzification process thus may involve assigning membership values for the given crisp quantities (Sivanandam et al. 2007).



### **2.5.1 Membership Value Assignments**

There are various methods to choose from to assign the membership values or the membership functions to fuzzy variables. The assignment can be just simply by intuition or by using some algorithms or logical procedures (Sivanandam et al. 2007). The methods for assigning the membership values are as follows (Sivanandam et al. 2007 and Ross, 2004):

#### **1. Intuition**

Membership functions are developed based on a person's own intelligence and understanding. The knowledge of the problem and linguistic variable must be known.

#### **2. Inference**

This method involves interacting with the knowledge to perform deductive reasoning. The membership function is formed from the known facts and the knowledge.

#### **3. Rank Ordering**

The polling concept is used to assign membership values by rank ordering process. The rank of pair wise comparisons will be used to rank of ordering the preference to form the membership function.

#### **4. Neural Networks**

Neural networks are used to determine the membership function from fuzzy classes of an input data set. The procedure starts with the selection of the number of input data values. Then it is divided into a training data set and a testing data set. The training data will be used to train the network. The neural network is created, from which the training is done

between corresponding memberships values in different classes. It simulates the relationship between the coordinate locations and the membership values. The neural network uses the set of data values and membership values to train itself. This training process is continued until the neural network can simulate for the given entire set of input and output values. After the net is trained, its validity can be checked against the testing data. After the training and testing process is completed, the neural network is ready and it can be used to determine the membership values of any input data in the different regions.

### ***5. Genetic Algorithms***

Genetic algorithms can be used to compute membership functions (Karr and Gentry, 1993) given some functional mapping for a system, some membership functions, and their shapes are assumed for the various fuzzy variables. These membership functions are then coded as bit strings which can then be concatenated. An evaluation (fitness) function is used to evaluate the fitness of each set of membership functions.

### ***6. Inductive Reasoning***

To develop membership functions from the data there is a procedure to establish fuzzy thresholds between classes of data. A threshold line can be determined with an entropy minimization screening method, and then the segmentation process can be started; first into two classes. By partitioning the first two classes one more time, three different classes will be generated. A repeated partitioning with threshold value calculations will divide the data set into a number of classes, or fuzzy sets, depending on the shape used to describe membership in each set. This method can be suitable for complex systems where

the data are abundant and static. When the data are dynamic, this method is not appropriate, since the membership functions continually change with time.

### ***2.5.2 Fuzzy Rule Induction and Fuzzy Models***

The goal of rule induction is to generate actual models of the underlying processes. As the term implies, rule induction creates a knowledge base of fuzzy if-then rules. It describes one or more behaviors in a large collection of data. Fuzzy rule induction is a supervised knowledge discovery approach that deals with a dependent or outcome variable and a wider set of independent variables. Rule induction discovers the functional relationships between the independent and dependent variables expressed as a set of fuzzy if-then rules. The concept behind a rule-based model is the mechanics of understanding what data means and how this meaning can be exposed and used. Such models have significant benefits over other representations, such as neural networks and decision trees; particularly because the rules are easy to understand and their reasoning can be explained. Induced rules can be mixed with expert opinion and rules can be regenerated based on the will-built model (Cox, 2005).

#### ***2.5.2.1 The Vocabulary of Rule Induction***

Berkan (1979) and Cox (2005) presented some vocabulary that will make understanding this topic much easier:

- i. Belief Function

The belief function parameter is a measure of the contribution or use of the underlying data associated with a rule. It provides rule degree scaling based on a subjective utility function associated with the data.

ii. Premise

In a fuzzy rule, the set of fuzzy propositions associated with the “IF” condition of the rule is known as the premise or the antecedent. For example, in the rule: If “a” is “Low” And “b” is “High” then “y” is “Small”, the premise consists of the two fuzzy propositions “a” is “Low” and “b” is High connected by the “And” operator.

iii. Degree of the Rule

The degree of a rule is the product of the fuzzy memberships for each of the variables. It measures the strength and applicability of the rule relative to all other rules that relate to the same outcome.

iv. Standard Error of Estimate

The difference between the predicted outcome and the actual outcome associated with a set of valid data points is the standard error of estimate in a fuzzy model categorization or prediction model. The standard error of estimate is a measure of model forecast precision.

v. Compression Tournament

The compression tournament is the process of finding the rules that best predict the outcome variable. The tournament is essentially a contest between similar rules to determine those that have the highest rule degree of a particular outcome.

vi. Fuzzy Associative Memory (FAM)

A fuzzy associative memory (FAM) is a multidimensional matrix representation of fuzzy rules. Each dimension of the FAM has a row for each fuzzy set in the term set associated with that variable.

### **2.5.3 Knowledge Base Generation Cycle**

The knowledge engineer specifies a dependent variable and a collection of independent variables. The induction process discovers the rules connecting the dependent to the independent variables from this collection as shown in Figure 2.11. The rules represent the relations in a functional relationship. Rule induction generates a FAM describing the underlying behavior patterns. This associative memory is a collection of evidence-weighted if-then rules mapping the behavior of the independent variables to the dependent variable. Figure 2.12 shows the basic knowledge-base generation cycle (Cox, 2005). Each field in the training file is defined through a collection of fuzzy sets. The data points are mapped to one or more corresponding fuzzy sets, generating a relationship between fuzzy spaces in the independent variables and a fuzzy space in the dependent variable. The relationship is developed in the form of rules. There are many candidate rules produced during rule generation, but rule induction eventually compresses these to an effective rule combination (Cox, 2005).

### **2.5.4 The Rule Induction Algorithm**

Fuzzy rule induction consists of three phases: the description of the model variables, the generation of candidate rules, and the selection of the final rule set. The goal of this algorithmic process is to produce fuzzy relations in the form of if-then rules (Berkan, 1979). In the description of the model variables phase, each variable is decomposed or partitioned into a set of fuzzy sets. These completely encompass the variable's domain or range of values. In the generation of candidate rules, a set of fuzzy relations is produced from the dependent and independent variables via three steps.

The first is determining the degrees of membership in each of the fuzzy sets associated with that variable's domain. The second is isolating the fuzzy set to data point mapping with the highest membership degree. In the final step, these fuzzy relations are converted to candidate if-then rules. In the selection of the final rule set phase, the degree of effectiveness (E) for each rule will be computed to select the most effective rules among the candidate rules. The effectiveness of each rule will be the product of its component of fuzzy set membership degrees. For example, the rule “**If** v1 is X **and** v2 is Y **then** v3 is Z” the effectiveness (E) will be Equation 2.3 (Berkan, 1979), (Cox, 2005) and (Wang et al. 1992).

$$E(r_i) = \mu_x(v_1) \times \mu_y(v_2) \times \mu_z(v_3) \quad (\text{Equation 2.3})$$

## 2.6 Multi objective Optimization

Optimization is the task of finding one or more solutions which match up to minimize or maximize one or more specified objective function(s) and satisfy all constraints. A single objective optimization problem involves a single objective function and usually results in a single solution, called an optimal solution.

An optimal solution is the best alternative among the feasible solutions. On the other hand, a multi-objective optimization considers several conflicting objectives simultaneously. In such a case, there is usually no single optimal solution, but a set of alternatives with different trade-offs that are called Pareto optimal solutions, or non-dominated solutions. Compared to single-objective optimization problems, in multi-objective optimization, there are at least two equally important tasks: an optimization task for finding the optimal Pareto solutions (involving a computer-based procedure) and a decision-making task for choosing a single most-preferred solution.

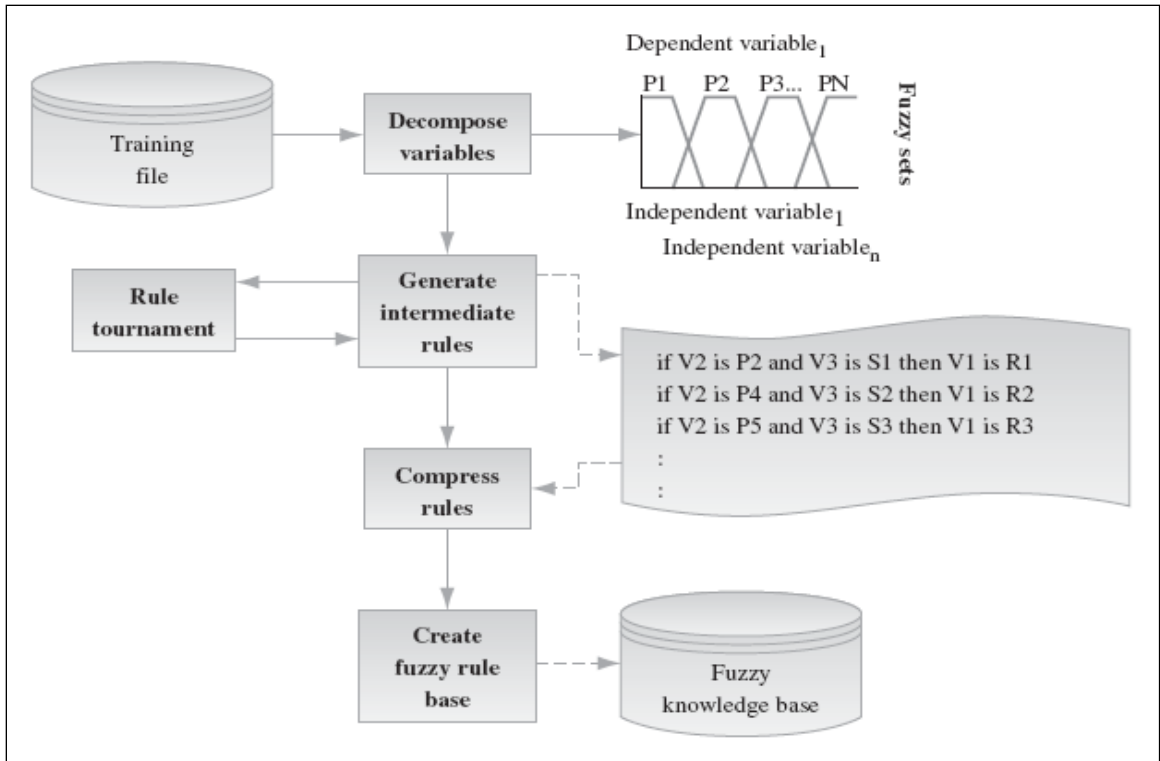


Figure 2-11 Rule induction process (Cox, 2005)

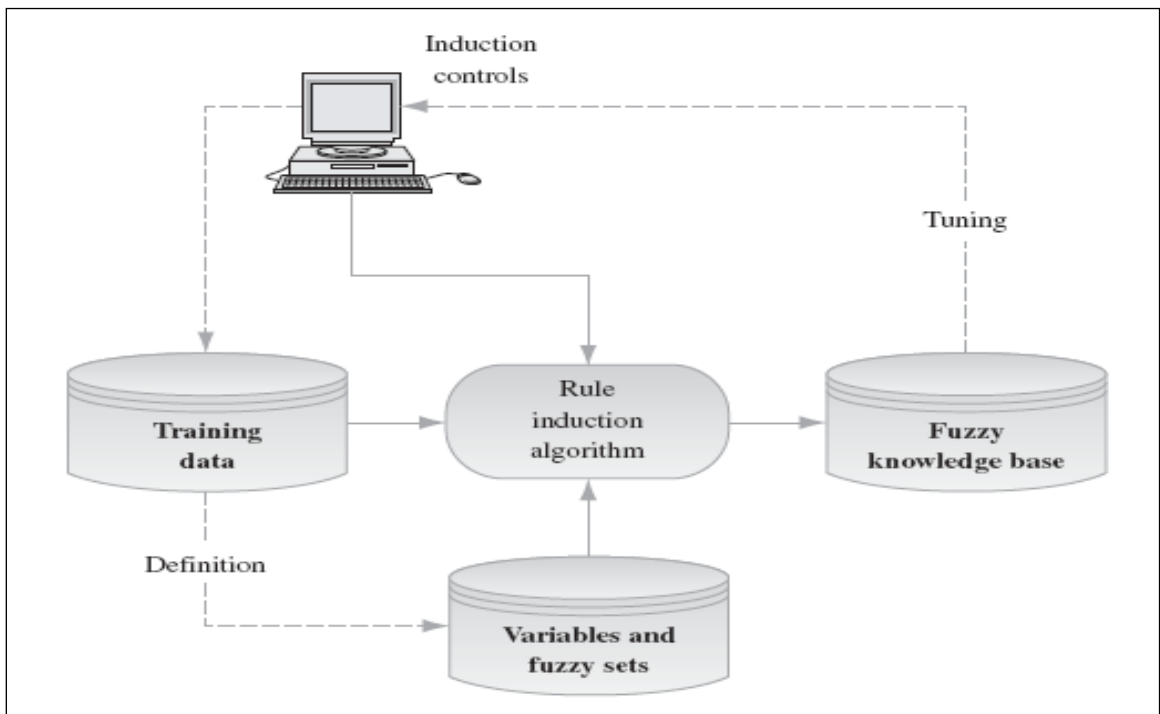


Figure 2-12 The basic knowledge-base generation cycle (Cox, 2005)

There is no unique solution in multi-objective optimization problems but a set of equally mathematical solutions can be identified. These solutions are known as non-dominated, efficient or Pareto optimal solutions (Branke et al. 2008). The main two groups of solution methods of multi-objective optimization are Classical and evolutionary algorithms. There are therefore three key concepts in evolutionary multi-objective optimization: i) Pareto Ranking, ii) Fitness Sharing, and iii) Elitism. These concepts are summarized below:

### **2.6.1 *Pareto Ranking***

Goldberg (1989) first proposed a Pareto-based fitness assignment based on Pareto dominance. This method utilizes a ranking system to assign equal probability of reproduction to the best individuals in the population. Rank one will be assigned to the first group of non-dominated solutions and they will then be temporarily removed from the population. The next group of non-dominated individuals is assigned rank two, temporarily removed, and so on as shown in Figure 2.13. The rank assigned to each individual represents its fitness level. This concept has been adopted with successful results with sum limitations of difficulties in high dimensionality of search space, for example a large number of objective functions (Fonseca and Fleming, 1993, 1995; Srinivas and Deb, 1994).

### **2.6.2 *Fitness Sharing***

The most important component of evolutionary multi-objective optimization is the definition of the diversity measure of a population. Goldberg and Richardson (1987) introduced the concept of sharing depending on which individuals share the same fitness when they are closer than a sharing distance.



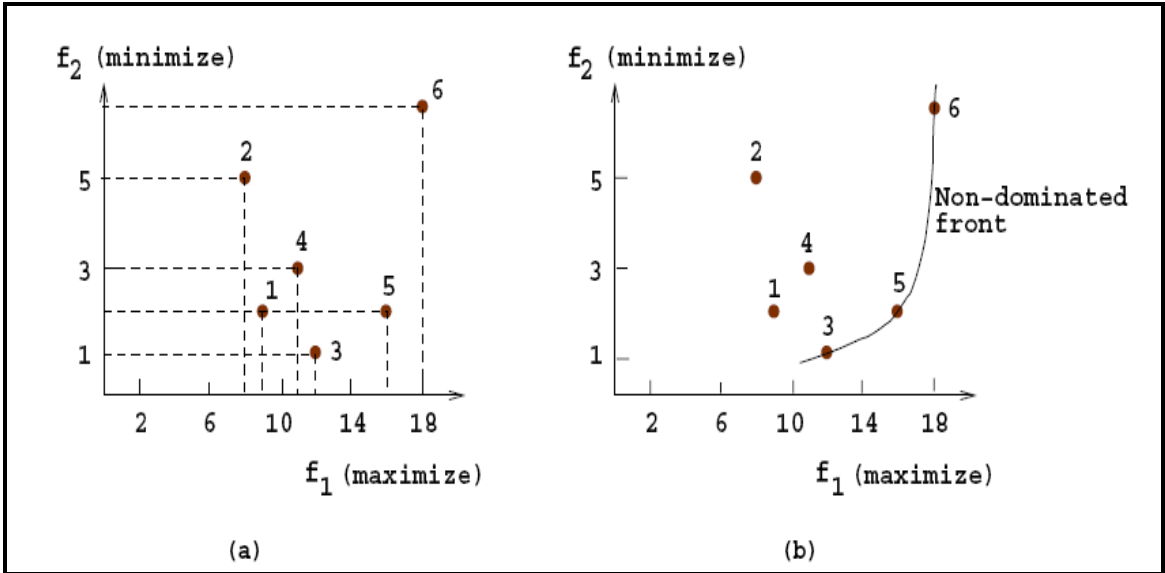


Figure 2-13 A set of points and the first non-dominated solution (Branke et al. 2008)

The fitness sharing technique has been successfully implemented in different applications, as described by Srinivas and Deb (1994), but the major criticisms of this technique is the need to specify a sharing parameter, which adds an additional complexity to the optimization problem. Many improvements have been made to the fitness sharing technique but it remains more advantageous to decrease the number of parameters that require certain tuning in the optimization problem (Cedeno et al., 1994).

### 2.6.3 Elitism

Various attempts have adopted a procedure to preserve better solutions unaltered through the generations to avoid the loss of good individuals during the evolution process. This procedure has been called elitism. Elitism has been applied in different evolutionary multi-objective optimization techniques. Various innovative evolutionary multi-objective algorithms have been proposed, including new procedures to account for the issues of fitness assignment, population diversity and elitism (Fonseca and Fleming, 1995). Elitism

is not considered to be a separate technique but rather a complementary technique to improve the optimization process by adding more complex procedures (Branke et al. 2008).

## **2.7 Summary and Limitations of the Literature**

This chapter reviewed many topics that present an overview on how to approach the stated problem. From the literature review, it is clear that most research in simulating construction processes has focused on modeling and simulation processes. This research has neglected the simulation input data problems, such as missing data and outliers. Even though many studies have discussed the effect of variables, such as weather, on productivity of construction processes, there is still a lack of studies on the effect of quantitative and qualitative variables on simulation process. In addition, most of the currently available simulation systems do not allow the user to optimize simulation results to find the optimum resource combination(s). In addition, results are not presented effectively enough to be understood by typical decision makers in the construction industry.

There is a massive amount of recorded data in the construction industry, but these data have not been used to develop a knowledge data bases. Most of the developed systems neglect the interaction between the user and the system during the modeling process. There is a need for a platform that integrates the simulation and programming environments. It is clear that the research works on construction simulation have several limitations, which are listed as follows:

- 1- A lack of simulation systems that model the uncertainty in the input data.

- 2- A lack of data problems treatment before simulation, such as data uncertainty, missing data and outliers.
- 3- A lack of using the available data sets to form knowledge data bases to improve the modeling process.
- 4- A lack of studying the effect of qualitative and quantitative variables on the simulation process.
- 5- A lack of simulation output analysis to find the optimum solution(s) for resource distribution.
- 6- A lack of interactive systems that interact with the user during simulation and modeling processes.

# **Chapter 3: Research Methodology**

## **3.1 Chapter Overview**

As shown in Figure 3.1, this chapter consists of seven sections that provide detailed explanation of the research methodology. It begins with a comprehensive literature review section that includes the state of the art review of construction simulation, data mining and knowledge discovery, and approaches to model building, Fuzzy knowledge base building and design as well as multi objective optimization. The second section pertains to system development and contains the following three stages: Knowledge Discovery, Simulation, and Optimization. The third section focuses on data collection, which includes a number of case studies so to verify and validate the credibility of the developed system. The fourth section denotes verification and validation of the developed system using a case study and comparison of the results with existing simulation systems. The fifth section presents the development of a general-purpose simulation language and computer based system that includes procedure for system automation under the development platform of C# Package. The sixth section is the verification and validation of the developed model/software package using a case study. The final section of this research presents conclusions and recommendations for future work. A detailed explanation of the aforementioned seven study sections and their sub sections follows.

## **3.2 Literature Review**

The literature review is performed in Chapter 2. It exhaustively covers the major fields that are crucial to this research topic. As illustrated in Figure 2.1, it consists of the following six sub-sections as follows:

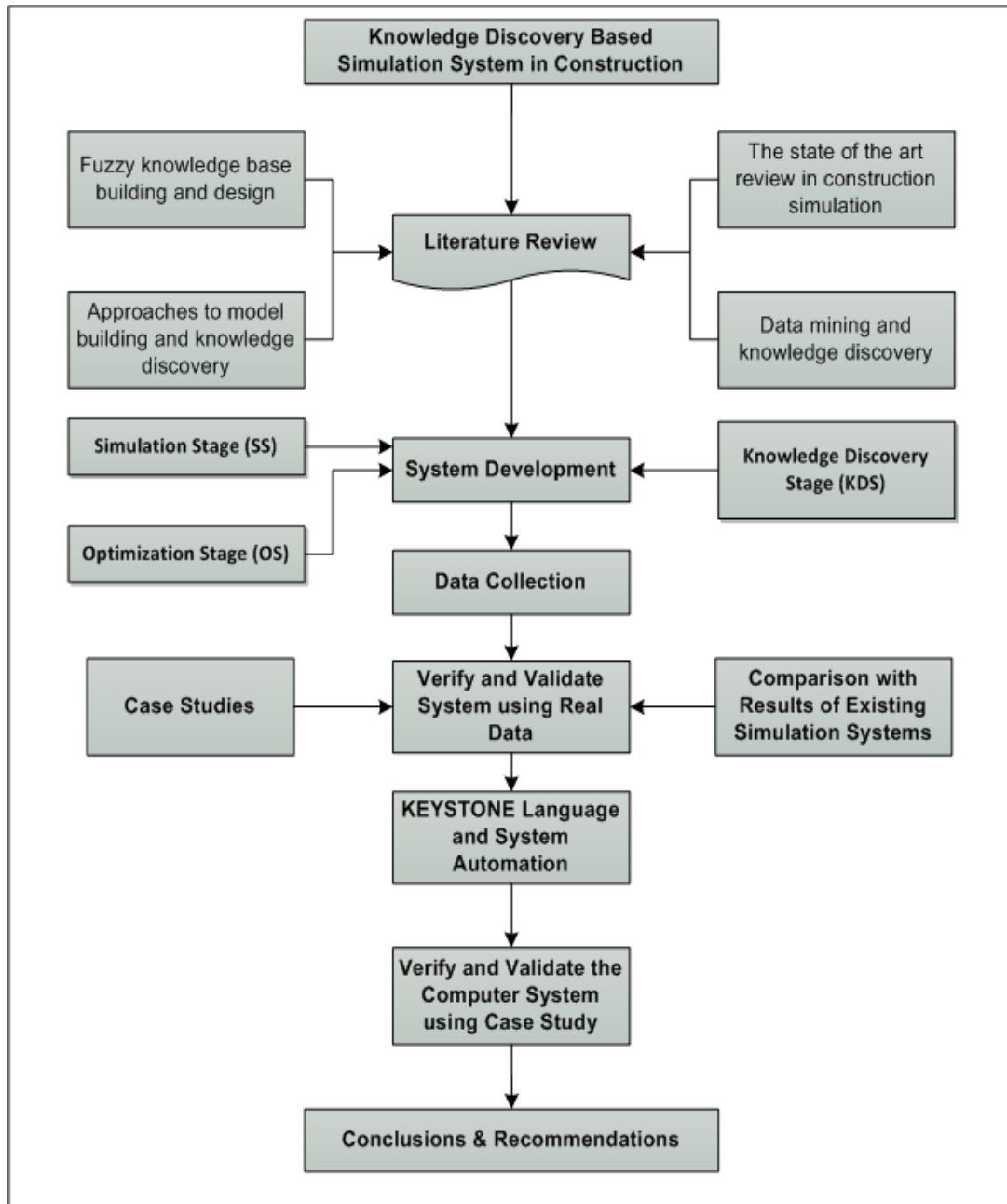


Figure 3-1 Overview of the research methodology

- 1- The state of the art review in construction simulation
- 2- Data mining and knowledge discovery
- 3- Approaches to model building and knowledge discovery
- 4- Fuzzy knowledge base building and design

5- Multi objective optimization

6- Summary and limitations of the literature review

Each sub-section is explained under their headings in accordance with the required details. The headings explain the theory, techniques, and applications of the systems that have been used in previous research studies. Also discussed in these sub-sections are how the techniques and applications could be used in this research. For further detail, the main literature review sub-sections are concisely presented in chapter two of this thesis.

### **3.3 System Development**

The system consists of three stages as shown in Figure 3.2. The stages are:

1- Knowledge Discovery Stage (KDS)

2- Simulation Stage (SS)

3- Optimization Stage (OS)

#### **3.3.1 Knowledge Discovery Stage (KDS)**

In this stage, raw data is prepared to be ready for the simulation stage. Moreover, patterns representing knowledge implicitly stored or captured in large databases are extracted and made available. The KDS includes data cleaning, integration, selection, transformation, and mining as well as pattern evaluation and knowledge presentation. Knowledge discovery in a database identifies valid, novel, potentially useful, and ultimately understandable patterns in data. The final phase is a data mining engine that is used to extract hidden knowledge from a data set. The result of this stage would be knowledge that is not readily obtained by traditional means such as queries or statistical analysis.

### **3.3.1.1 System Definition Phase**

This phase defines the system components, which consists of construction operations; processes, auxiliaries, and servers. Each component is represented by a model. The output induced by this phase leads to the definition of model's database. The project consists of at least one operation. Operations may be connected by a relation such as to a server or a process. System definition is the first phase to define project structure and its details.

### **3.3.1.2 Data Identification Phase**

In this phase, data corresponding to construction operations are identified. These data include quantitative and qualitative variables and work task durations for each operation. As shown in Figure 3.3, identification and collection of this data is through site observations, expert opinion and historical data. The most distinguishing characteristic between the construction industry and other industries is that each project has a unique nature. This means that even though the same resources and planning can be applied to other construction projects, identical job conditions do not exist. Based on this particular characteristic of the construction industry, on-site data collection from real construction projects is an indispensable requirement for research in KDS. Data are recorded and measured from jobsites and the Internet provided consistent observations, like Weather Networks, for analyzing the event times of construction processes.

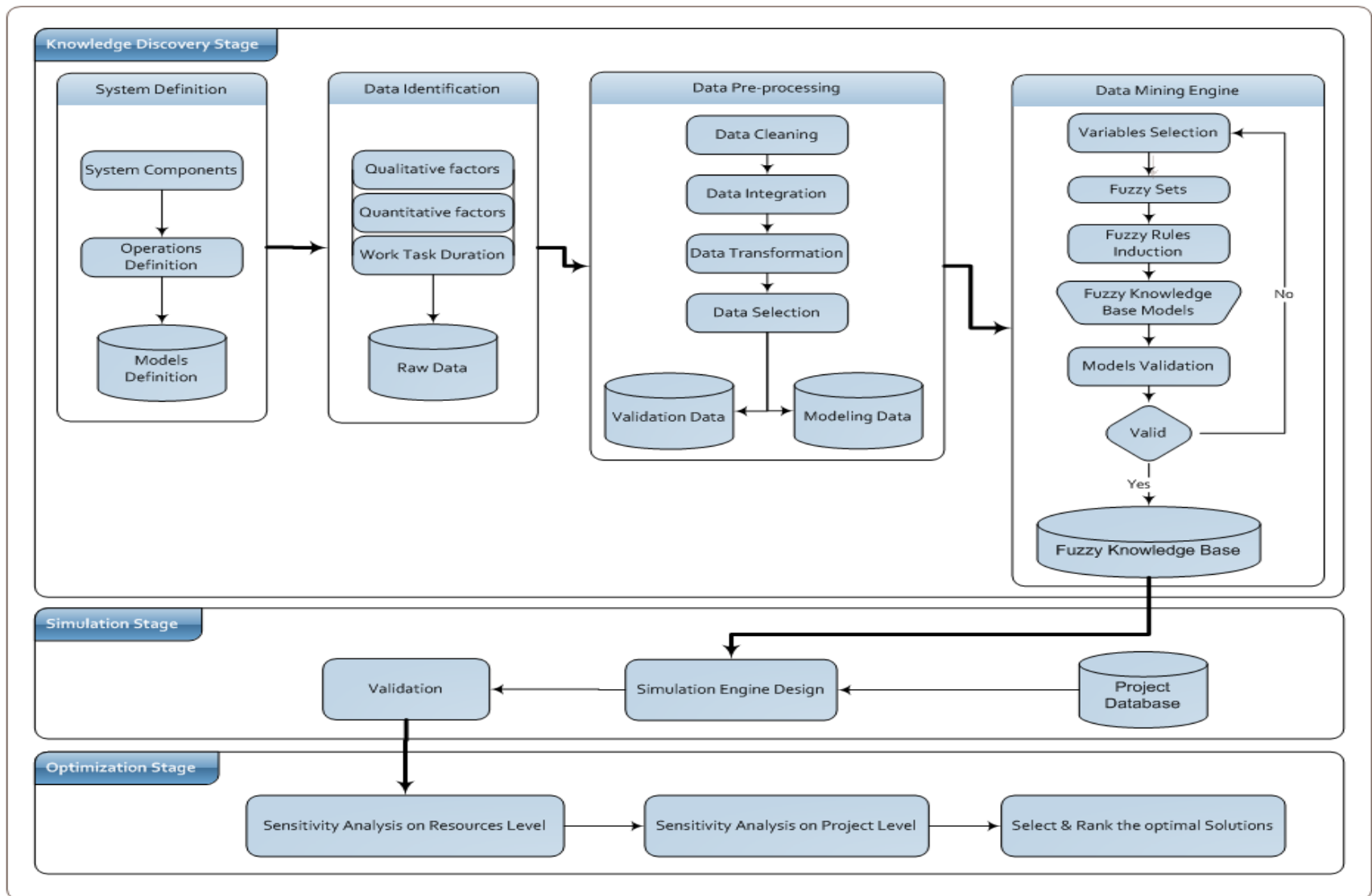


Figure 3-2 Overview of the system development



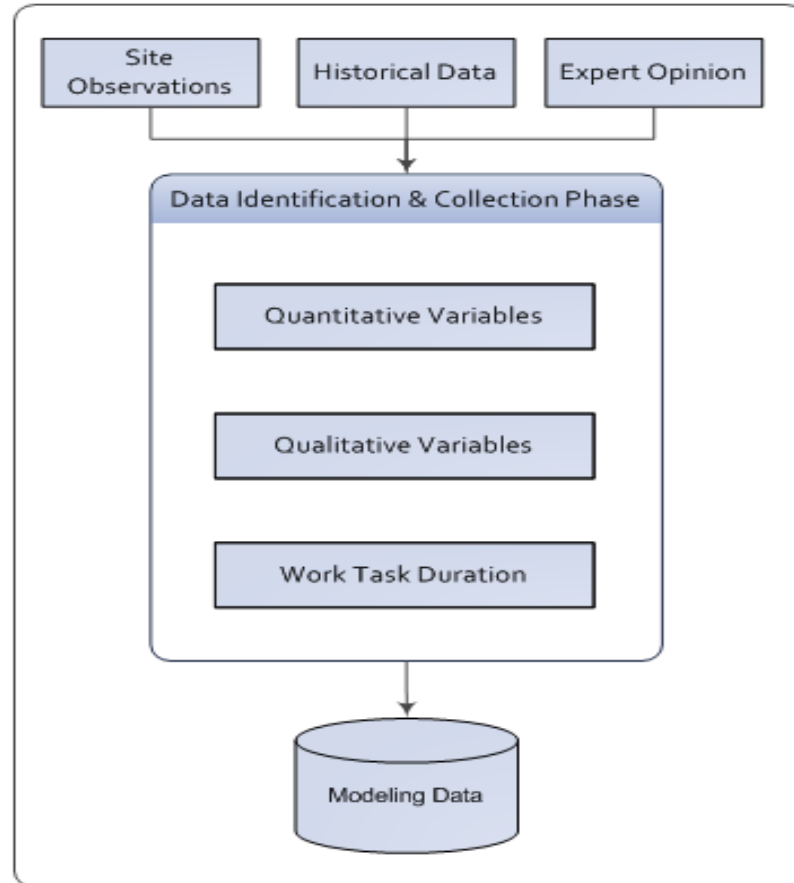


Figure 3-3 The architecture of the data collection phase

### 3.3.1.3 Data Preprocessing Phase

In this phase, as shown in Figure 3.4, raw data is processed to eliminate the problems (noise and inconsistency), improve quality, consolidate data from multiple sources and ease the data mining process. The data preprocessing stage includes data cleaning, integration, transformation, and selection as explained in the following sections:

#### I. Data Cleaning

Construction site data are often incomplete, noisy, and inconsistent. The data cleaning step attempts to fill in missing values, smooth out noise while identifying outliers, and rectify inconsistencies in the data. Data cleaning suppresses the drawbacks and shortcomings of missing data and outliers. The fundamental features of data cleaning

involve outlier detection, adaptation and estimation. In addition, the aim of data cleaning step is to cleanse the corrupted data from data sets, to deal with outliers and to replenish the data stream with an appropriate values as follows:

**a) Missing Data**

The Fuzzy K-means clustering technique is used to fill in missing data attributes. The fuzzy approach is used because fuzzy clustering provides a better description tool when clusters are not well-separated, as is the case in missing data imputation. In fuzzy clustering, each data object  $x_i$  has a membership function which describes the degree that this data object belongs to a certain cluster  $U_k$ . The membership function is calculated using Equation 3.1 (Deogun et al. 2004).

$$U = \frac{d(v_k, x_i)^{-2/(m-1)}}{\sum_{j=1}^k d(v_j, x_i)^{-2/(m-1)}} \quad (\text{Equation 3.1})$$

*Where:*

- m:* *m > 1 is the fuzzifier*
- X<sub>i</sub>:* *for any data object (1 ≤ i ≤ N)*
- k:* *cluster numbers*
- d(v<sub>k</sub>, x<sub>i</sub>):* *The distance between centroid v<sub>k</sub> and object x<sub>i</sub>*

The membership degree of each data object is considered in order to compute the clusters' centroids. Equation 3.2 is used to calculate the centroids (Deogun et al. 2004).

$$v_k = \frac{\sum_{i=1}^N U(v_k, x_i) * x_i}{\sum_{i=1}^N U(v_k, x_i)} \quad (\text{Equation 3.2})$$

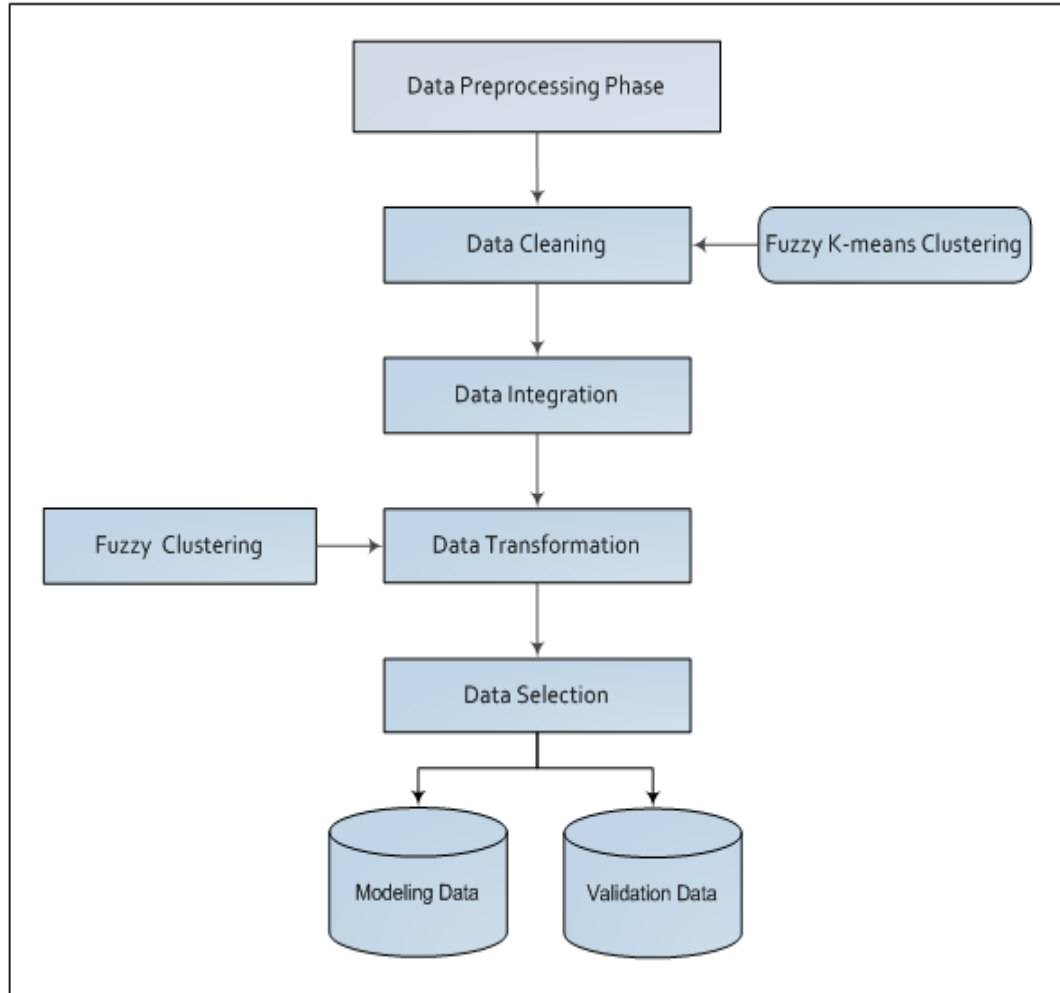


Figure 3-4 Architecture of the data preprocessing phase

The missing attributes are filled by replacing the attributes for each incomplete data object  $x_i$  based on the information about membership degrees and the values of cluster centroids as shown in Equation 3.3.

$$x_{i,j} = \sum_{k=1}^k U(x_i, v_k) * v_{k,j} \quad (\text{Equation 3.3})$$

## **b) Outliers**

Usually, there are some points that have suspicious attributes from a statistical perspective. It does not mean that these points should be eliminated. The removal of these data points could be dangerous as they may be presenting certain data patterns under special circumstances. Hence, the outliers will be retained in some clusters to be used in the simulation of forthcoming stages. The distance based outliers method is used because it does not require any prior knowledge of the underlying data distribution. This method uses the intuitive explanation for an outlier as an observation that is sufficiently far from most other observations in the dataset. A local metric-based outlier is observed when the samples are compared with neighborhood points.

## **II. Data Integration**

In this step, data from multiple data sources are merged by combining the data from multiple sources into coherent data storage. These sources may include multiple databases, data cubes, or flat files. There are some issues to be considered during data integration, such as schema integration, object matching and entity identification, which refers to the way that equivalent real-world entities from multiple data sources are matched up. Metadata are used to describe each attribute including the name, meaning, data type, and range of values permitted for the attribute, and null rules for handling blank, zero, or null values. Knowledge integration includes processing knowledge to prepare that acquired knowledge in consistent and usable forms. The success of data integration depends on how the possible undesirable events are identified and interrelated. The integration of acquired knowledge is an essential step to identify the conflicting pieces of information, gaps, and redundancies.

### **III. Data Transformation**

In data transformation, data is transformed or consolidated into forms deemed appropriate for mining, such as fuzzy clusters. This step converts information or data from one format to another, in order to allow the system to run on a specific platform. This method converts multiple data streams into a common format.

### **IV. Data Selection**

Data selection is a repeated step for data relevant to the analysis task which are retrieved from the databases. Data selection is performed to provide the data-mining engine with the required data. In addition, data selection is randomly selected from the validation and modeling data. The percentage of validation data is assigned by the user. This is an auxiliary step for the data-mining engine during any processing of data sets.

#### **3.3.1.4 Data Mining Engine Phase**

As depicted in Figure 3.5, in this phase, the processed data is prepared to be modeled and validated using the validation data set. This phase is the key component of simulation stage. The data-mining engine performs multiple functions such as modeling the processed data, preparing the fuzzy knowledge base, and simulating the predicted environment for future projects. This phase is essential to the data mining system and ideally consists of a set of functional steps for tasks such as variable selection, forming fuzzy sets, Fuzzy rules' inductions, and building a fuzzy knowledge base. The knowledge base will be finalized and recorded in the system after passing the validation process with an acceptable percentage. Data mining engine phase includes several steps as follows:

## I. Variables Selection

The variables that affect output variable (work task durations) are selected. The fuzzy average method is used, consisting of two steps: producing fuzzy curve and ranking process. The fuzzy average method consists of the following steps:

### a. Produce the fuzzy curve

A fuzzy curve is produced as follows:

- 1- Plot the M data points  $(X_i, Y_j)$ ,  $k = 1, 2, \dots, M$ , in each  $x_i$ -y space,  $i = 1, 2, N$  For each input variable  $x_i$ .
- 2- Create fuzzy membership function for each data point  $(X_i, Y_j)$  in each  $x_i$ -y space using Equation 3.4 to calculate the membership function for each input and output variables  $(X_i, Y_j)$  (Lin and Cunningham III 1995).

$$\mu_{i,k}(x_i) = \exp\left(\frac{x_{i,k} - x_i}{b}\right)^2 \quad (\text{Equation 3.4})$$

Where:

K: 1, 2, 3, ..., M

b :The width of the influence interval is 20% of variable range

- 3- Create a fuzzy curve  $c_i$  point for each data point by defuzzifying the fuzzy membership functions to produce a fuzzy curve, using Equation 3.5.

$$c_i(x_i) = \frac{\sum_{k=1}^M y_k * \mu_{i,k}(x_i)}{\sum_{k=1}^M \mu_{i,k}(x_i)} \quad (\text{Equation 3.5})$$

### **b. Ranking the variables**

The variables are ranked via Mean Square Error (MSE) using Equation 3.6. If this ranking is high, then the fuzzy curve is doing a poor job of representing the output with respect to the input. If there is a completely random relationship between the input and output, then the fuzzy curve is flat, and the MSE is very large. If the MSE is small, then the relationship can be judged to be more significant. The ranking for all input variables  $x_i$  are created as follows: compute the mean square error MSE for each fuzzy curve and rank the input variables in ascending order using MSE. The input with the smallest MSE is the most important, and the input with the largest MSE is the least important.

$$MSE_{ci} = \frac{1}{M} \sum_{k=1}^M (c_i(x_i, k) - y_k)^2 \quad (\text{Equation 3.6})$$

## **II. Fuzzy Sets (Fuzzification)**

This step is a process wherein the crisp quantities are converted into fuzzy sets. The fuzzy values are formed by identifying some of the uncertainties present in the crisp values of process durations and the related variables. The conversion to fuzzy values is represented by the membership functions. The Fuzzification process involves assigning membership values for the given crisp quantities. Membership values are assigned using neural networks technique because it doesn't need expert opinion or a fitness function like inference method and genetic algorithm. The Neural networks build the membership functions using the historical data that are used in the training process. Membership values are assigned using the neural network technique as follows:

**a. Assigning membership functions using Neural Networks**

A neural network is created from the modeling data sets of selected variables to determine the membership function, as shown in Figure 3.6. The training is done between the corresponding membership values in different classes in order to simulate the relationship between the coordinate locations and membership values. The neural network uses the sets of data values and membership values to train itself. This training process is continued until the neural network can simulate the entire given set of input and output values. After the net is trained, its validity can be verified by the testing data. The neural network is then ready and can be used to determine the membership value of any input data set in the different regions.

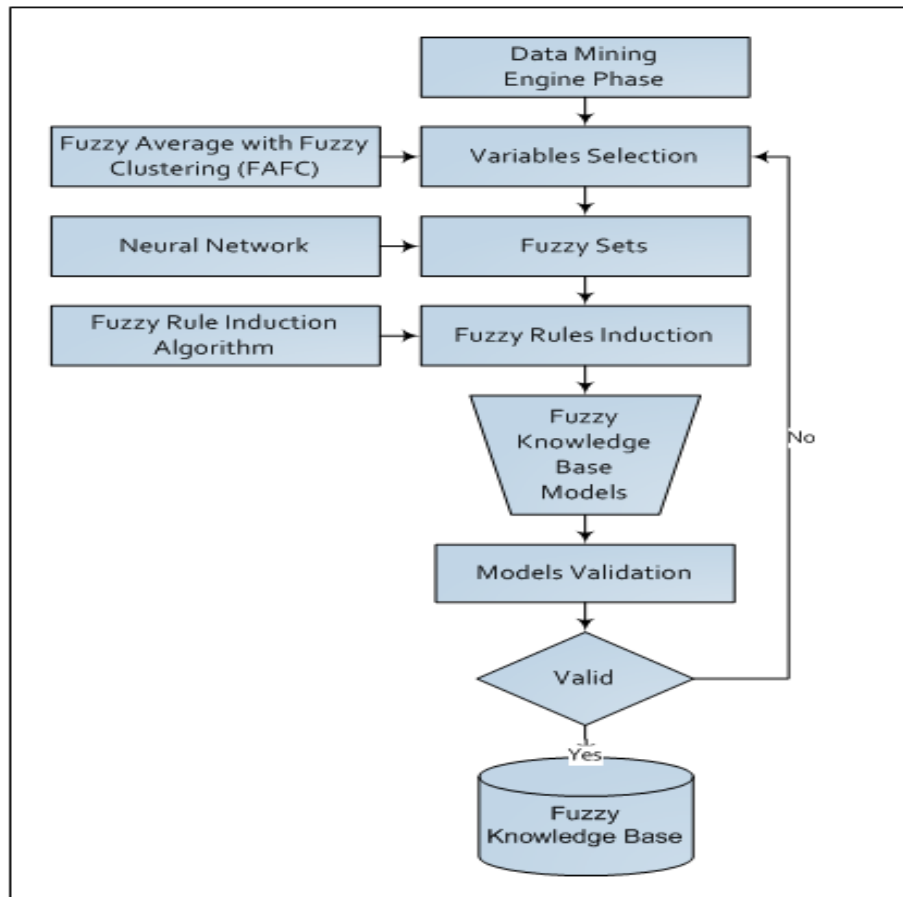


Figure 3-5 Architecture of the data mining engine phase



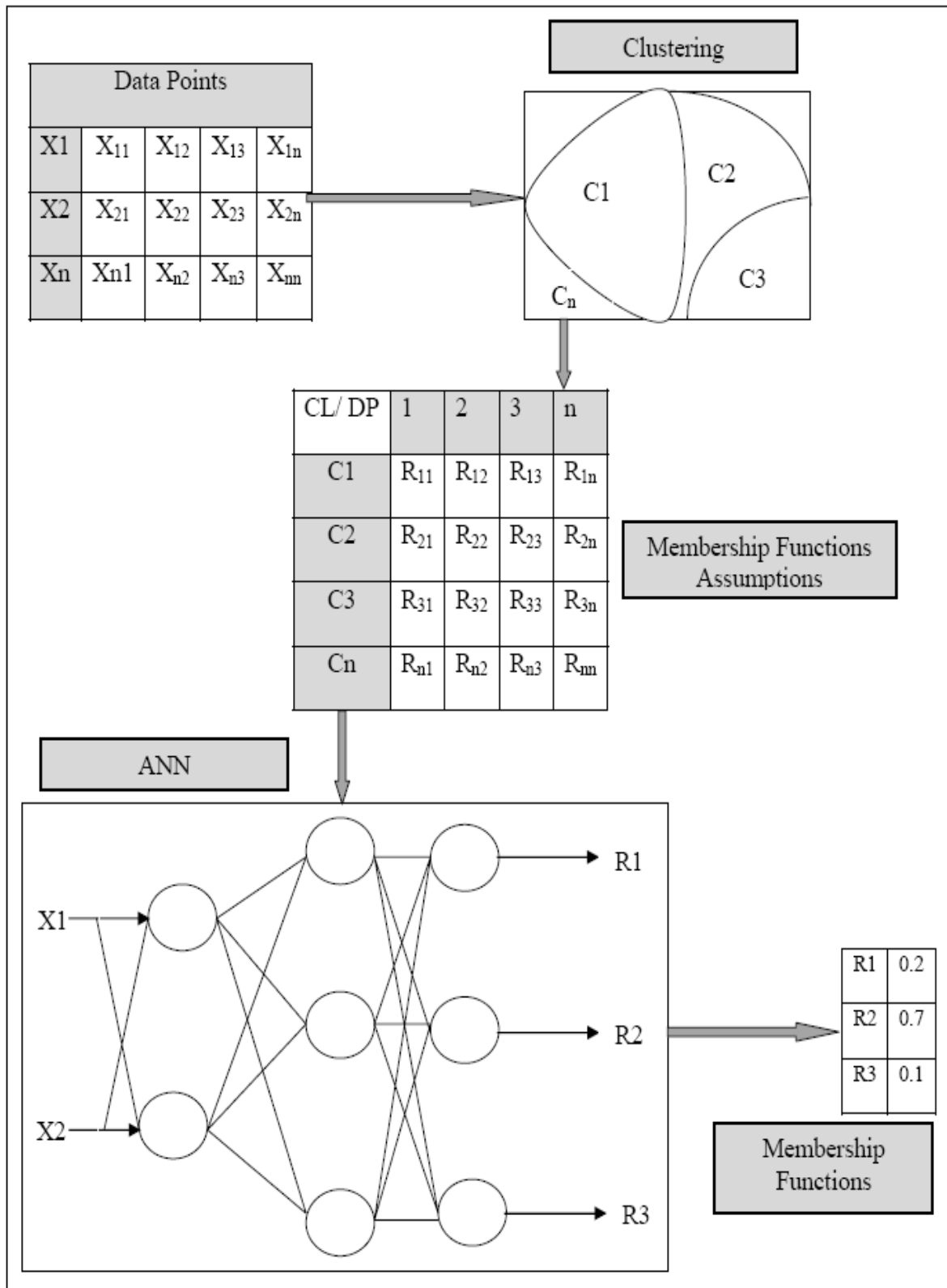


Figure 3-6 Neural network as a tool to determine membership functions

*i. Building and Training the Neural Network*

In this step, the modeling data, in the form of clusters, is used to build and train the proposed neural network. As shown in Figure 3.6,  $C_i$  represents cluster numbers,  $R_{ij}$  represents membership function values and  $X_{ij}$  represents the variables' data points. The data points  $X_{ij}$  are used as input values and the corresponding membership functions  $R_{ij}$  are used as output values. Random values are assigned to all the weights connecting the paths between the elements in the layers of the network  $W_{mn}^n$ . The outputs are calculated for each of the network layers using Equation 3.7 (Takagi and Hayashi, 1991). An input  $x$  from the training data set is then passed through the neural network. The neural network computes a value ( $f(x)$  output), which is compared with the actual value ( $f(x)$  actual =  $y$ ). The error measure  $E$  is computed from these two output values using Equation 3.8. The error measured in the previous step is associated with the last layer of the neural network. This error is distributed to the elements in the hidden layers via the back propagation technique using Equation 3.9 (Takagi and Hayashi, 1991). At this point, the errors associated with each element in the network are known. Therefore, the weights associated with these elements can be updated using Equation 3.10 (Takagi and Hayashi, 1991). Thus, the network approximates the output more closely. The input value  $X_{ij}$  is passed again through the neural network with the updated weights, and the errors, if any, are computed once more. This procedure is iterated until the error value of the final output is within certain limits or remains constant. Finally, a validation data set is used to verify how well the neural network can simulate the relationship.

$$O = \frac{1}{1 + \exp[-(\sum x_i w_i - t)]} \quad (\text{Equation 3.7})$$

*Where:*

O : output of the threshold element computed using the sigmoidal function

$x_i$  : inputs to the threshold element ( $i = 1, 2, \dots, n$ )

$w_i$  : weights attached to the inputs

t : threshold for the element

$$E = f(x)_{\text{actual}} - f(x)_{\text{output}} \quad (\text{Equation 3.8})$$

$$E_n = O_n(1 - O_n) \sum_j w_{nj} E_j \quad (\text{Equation 3.9})$$

$$w_{jk}^i(\text{new}) = w_{jk}^i(\text{old}) + \alpha E_k^{i+1} x_{jk} \quad (\text{Equation 3.10})$$

Where:  $w_{jk}$ : represents the weight associated with the path connecting the  $j^{\text{th}}$  element of the  $i^{\text{th}}$  layer to the  $k^{\text{th}}$  element of the  $(i + 1)^{\text{th}}$  layer

$\alpha$  : learning constant

$E_k^{i+1}$  : error associated with the  $k^{\text{th}}$  element of the  $(i + 1)^{\text{th}}$  layer; and

$x_{jk}$  : input from the  $j^{\text{th}}$  element in the  $i^{\text{th}}$  layer to the  $k^{\text{th}}$  element in the  $(i + 1)^{\text{th}}$  layer ( $O_{ij}$ ).

### III. Fuzzy Rules Induction

This step is to find the functional relationships between the independent variables' membership values and dependent variables (Task Duration). This step will also generate actual models of the underlying processes. Rule induction creates a knowledge base of fuzzy if-then rules. These rules describe one or more behavior in a large collection of data sets. This is a supervised knowledge discovery approach. It deals with a dependent or outcome variable and a wider set of independent variables. The functional relationships between the independent and dependent variables are expressed as a set of fuzzy if-then

rules. The concept of a rule-based model is the mechanics of understanding what data means and how this meaning can be exposed and used.

Rule induction presents some rules that are both easy to understand and to comprehend their reasoning. Fuzzy rule induction consists of the following two processes:

**a. Generation of Candidate Fuzzy Rules**

The inputs of this process are fuzzy sets and the model variables from the fuzzy sets step and system definitions step are discussed earlier. The fuzzy relations are produced from the dependent and independent variables. The functional relationships between the independent and dependent variables are expressed as a set of fuzzy if-then rules through two steps:

*i. Isolate the fuzzy sets to data points*

The inputs of this step are fuzzy sets that are isolated to data point mapping with the highest membership degree that could occur if there are two data sets in the same fuzzy set. The higher degree is selected. Even if there are misleading points in the data sets, these are avoided in the subsequent steps for example selection of the final rules.

*ii. Convert all fuzzy relations to rules*

As the final step, these fuzzy relations are converted into candidate if-then rules through the selected rules of each group of rules. In this step, the functional relations between dependent and independent variables are transformed into rules without any relation with the data set. Each rule will be separated in a final rule selection process.

### **b. Selection of the Final Rule Set**

As the final rules are selected from a vast pool of data pairs, and in respect to any conflicted rules, the degree of effectiveness (E) for each rule is computed. The most effective rules amongst the candidate rules are selected. The effectiveness of each rule is the product of its component of fuzzy set membership degree, as shown in Equation 3.11 (Cox, 2005).

$$E(ri) = \mu_x(v1) \times \mu_y(v2) \times \mu_z(v3) \quad (\text{Equation 3.11})$$

## **IV. Fuzzy Knowledge Base (FKB)**

The fuzzy knowledge base is the collection of all previous steps. It is a collection of variables, fuzzy sets, and fuzzy rules. The FKB is a representation of a particular model. Although a FKB is, generally, not a complete representation of a system, a collection of FKBs may cooperate to form a solution to a complex system. The FKBs work like containers; they store the data definitions (variables) and rules required to solve a set of outcomes associated with the system. The stored knowledge can include concept hierarchies to organize attributes or attribute values into different levels of abstraction. The FKB will not be finalized until validated.

## **V. Validation**

The goal of this step is to test the prediction effectiveness of the developed system. The first test is plotting the actual versus predicted values. If the plot shows that the predicted values are within acceptable limits, then, they are scattered around the actual values of the response variable. The second test is a mathematical and descriptive validation. Equation 3.12 represents the average invalidity percent (AIP), which shows the prediction error. If the AIP value is closer to 0.0, the model is sound and a value closer to

1 shows that the model is not appropriate (Zayed and Halpin, 2005). Similarly, the root mean square error (RMSE) is estimated using Equation 3.14. If the value of the RMSE is close to 0, the model is sound and vice versa. In addition, the mean absolute error (MAE) is defined as shown in Equation 3.15. The MAE value varies from 0 to infinity. However, the value of the mean absolute error should be close to zero for sound results (Dikmen et al. 2005).

$$AIP = \left( \sum_{i=1}^n \left| 1 - \left( \frac{E_i}{C_i} \right) \right| \right) * 100/n \quad (\text{Equation 3.12})$$

$$AVP = 100 - AIP \quad (\text{Equation 3.13})$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (C_i - E_i)^2}{n}} \quad (\text{Equation 3.14})$$

$$MAE = \frac{\sum_{i=1}^n |C_i - E_i|}{n} \quad (\text{Equation 3.15})$$

Where:

*AIP* : Average Invalidity Percent

*AVP* : Average Validity Percent

*RMSE* : Root Mean Squared Error

*MAE* : Mean Absolute Error

*E<sub>i</sub>* : Estimated value

*C<sub>i</sub>* : Actual value

### 3.3.2 Simulation Stage (SS)

In this stage, the movement of units in a real world system is modeled. The objective of developing this stage is to examine the interaction between flow units and idle times of the resources to flag bottlenecks and estimate productivity of the proposed system. This

stage consists of two phases; (1) simulation engine design and (2) validation of the developed system. The phases of simulation stage are explained in the following sections.

### **3.3.2.1 Simulation Engine Design Phase**

The simulation engine is designed as shown in Figure 3.7. The framework of the engine is implemented within the “during construction” and “planning” stages. The inputs of this engine are the results of data mining engine, fuzzy knowledge base and project database. If the project is in the planning stage, the input variables, that affect the duration, are identified, studied, and transformed into fuzzy membership functions through the data mining engine. Then, the fuzzy knowledge base is used to predict task durations. During the construction stage, durations are collected from real operations. The discrete event simulation system starts when the work task commences when simulation time equals to zero in the first run. If the work task cannot achieve this condition, the simulation clock is advanced. After this startup, the procedure will be as follows: (1) units will be moved to their work task, (2) the generated durations will be used, and (3) the end event time will be calculated by adding the durations to TNOW and the end of event time (EET) will be recorded in the event list. The chronological list is produced based on the events list.

### **3.3.2.2 Validation Phase**

The goal of this phase is to examine the system modeling effectiveness in order to validate the developed simulation tool/model(s). During construction stage, simulation results are compared to actual results for productivity and waiting times in queues where the validation Equation (3.15) can be used. If the percentage of difference is acceptable, the model is performing well, otherwise the model is rejected. The acceptance of the

validation percent is related to the model builder or decision maker depending on some factors as importance of the process. In the planning stage, simulation results are validated using historical data from similar projects and expert opinion on the results of developed simulation system.

### **3.3.3 Optimization Stage (OS)**

The first goal of this stage is to evaluate the effect of changing input variables on existing operation and on simulation system output(s). The second goal is to rank and select the optimum resource combinations. This stage consists of three phases. In the first phase, only sensitivity analysis at the resource level is conducted. In the second phase, sensitivity analysis at the project level is carried out to assess the effect of each variable on the optimum resource combination(s) as well as other feasible resource solutions. In the third phase, the selection and ranking of optimal solutions is done using rank assignment algorithm.

#### **3.3.3.1 Sensitivity Analysis on the Resource Level Phase**

In this phase, the feasible solutions are generated by changing one resource while fixing all others. This phase indicates the effect of each resource on the feasible solutions and the optimum solution(s). It will help decision makers to see the project in different situations and help them to understand the impact of any resource allocations on project's productivity. There is no difference between this phase and the current common practice in the other available systems.



### **3.3.3.2 Sensitivity Analysis on the Project Level Phase**

In this phase, as shown in Table 3.1, the system is run under different variable conditions, taking into consideration changing one variable while fixing all other variables and resources. This phase indicates the effect of each variable on the feasible solutions and the optimum solution(s). This phase shows the validity of the optimum solution on all of the system's states. This phase combine the effect of changing resource allocation and variables.

This combination provides a wide overview of the project to a decision maker. The decision maker can see the project under the best and worst cases of weather conditions, for example, and also same for resource allocation situation. This phase presents many combinations for the same productivity and different cost per time units.

### **3.3.3.3 Selection and Ranking of the Optimal Solutions Phase**

In this phase, as shown in Figure 3.8, the optimal solution(s) is selected. The ideal case is to find one solution (dominant solution), but in most cases a set of solutions are presented (non dominant – Pareto optimal). A Pareto-based fitness assignment is selected based on the Pareto dominance to utilize a ranking system to assign equal probability of reproducing the best individuals in the population. Rank one is assigned to the first group of non-dominated solutions which are then temporarily removed from the population. The next group of non-dominated individuals is then assigned rank two, and so on. The rank assigned to each individual represents its fitness level.

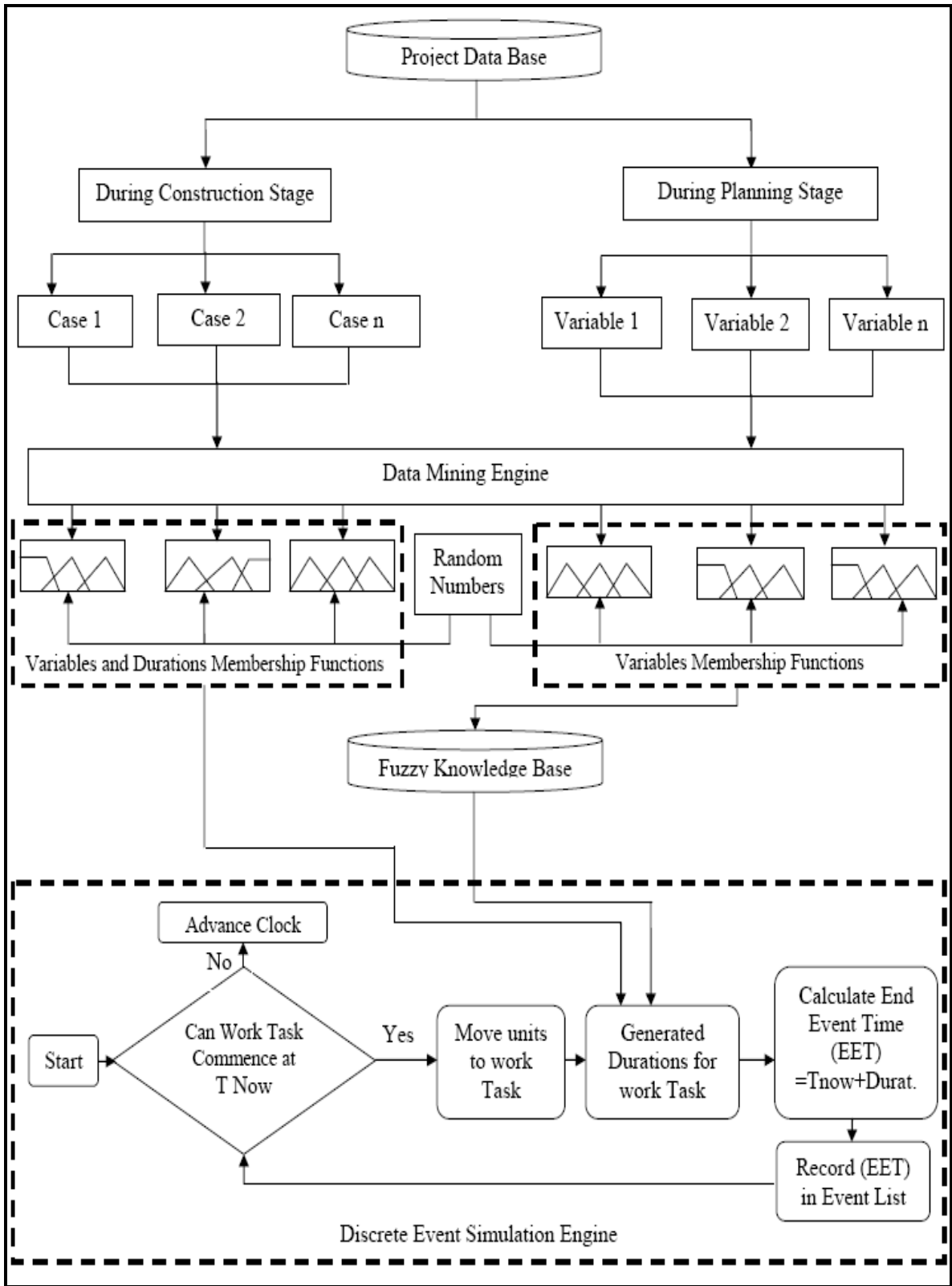


Figure 3-7 Framework of the simulation engine

Table 3-1 Sensitivity analysis matrix

Variables Information			Resources Information			Productivity Information		
V1	V2	V3	R1	R2	R3	Productivity Per Unit Time	Cost Per Unit Time	Cost Per Production Unit
V1 <sub>1</sub>	V2 <sub>1</sub>	V3 <sub>1</sub>	R1 <sub>1</sub>	R2 <sub>1</sub>	R3 <sub>1</sub>	P01	T01	U01
V1 <sub>2</sub>	V2 <sub>1</sub>	V3 <sub>1</sub>	R1 <sub>1</sub>	R2 <sub>1</sub>	R3 <sub>1</sub>	P02	T02	U02
V1 <sub>3</sub>	V2 <sub>1</sub>	V3 <sub>1</sub>	R1 <sub>1</sub>	R2 <sub>1</sub>	R3 <sub>1</sub>	P03	T03	U03
V1 <sub>1</sub>	V2 <sub>1</sub>	V3 <sub>1</sub>	R1 <sub>1</sub>	R2 <sub>1</sub>	R3 <sub>1</sub>	P04	T04	U04
V1 <sub>1</sub>	V2 <sub>2</sub>	V3 <sub>1</sub>	R1 <sub>1</sub>	R2 <sub>1</sub>	R3 <sub>1</sub>	P05	T05	U05
V1 <sub>1</sub>	V2 <sub>3</sub>	V3 <sub>1</sub>	R1 <sub>1</sub>	R2 <sub>1</sub>	R3 <sub>1</sub>	P06	T06	U06
V1 <sub>1</sub>	V2 <sub>1</sub>	V3 <sub>1</sub>	R1 <sub>1</sub>	R2 <sub>1</sub>	R3 <sub>1</sub>	P07	T07	U07
V1 <sub>1</sub>	V2 <sub>1</sub>	V3 <sub>2</sub>	R1 <sub>1</sub>	R2 <sub>1</sub>	R3 <sub>1</sub>	P08	T08	U08
V1 <sub>1</sub>	V2 <sub>1</sub>	V3 <sub>3</sub>	R1 <sub>1</sub>	R2 <sub>1</sub>	R3 <sub>1</sub>	P09	T09	U09
V1 <sub>1</sub>	V2 <sub>1</sub>	V3 <sub>1</sub>	R1 <sub>1</sub>	R2 <sub>1</sub>	R3 <sub>1</sub>	P10	T10	U10
V1 <sub>1</sub>	V2 <sub>1</sub>	V3 <sub>1</sub>	R1 <sub>2</sub>	R2 <sub>1</sub>	R3 <sub>1</sub>	P11	T11	U11
V1 <sub>1</sub>	V2 <sub>1</sub>	V3 <sub>1</sub>	R1 <sub>3</sub>	R2 <sub>1</sub>	R3 <sub>1</sub>	P12	T12	U12
V1 <sub>1</sub>	V2 <sub>1</sub>	V3 <sub>1</sub>	R1 <sub>1</sub>	R2 <sub>1</sub>	R3 <sub>1</sub>	P13	T13	U13
V1 <sub>1</sub>	V2 <sub>1</sub>	V3 <sub>1</sub>	R1 <sub>1</sub>	R2 <sub>2</sub>	R3 <sub>1</sub>	P14	T14	U14
V1 <sub>1</sub>	V2 <sub>1</sub>	V3 <sub>1</sub>	R1 <sub>1</sub>	R2 <sub>3</sub>	R3 <sub>1</sub>	P15	T15	U15
V1 <sub>1</sub>	V2 <sub>1</sub>	V3 <sub>1</sub>	R1 <sub>1</sub>	R2 <sub>1</sub>	R3 <sub>1</sub>	P16	T16	U16
V1 <sub>1</sub>	V2 <sub>1</sub>	V3 <sub>1</sub>	R1 <sub>1</sub>	R2 <sub>1</sub>	R3 <sub>2</sub>	P17	T17	U17
V1 <sub>1</sub>	V2 <sub>1</sub>	V3 <sub>1</sub>	R1 <sub>1</sub>	R2 <sub>1</sub>	R3 <sub>3</sub>	P18	T18	U18

The ranking results will give decision maker a group of feasible solutions ranked in a descending way according to the unit cost and productivity. For example, if P01 equal to P02 and U01 more than U02, the ranking will select the point of less cost where the (P02, U02) solution is selected. In addition, the ranking will consider the difference in both productivity and unit cost. The selection and ranking, as shown in Figure 3.8, will be done according to the following steps:

- 1- Definition of the objective functions. Maximize productivity objective function (f1) for productivity data sets (Pn) and minimize cost objective function (f2) for unit cost data sets (Un) from all the feasible solutions that are generated from the sensitivity analysis.

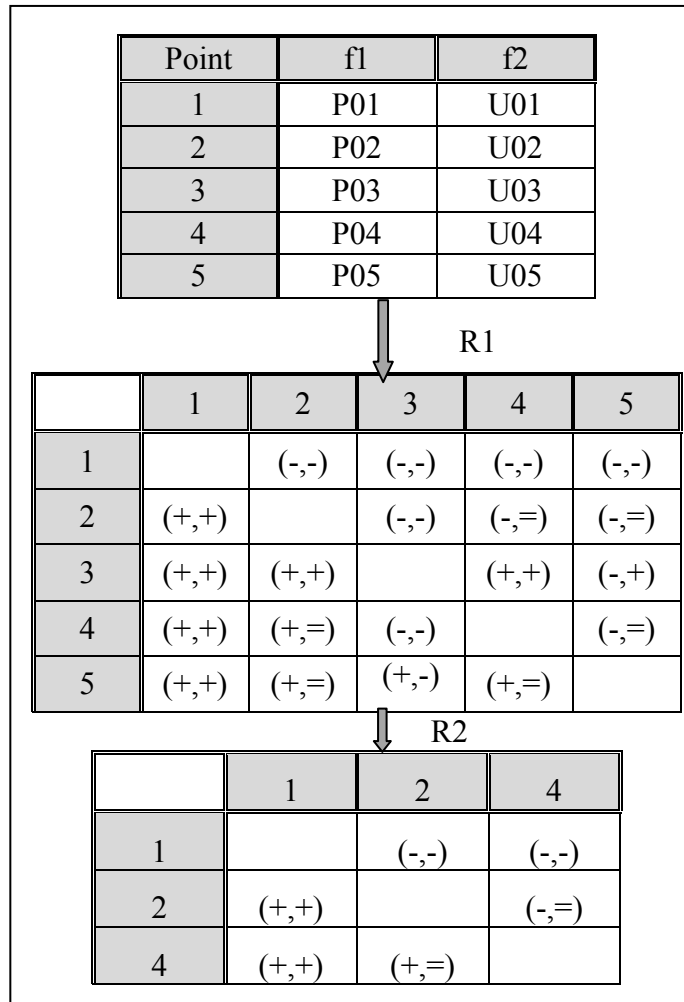


Figure 3-8 The selection and ranking procedure

- 2- The comparison between two solutions is presented by a pair of symbols. Each of these symbols can take three values, +, -, or =, according to whether it is better than, worse than or equal to the other points.
- 3- Extract the non-dominated solutions.
- 4- The process will stop when the data sets to be compared are empty.

### **3.4 Data Collection**

Data are collected from several case studies. A variety of cases are taken into consideration to cover all types of data that the system may have to deal with. Three groups of variables are collected. The first group consists of quantitative variables, which are measured from the jobsite. The second group includes qualitative variables subjected to different scales. The third group involves the related duration for each combination of variables and their related data sets. The collected data for this research is used to develop and validate the developed system.

### **3.5 Verify and Validate the System using Real Data**

After building the developed system, it is verified and validated using real data using a case study. It is necessary to compare the results with other existing systems. It is predicted however that there is a deviation among the results of various systems. The causes of this deviation are discussed and analyzed. Based on the previous analysis, the system is validated and verified. The system validation is conducted via three steps. The first is performed at the end of developing the data mining engine phase. The second is done at the end of the building and designing simulation engine phase. The third is conducted after developing the complete system. In validation step, the developed system allows the user to control the processes. These validation ‘stops’ give the decision maker (i.e. user) the ability to interact with the system during the modeling and simulation processes.

### **3.6 The KEYSTONE Language and System Automation**

A general-purpose construction simulation language (KEYSTONE), an acronym for a KnowlEdge discoverY baSed construcTion simulatiON systEm, is designed and built to

provide a simulation environment platform for modeling construction operations. The KEYSTONE language includes most of the features that a general-purpose simulation language should have. The KEYSTONE language is designed to write and generate object-oriented simulation code. Construction operations are automatically translated into C#. The system is developed under the development platform of KEYSTONE and C#. The C# offers the potential of being available across many platforms. It is a very powerful high-level language, an object-oriented programming language encompassing imperative, declarative, functional, generic, and component-oriented programming language.

### **3.7 Verify and Validate the System using a Case Study**

After building the developed computer/simulation system in the form of a software package, it is verified and validated using a case study. Screen shoots of the software are presented to show the procedure of using the developed system and the actual data flow. This step revalidates the system in its new form as a software package.

# **Chapter 4: Data Collection**

## **4.1 Chapter Overview**

This chapter provides a detailed explanation of the data collection phase including case study project details, the data collection process, and a sample of collected data.

## **4.2 Data Collection**

### **4.2.1 Case Study Project**

The field observations and data collection were carried out for a period of eighteen months on a building under construction: the Engineering, Computer Science and Visual Arts Complex of Concordia University. It is a 17-storey integrated educational facility with a surface area of 86,000 square meters. Concrete pouring operations are considered in this research.

### **4.2.2 Data Collection Process**

The acquisition of data from historical records is usually a less costly and more convenient approach. A part of collected data was collected to study labor productivity by Khan (2005) and Wang (2005). This study is meant to investigate the factors and input parameters that affect productivity. The rest of data are collected from available data records at World Wide Websites, such as Infrastructure Canada and the weather networks. The selected variables are shown in Figure 4.1, which consist of four groups. The first group is for 'weather variables' such as temperature. The second group is the crew variables', such as gang size and labor percentage. The third group is 'project variables' and includes variables, such as work type and method. The fourth group is the

'operational variables', including driver skills and truck status. Some qualitative variables are taken into consideration in each group to study both the quantitative and qualitative variables and their effect on process duration. The descriptions of these variables are given in Table 4.1.

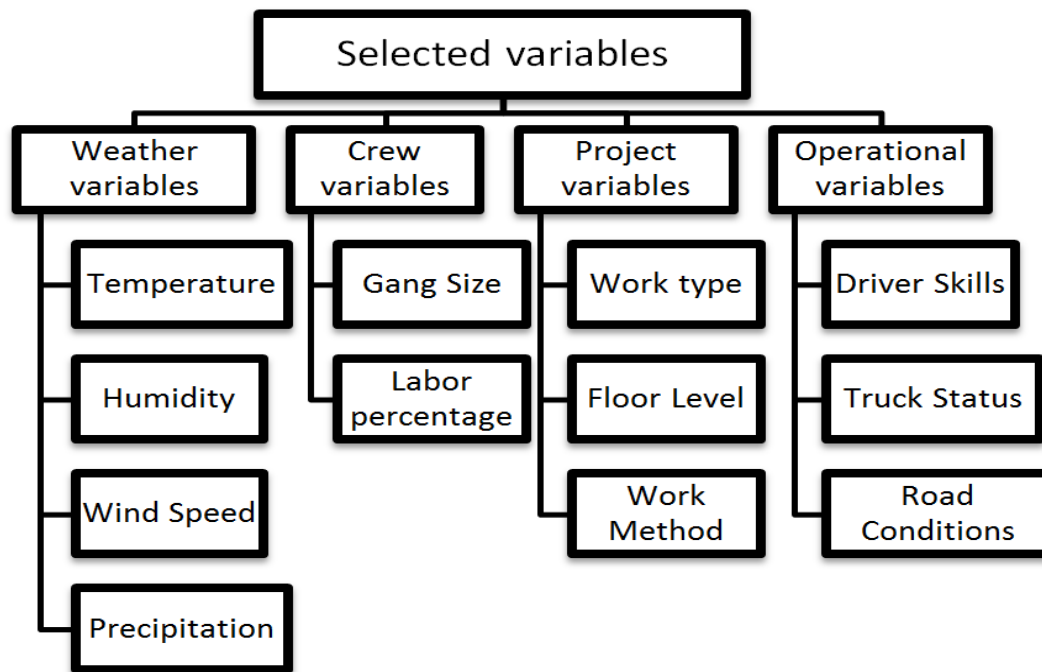


Figure 4-1 The considered variables in the present study

#### 4.2.3 The Collected Data

Data that are recorded and measured from jobsites and the internet provide consistent observations for analyzing the event times of concrete pouring operations. Through these observations and analysis, pouring, hauling, loading and return times are acquired. As shown in Figure 4.2, concrete pouring operation consists of four processes:

##### i. Concrete Pouring Process

The concrete pouring process is the first process in the concrete pouring operation. As shown in Table 4.2, there are eight variables that affect the process duration; two



qualitative and six quantitative. The qualitative variables are: precipitation and method. These variables are collected from the site depending on expert opinion and a scale that was shown in Table 4.1. The quantitative variables are: temperature, humidity, wind speed, gang size, labor percentage and floor level. Those variables are measured directly at the site and collected from referenced bodies such as the weather network. There are 95 data points to be collected and used in the modeling and validation processes.

Table 4-1 Variables description

No.	variables	Description
1	Temperature (°C)	Average of eight working hours of the day
2	Humidity(%)	Average of eight working hours of the day
3	Precipitation	Incorporated in terms of four numerical values as follows: No precipitation = 0, Light rain = 1, Rain = 2, and Snow = 3
4	Wind Speed (km/h)	Average of eight working hours of the day
5	Floor Height	The floor number
6	Work Type	Two types of activities will be considered as follows: Slabs = 1 and walls = 2
7	Gang Size (workers)	Number of workers (skilled + labor) in the gang
8	Labor Percent (%)	The percentage of the labor (non skilled workers) in the gang
9	Driver Skills	1 = Excellent , 2 = Very Good , 3 = Good and 4 = Bad (depending on the number of years of experience)
10	Truck Status	1 = Excellent , 2 = Very Good , 3 = Good and 4 = Bad (depending on the annual safety check for insurance)
11	Road Conditions	1 = Excellent , 2 = Very Good , 3 = Good and 4 = Bad (depending on the number of accidents per year)

## ii. Hauling process

Hauling is the second process in the concrete pouring operation. As shown in Table 4.3, there are six variables that affect the process duration; four qualitative and two quantitative. The qualitative variables are: precipitation, driver skills, truck status, and road condition. These variables are collected from a weather network website

(precipitation) and the Infrastructure Canada website (driver skills, truck status, and road condition). The quantitative variables are: temperature, and wind speed, also collected from a weather network website. One hundred and ten data points were collected to be used in the modeling and validation processes.

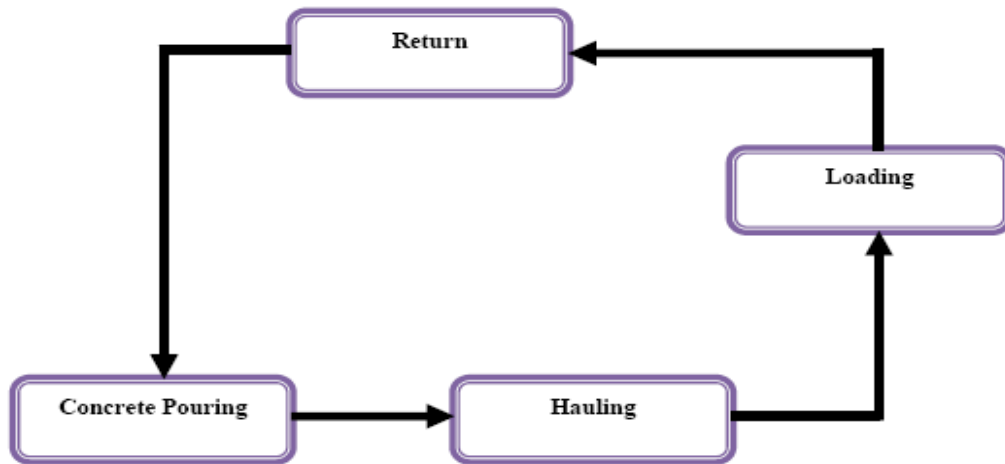


Figure 4-2 Model of the concrete pouring operation

Table 4-2 Variables affecting the concrete pouring process

Temperature °C	Humidity %	Precipitation	Wind speed (K/h)	Gang size (workers)	Labor Percentage %	Floor level	Method	Time (min)
-8.5	46	0	6.6	7	57	2	1	63.00
-15	50	0	12	5	60	2	1	53.70
7	90	1	-	5	60	9	1	53.40
-14.5	42	0	7.5	11	63	4	1	59.10
23	82	0	11	8	-	12	2	75.90
3	97	0	8	11	63	5	1	62.70

Table 4-3 Variables affecting the hauling process

Temperature °C	Precipitation	Wind speed (K/h)	Driver Skills	Truck Status	Road Conditions	Time (min)
6.5	0	11.3	1	2	2	23.75
5.5	0	12	3	3	4	27.50
-5	0	15.8	3	2	4	28.75
7	1	5.4	3	4	4	30.62
6	1	11.9	3	3	3	29.37
15.5	0	18.3	2	4	3	26.25

### iii. Loading process

Loading is the third process in the concrete pouring operation. As shown in Table 4.4, there are six variables that affect the process duration; one qualitative and five quantitative. The qualitative variable is the precipitation, collected from the site depending on expert opinion and a scale that was shown in Table 4.1. The quantitative variables are: temperature, humidity, wind speed, gang size and labor percentage. Those variables are collected from the weather networks website and from the site. One hundred data points were collected to be used in the modeling and validation processes.

Table 4-4 Variables affecting the loading process

Temperature °C	Humidity %	Precipitation	Wind speed (K/h)	Gang size(workers)	Labor Percentage %	Time (min)
23	82	0	11	4	45	30.36
24	82	0	8	5	39	30.96
16	73	1	14	4	36	25.44
15	64	1	19	6	40	21.84
16	60	0	6	5	36	30.36
18	58	0	6	6	40	31.32

### iv. Return process

Returning is the fourth and last process in the concrete pouring operation. As shown in Table 4.5, there are six variables that affect the process duration; four qualitative and two quantitative. The qualitative variables are: precipitation, driver skills, truck status, and road condition. These variables are collected from the site, based on expert opinion and a scale that is shown in table 4.1. The quantitative variables include temperature and wind speed, collected from a weather network website. One hundred and ten data points were collected to be used in the modeling and validation processes. As shown in Figure 4.3, several data points were randomly removed from the data sets to check the validity of the proposed system.

Table 4-5 Variables affecting the return process

Temperature °C	Precipitation	Wind speed (K/h)	Driver Skills	Truck Status	Road Conditions	Time (min)
16	1	14	3	4	4	24.25
13	0	13	2	4	3	23.00
21	0	8	1	2	2	18.50
20	0	23	1	2	2	17.45
16	0	8	1	2	2	16.50
17	0	6	1	3	2	19.00

As indicated in Figure 4.4, the statistical parameters and probability density functions were calculated for all variables that affect the processes. Figure 4.3 shows for each variable the number of points in each data set and number of missing data. In addition to being presented in graphs, some common parameters often can be used to describe sets of numbers. As shown in Figures 4.4 and 4.5, these parameters can measure how a set of numbers is centered around a particular point on a line scale or, in other words, where the numbers bunch together. This category of parameters is called the measures of central tendency, for example mean and median. The parameters’ mean, median, and standard deviation’ are used to calculate the Mean Square Error (MSE), which helps in the variables’ ranking procedure and fuzzy average curves. Probability distributions of activity’s durations are used in the available systems in order to validate its results.

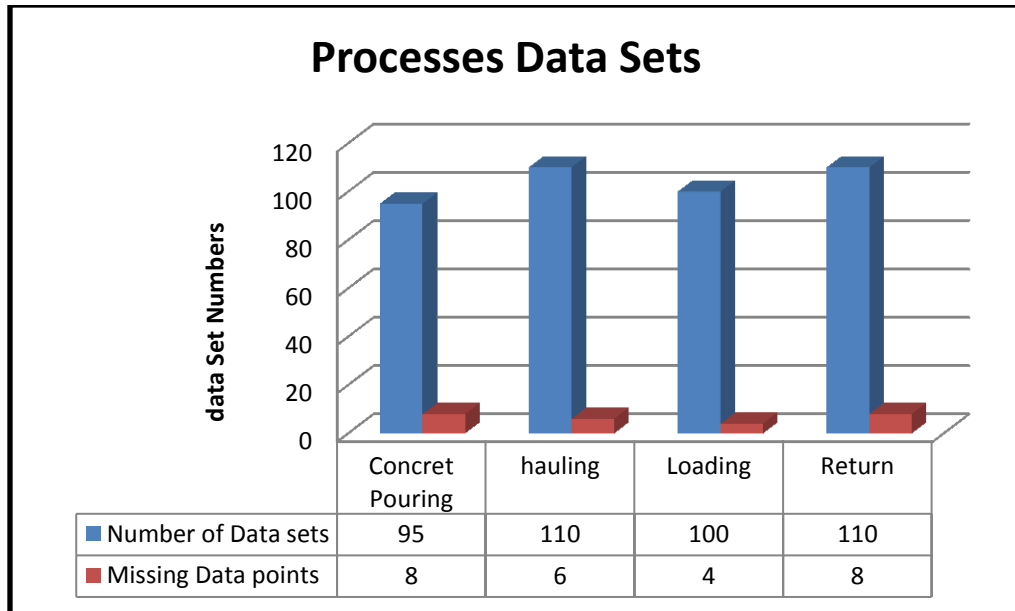


Figure 4-3 The data sets of concrete pouring operation

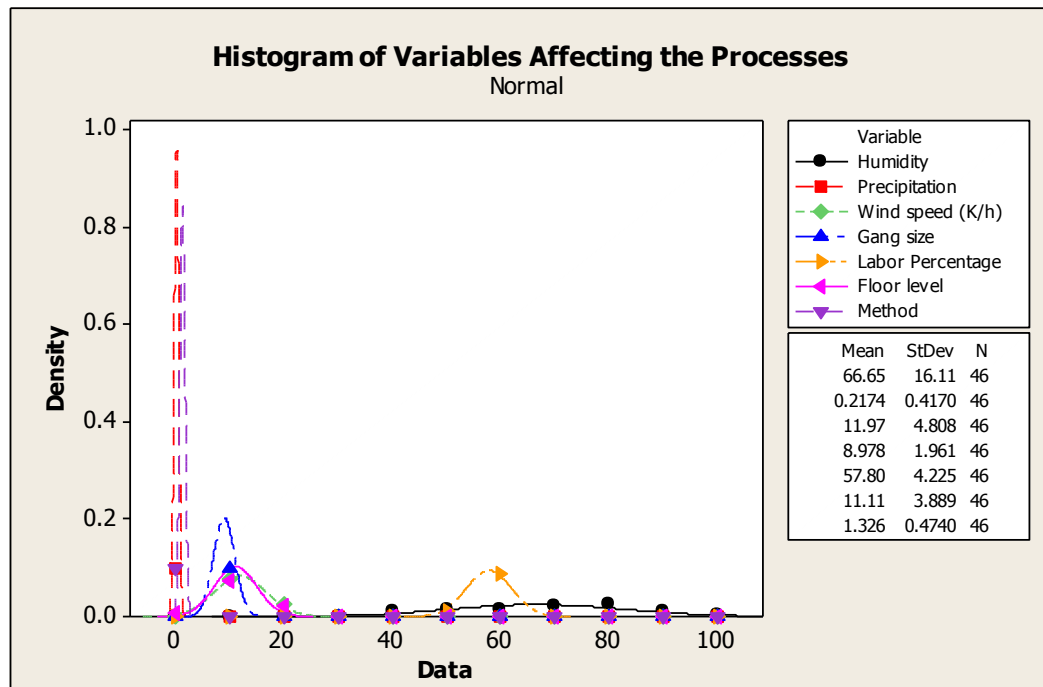


Figure 4-4 Statistical parameters and probability density functions

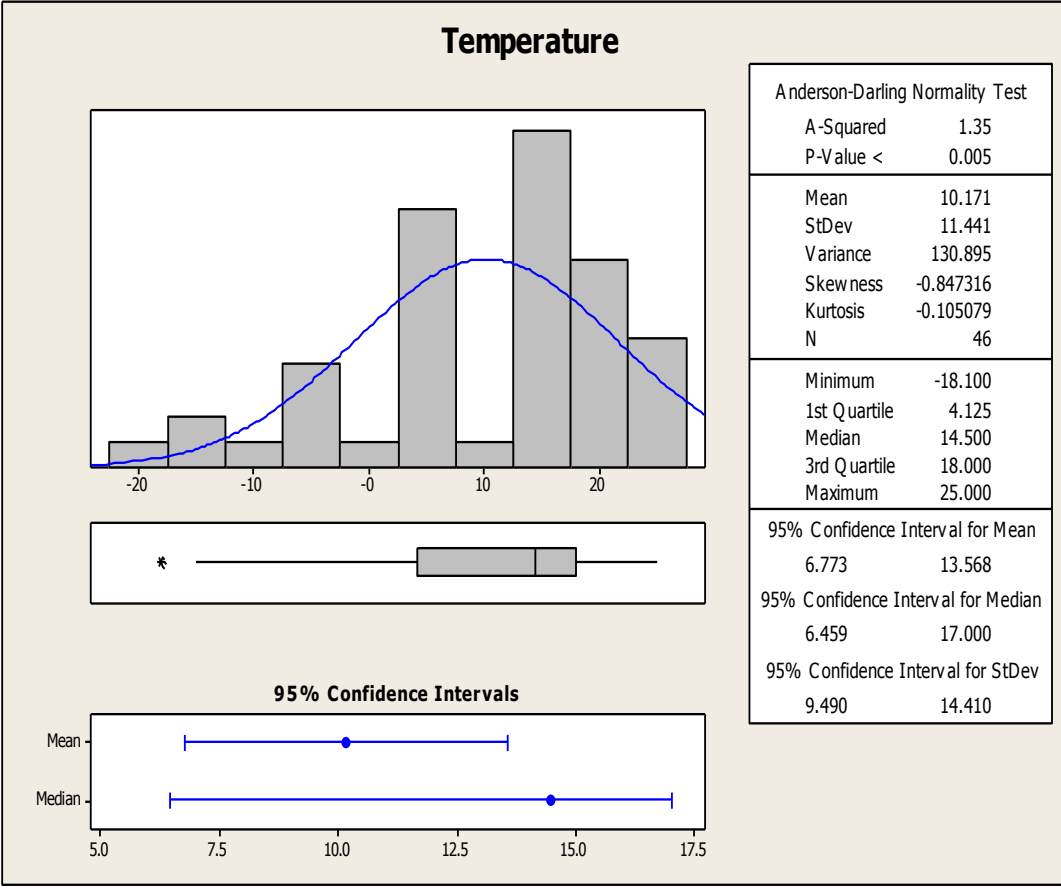


Figure 4-5 Statistical parameters and probability density functions for temperature

# **Chapter 5: Results and Analysis**

## **5.1 Chapter Overview**

In this chapter, the developed system is applied to the collected datasets where the results are analyzed in order to demonstrate a more realistic simulation model application. This chapter consists of three main sections which provide detailed analysis of the case study results. It begins with building a Knowledge Discovery Stage (KDS) for a concrete pouring operation, including cleaning and preparing the data, developing a data mining engine and validating the developed system/models. The second section presents the Simulation Stage (SS), which includes building and validating the simulation methodology. The third section presents the Optimization Stage (OS) in which the sensitivity analysis and the procedure of ranking and selecting optimum solution(s) for resource combinations take place. What follows is a detailed explanation of these three sections and their sub sections.

## **5.2 Building the Knowledge Discovery Stage (KDS)**

In the KDS, raw data is prepared for the simulation stage, and patterns, representing knowledge implicitly stored or captured in large databases, are extracted and made available for processing. The KDS includes data cleaning, integration, selection, transformation, and mining, as well as pattern evaluation and knowledge presentation. The final phase is data mining, which is used to extract hidden knowledge from data sets. In this section, there is a validation strategy wherein each step is validated in comparison with the results of other techniques and methods. Several steps of this stage are discussed

in the previous chapter, such as system definition and data identification. The KDS is developed as depicted in the following steps/sections.

### **5.2.1 Data Cleaning**

Data cleaning is the process of treating missing data and outliers. The fundamental features of data cleaning involve outlier detection, adaptation and estimation. In this step outliers are identified and kept in special clusters to be used in future stages, missing data points are filled in, and the developed method is compared to other available methods.

#### **5.2.1.1 Data Clustering**

The objective of cluster analysis is to classify the objects of data sets according to their similarities in characteristics and subdividing data into groups. Clustering techniques are among the unsupervised methods in which they do not use prior class identifiers to investigate the hidden pattern without output considerations. Fuzzy clustering is used to detect the hidden patterns and structures in data, not only for classification and pattern recognition, but also for model reduction and optimization. A Matlab ® package, Fuzzy library, is used to identify clusters. As shown in Figure 5.1, the 4-D plot of data clusters is illustrated and the identification of outliers, depending on density and based on local outliers, is analyzed. Table 5.1 shows Fuzzy membership values for the clustering results of raw data. Data are grouped into three clusters (C1, C2, and C3) as an initial selection. Each data point belongs to a cluster, to a certain degree that is specified by a membership value. Therefore, a point belongs to all clusters by a degree that is presented by various membership values. As shown in Table 5.1, for example, the first point belongs to cluster one by a membership value equal to 0.5488, to cluster two by 0.2387, and to cluster three by 0.2125. Fuzzy clustering makes a more accurate fit in the case of clusters that are not



well-separated, such as the case of qualitative variables. The fuzziness of the data is treated through fuzzy clustering, which allows any data set value to belong to several clusters and not to one cluster, as with the regular clustering method.

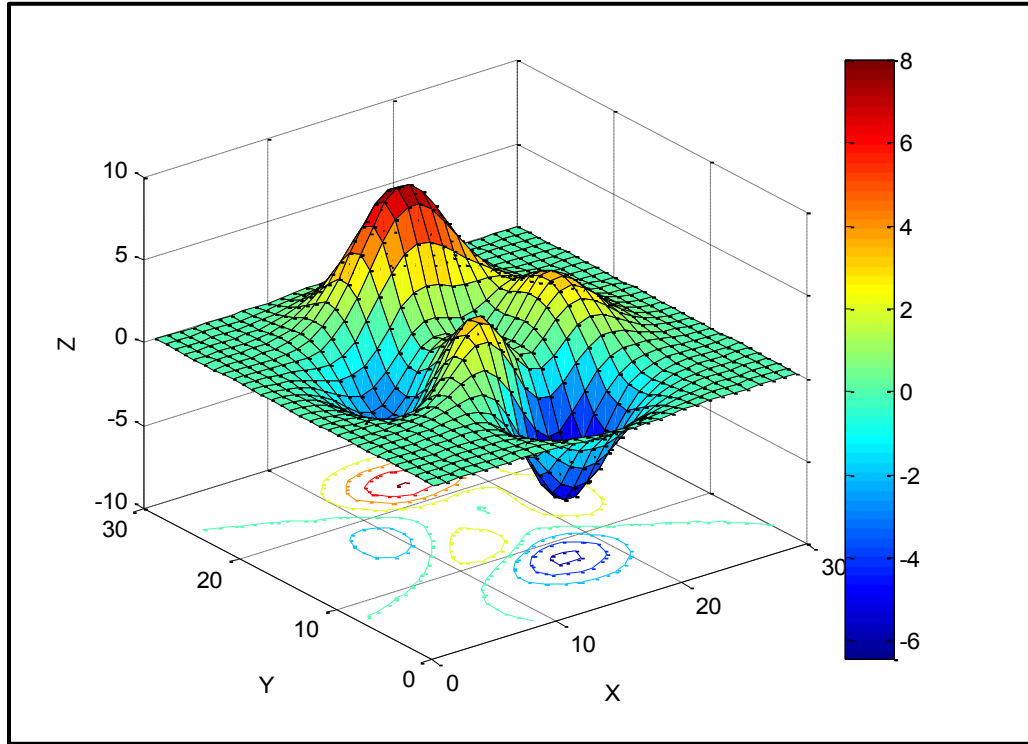


Figure 5-1 (4-D) Plot of data clusters

Table 5-1 Fuzzy membership values

<b>C1</b>	<b>C2</b>	<b>C3</b>
0.5488	0.2387	0.2125
0.6255	0.2049	0.1696
0.5461	0.2571	0.1968
0.7504	0.1273	0.1224
0.7221	0.1411	0.1368
0.6182	0.2067	0.1751
0.4964	0.3138	0.1898
0.6287	0.2007	0.1706
0.5443	0.2758	0.1799
0.5713	0.2433	0.1854
0.4422	0.3323	0.2255
0.4686	0.3416	0.1898
0.4438	0.3140	0.2422
0.4764	0.3072	0.2164
0.6563	0.1799	0.1638
0.4630	0.3096	0.2274

0.5289	0.2885	0.1825
0.6187	0.1751	0.2062
0.3454	0.2165	0.4381
0.4806	0.2342	0.2853
0.5063	0.2639	0.2297
0.5351	0.2522	0.2127
0.5727	0.2482	0.1791
0.4277	0.2974	0.2748
0.4184	0.3372	0.2443
0.4900	0.2588	0.2512
0.5911	0.2388	0.1701
0.4555	0.3339	0.2106
0.1571	0.1432	0.6997
0.5224	0.2316	0.2460
0.3055	0.2070	0.4875
0.2780	0.2142	0.5078
0.6746	0.1809	0.1445

### **5.2.1.2 Missing Data**

Fuzzy K-means clustering is used to fill in the missing data attributes. The calculated membership values are shown in Table 5.1. The membership degree of each data object is considered to compute the clusters' centroids. Equation 3.2 calculates centroids that are then used to identify and distinguish between clusters. The calculated centroids are shown in Table 5.2. Equation 3.3 is used to fill in the missing attributes by replacing the attributes for each incomplete data object  $x_i$ , based on membership degrees and the values of the cluster membership functions. To check the validity of the K-means clustering method, it was compared the other methods where the AVP and AIP are used. Some known values are removed to check the effectiveness of the proposed method to predict missing values. As shown in Table 5.3, to predict the missing values, K-means, Fuzzy K-means, and the average method are used to validate the proposed method by comparing the results. The AVP calculations show that the best method is the Fuzzy K-means method with a result of 85%. The worst method is the average method with an AVP equal to 70%. The Fuzzy approach dealing with fuzzy data and not well-separated clusters, shows better results, while the average method proves to be less accurate because it does not consider the distribution of data sets as well as that of the outliers.

### **5.2.2 Comparison between Data Models before and after Cleaning**

This section addresses the difference between using data cleaning techniques and raw data. This comparison verifies the effect of removing outliers and some missing data on data patterns and behavior. The first case uses raw data that does not include missing values and or outliers to model the relation between the variables affecting concrete pouring process and its duration. The second case shows the possible results after

applying the Fuzzy K-means and keeping the outliers before modeling the same relation. The two cases are presented in the following sections.

Table 5-2 The calculated centroids of data sets

Temperature °C	Humidity %	Precipitation	Wind speed (K/h)	Gang size (workers)	Labor Percentage %	Floor level	Method
12.0467	54.9975	0.1442	13.8985	9.2643	58.3231	13.3456	1.6532
11.0359	59.9995	0.1203	11.8972	9.2806	57.2475	11.3656	1.2495
16.1498	70.5887	0.0590	10.1178	9.8462	55.7028	13.3070	1.7565

Table 5-3 Comparison with filling in the missing values

Point Number	Missing Value	K-means	Fuzzy K-Means	Average
1	10	9.5	8.25	10.15
2	16	15	13.5	10.15
3	65	63	64	57.31
4	72	69	66	65.92
5	1	0	1	1.22
6	12	11	18	9
7	15	14.5	12	9
8	18	19	16.5	9
9	13	12.5	11.75	9
10	5	3.5	6	9
Validation	AIP	17	15	30
	AVP	83	85	70

### 5.2.2.1 Case I: Data modeling after removing missing data and outliers

In this case, the data sets are modeled after removing missing data and outliers. To build a regression model for further analysis, the corresponding data for specific variables is processed using Minitab® software. The output of regression model includes certain coefficients for the specified variables and statistical figures for further analysis. The developed regression model is shown in Equation 5.1.

$$\mathbf{Work\ task\ Duration} = 93.1 + 1.16 \mathbf{T} + 0.0047 \mathbf{H} - 18.8 \mathbf{P} + 0.098 \mathbf{W} + 0.555 \mathbf{G} - 0.357 \mathbf{L} - 2.33 \mathbf{F} + 0.59 \mathbf{M} \quad (\text{Equation 5.1})$$

Where:

*T*: Temperature (°C) of the average of eight working daytime hours

*H*: Humidity (%) of the average of eight daytime working hours

*P*: Precipitation: No precipitation = 0, Light rain = 1, Rain = 2, and Snow = 3

*W*: Wind Speed (Km/h) average of eight daytime working hours

*G*: Number of workers (skilled + labor) in the gang

*L*: The percentage of non-skilled workers (labor) in the gang

*F*: Floor height

*M*: Method that relates to the type of work: Slabs = 1 and walls = 2

In this case,  $R^2$  equal 79.8%, which provides a measure of how well the raw data are scattered around the developed model. The range of  $R^2$  is 0 to 100, so that if the value of  $R^2$  is close to 100, the model is sound and vice versa. The best subset analysis is used to determine the best possible combination of variables with regards to lowest error, lowest variation, and the highest  $R^2$  value. Therefore, the best subset analysis identifies the best-fit regression model that can be constructed with the specified number of variables. As shown in Table 5.4, the best subset is a combination of four instead of five variables, even though the same  $R^2$  is slightly higher for five variables because the lower number of variables in a model is better at presenting the model effectively.

#### **5.2.2.2 Case II: Data modeling after filling missing data and keeping outliers**

In this case, data sets are modeled after filling in missing data and keeping the outliers. The same procedure is followed as stated in Case I. The new developed regression model is shown in Equation 5.2.

$$\begin{aligned} \text{Work task Duration} = & 91.9 + 1.32 T - 0.155 H - 13.20 P - 0.567 W - \\ & 0.652 G + 0.052 L - 1.40 F - 2.70 M \end{aligned} \quad (\text{Equation 5.2})$$

In this case,  $R^2$  equals to 91.3%. As shown in Table 5.5, the best subset is a combination of seven variables. The  $R^2$  in Case II is higher than that of Case I, which means that data are less scattered around the developed model. It is quite clear that the developed model is improved by keeping the outliers and filling in the missing data. It is apparent that data cleaning step affected the modeling process by improving  $R^2$  and by incorporating more variables. It is concluded that removing outliers and missing several data points that contain new data patterns or knowledge inversely affect the modeling and simulation processes.

Table 5-4 Best subset analysis case I

Vars.	R-Sq	R-Sq (adj)	S	Temperature	Humidity	Precipitation	Wind Speed	Gang Size	Labor Percentage	Floor Level	Method
1	26.1	23.8	10.42	*							
1	21.8	19.4	10.71			*					
2	59.5	56.9	7.83	*		*					
2	39.8	36.1	9.54	*						*	
3	78.4	76.4	5.80	*		*				*	
3	65.2	61.8	7.37	*		*					*
4	79.3	76.5	5.78	*		*			*	*	
4	78.7	75.8	5.86	*		*		*		*	
5	79.7	76.2	5.82	*		*		*	*	*	
5	79.3	75.7	5.88	*		*			*	*	*
6	79.8	75.5	5.91	*		*	*	*	*	*	
6	79.7	75.4	5.92	*		*		*	*	*	*
7	79.8	74.6	6.01	*		*	*	*	*	*	*
7	79.8	74.6	6.01	*	*	*	*	*	*	*	
8	79.8	73.6	6.13	*	*	*	*	*	*	*	*

### 5.2.3 Building Data Mining Engine

This phase assists in preparing the data for modeling, building the models, and developing the knowledge base. It is the key component for the simulation stage. The

data mining engine functions are: modeling the processed data, preparing a fuzzy knowledge base, and simulating the predicted environment for future projects. This phase is an essential part of the KDS and ideally consists of a set of functional steps for carrying out the data mining task. The data mining engine phase includes the following steps.

### **5.2.3.1 Selection of variables**

In this step, the variables that affect the output variable (work task duration) are selected using the fuzzy average method. It helps in producing fuzzy curve(s) for and ranks variables. To validate this process, the fuzzy average method's selection is compared to the outputs of Artificial Neural Network (ANN) and Regression methods.

#### **a. Building fuzzy curve**

A fuzzy curve is built by plotting the data points  $(X_i, Y_j)$  in the  $(x_i - y_i)$  space for all variables. Fuzzy membership functions are then created for each data point  $(X_i, Y_j)$  and for each input and output variable using Equation 3.4. A fuzzy curve  $c_i$  point for each set of data points is created by defuzzifying the fuzzy membership functions using Equation 3.5. For example, a fuzzy curve for Temperature is shown in Figure 5.2 and the other variables are shown in Appendix A. If there is a completely random relationship between the input and output variables, the fuzzy curve will be flat; otherwise the curve will be positive or negative in shape, which shows a strong relation.

#### **b. Ranking**

The variables are ranked using the Mean Square error (MSE) as shown in Equation 3.6. If it is relatively large, then, the fuzzy curve is doing a poor job of representing the output with respect to the input and there is a completely random relationship between the input

and output; therefore, the fuzzy curve is flat. If the MSE is small, then, the relationship is more significant. The mean square error MSE is computed as shown in Table 5.6 for each fuzzy curve. The input with the smallest MSE is most important and the input with the largest MSE is the least important. Table 5.6 gives the details for the temperature variable; the other variables are shown in Appendix B. All variables are ranked according to the ascending order of the MSE as shown in table 5.7.

Table 5-5 Best subset analysis case II

<b>Vars.</b>	<b>R-Sq</b>	<b>R-Sq (adj)</b>	<b>S</b>	<b>Temperature</b>	<b>Humidity</b>	<b>Precipitation</b>	<b>Wind Speed</b>	<b>Gang Size</b>	<b>Labor Percentage</b>	<b>Floor Level</b>	<b>Method</b>
1	40.6	38.9	1.45	*							
1	24.1	21.8	11.82			*					
2	70.2	68.4	7.51	*		*					
2	59.6	57.2	8.75	*						*	
3	85.5	84.1	5.32	*		*				*	
3	78.3	76.3	6.50	*		*					*
4	87.2	85.5	5.08	*	*	*				*	
4	86.8	85.1	5.16	*		*				*	
5	90.0	88.3	4.56	*	*	*	*			*	
5	88.5	86.5	4.90	*		*	*			*	*
6	90.9	89.1	4.42	*	*	*	*	*		*	
6	90.5	88.6	4.51	*	*	*	*			*	*
7	91.3	89.2	4.40	*	*	*	*	*		*	*
7	91.0	88.7	4.49	*	*	*	*	*	*	*	
8	91.3	88.8	4.47	*	*	*	*	*	*	*	*

In order to validate the fuzzy average method and how it selects and ranks critical variables, other methods, such as ANN and Regression methods, are used to compare the ranking and model building for the selected variables as shown in the following sections.

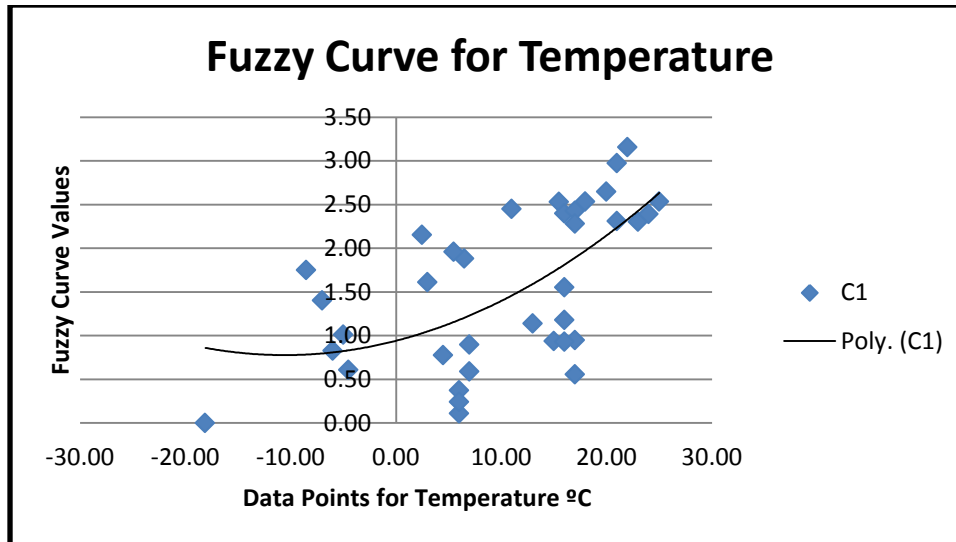


Figure 5-2 Fuzzy curve for temperature

Table 5-6 Fuzzy curve and mean square error calculations

Temperature °C						
X1	X1: Normalized	Y	Y: Normalized	Mi,k(Xi)	C1	(Ci(xi,k)-yk)^2
-8.50	0.22	63.00	0.46	3.79	1.75	11.08
3.00	0.49	62.70	0.46	3.53	1.61	9.46
2.50	0.48	69.90	0.61	3.54	2.15	8.61
4.50	0.52	51.60	0.22	3.50	0.78	10.75
-4.50	0.32	48.90	0.16	3.70	0.61	12.48
-18.10	0.00	41.10	0.00	4.03	0.00	16.21
-6.00	0.28	51.60	0.22	3.73	0.83	12.32
-7.00	0.26	58.80	0.37	3.76	1.40	11.44
6.50	0.57	66.90	0.54	3.46	1.88	8.49
5.50	0.55	67.80	0.56	3.48	1.96	8.50
-5.00	0.30	54.00	0.27	3.71	1.01	11.81
18.00	0.84	78.30	0.78	3.23	2.54	5.98
21.00	0.91	85.50	0.94	3.17	2.97	5.01
25.00	1.00	79.80	0.82	3.10	2.53	5.22
17.00	0.81	54.90	0.29	3.25	0.95	8.75
17.00	0.81	76.50	0.75	3.25	2.43	6.26
13.00	0.72	57.30	0.34	3.33	1.14	8.91
21.00	0.91	75.60	0.73	3.17	2.31	5.98
16.00	0.79	58.20	0.36	3.27	1.18	8.45
17.00	0.81	74.40	0.70	3.25	2.28	6.49
6.00	0.56	42.60	0.03	3.47	0.11	11.81
6.00	0.56	44.40	0.07	3.47	0.24	11.55
7.00	0.58	49.20	0.17	3.45	0.59	10.74
<b>MSE1</b>	8.98					



Table 5-7 Variables ranking using fuzzy average method

Rank	MSE <sub>i</sub>	Variables
3	8.98	Temperature
5	9.86	Humidity
7	11.49	Precipitation
6	10.72	Wind speed
2	8.97	Floor level
8	11.77	Method
1	8.92	Gang Size
4	9.67	Labor Percentage

### 1- Variables Selection Using the ANN Method

The ANN method is used to validate the fuzzy average method of selecting and ranking critical variables. Ranking of input variables is required in order to determine the relative importance of each variable and those that have the most effect on the duration. The contribution percentages are derived from analysis of weights of the trained neural network. The higher the number, the more that variable is contributing to the classification and/or prediction. Obviously, if a certain variable is highly correlated, the variable will have a high contribution percentage. Table 5.8 shows the contribution percentage (relative significance) of eight variables. By comparing the results of fuzzy average method and ANN methods, it is clear that four out of eight variables are similar in the top of each ranking and the other variables have different ranking. In addition, the last-ranked variable is the method variable in both rankings. The  $R^2$  for ANN model is 88.8 %, which is lower than the  $R^2$  of fuzzy average method model. This implies that the fuzzy average method model is more robust than the ANN model when dealing with this variable.

Table 5-8 Variables ranking using ANN

Rank	Contribution	Variables
1	0.3191	Temperature
4	0.1004	Humidity
2	0.1213	Precipitation
6	0.0946	Wind speed
3	0.1200	Floor level
8	0.0686	Method
5	0.1001	Gang Size
7	0.0754	Labor Percentage

## 2- Variables Selection using Regression Method

The Regression method is used to validate the fuzzy average method for selecting and ranking critical variables. The ranking of the input variables is to determine the relative importance of each variable and those that most-affect the duration. Table 5.9 lists the status of the contribution towards the eight variables.

Table 5-9 Variables ranking using regression method

Contribution	Variables
YES	Temperature
YES	Humidity
No	Precipitation
YES	Wind speed
YES	Floor level
YES	Method
YES	Gang Size
NO	Labor Percentage

This method does not present the contribution percentage itself but just shows the best subset of variables that build an effective model. By comparing the results of fuzzy average method and regression methods, it is clear that most of the eight variables have

been included in the model. In addition, two variables are not included, as they have less contribution to the fuzzy average method model. The  $R^2$  for the regression model is 88.0 %, lower than that of fuzzy average method model, which means that the fuzzy average method model is more robust than Regression model. This is one of the reasons why the fuzzy average method is recommended in the case of not unclear relation among input variables and between input and output variables. In addition, the fuzzy approach improved the modeling of qualitative variables and the modeling relation between qualitative variables and process durations.

#### **5.2.3.2 Fuzzy Sets (Fuzzification)**

This step is a process that converts the crisp quantities into fuzzy sets. The fuzzy values are formed by identifying some of the uncertainties present in the crisp values. The conversion to fuzzy values is represented by the membership functions. The Fuzzification process involves assigning membership values for the given crisp quantities. Membership values are assigned using the neural network technique in order to model the relation between the fuzzy membership functions and duration. The membership functions-durations model is used later to predict the duration. The ANN procedure includes the following activities.

##### **a. Assigning membership functions using Neural Networks**

To determine the membership function from modeling data sets of the selected variables, as shown in Figure 3.6, a neural network is created. The training is done for corresponding membership values in different classes to simulate the relationship between coordinate locations and membership values. The neural network uses the set of data value and membership values to train itself. This training process continues until the

neural network can simulate the entire given set of input and output values. After the net is trained, its validity can be checked by testing data. The neural network is then ready to be used to determine the membership values of any input data point. In the building and training neural network step, modeling data, in the form of clusters, is used to build and train the proposed neural network. Variables describing data points that are used to train and build the neural network are illustrated in Table 5-10 and in the following formulas.

012

$$= \frac{1}{1 + \exp[-((0.5 * -8.5) + (0.4 * 46) + (0.1 * 0) + (0.8 * 6.6) + (0.2 * 7) + (0.6 * 57) + (0.9 * 2) + (0.6 * 1) - 0)]}$$

$$= 1$$

Table 5-10 Variables describing data points

<b>Data point</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
<b>X1</b>	-8.5	-15	-14.5	3	2.5	4.5	-4.5	-18.1	-6	-7
<b>X2</b>	46	50	42	97	92	86	48	66	37	41
<b>X3</b>	0	0	0	0	0	1	0	0	0	0
<b>X4</b>	6.6	12	7.5	8	6.2	9.1	14.1	7	19.9	7.9
<b>X5</b>	7	5	11	11	5	11	11	11	7	5
<b>X6</b>	57	60	63	63	60	64	64	64	57	60
<b>X7</b>	2	2	4	5	6	6	6	7	7	7
<b>X8</b>	1	1	1	1	1	1	1	1	1	1

022

$$= \frac{1}{1 + \exp[-((0.25 * -8.5) + (0.6 * 46) + (0.7 * 0) + (0.4 * 6.6) + (0.3 * 7) + (0.7 * 57) + (0.8 * 2) + (0.4 * 1) - 0)]}$$

$$= 1$$

032

$$= \frac{1}{1 + \exp[-((0.3 * -8.5) + (0.8 * 46) + (0.7 * 0) + (0.5 * 6.6) + (0.4 * 7) + (0.8 * 57) + (0.7 * 2) + (0.3 * 1) - 0)]}$$

$$= 1$$

The output for the third layer is:

$$O1^3 = \frac{1}{1 + \exp[-((0.4 * 1) + (0.3 * 1) + (0.2 * 1) - 0)]} = 0.7109$$

$$O2^3 = \frac{1}{1 + \exp[-((0.5 * 1) + (0.5 * 1) + (0.3 * 1) - 0)]} = 0.7858$$

$$O3^3 = \frac{1}{1 + \exp[-((0.6 * 1) + (0.8 * 1) + (0.6 * 1) - 0)]} = 0.8808$$

And the output for the fourth layer is:

$$O14 = \frac{1}{1 + \exp[-((0.8 * 0.7109) + (0.2 * 0.7858) + (0.6 * 0.8808) - 0)]} = 0.7780$$

$$O24 = \frac{1}{1 + \exp[-((0.9 * 0.7109) + (0.4 * 0.7858) + (0.8 * 0.8808) - 0)]} = 0.8400$$

$$O34 = \frac{1}{1 + \exp[-((0.1 * 0.7109) + (0.5 * 0.7858) + (0.4 * 0.8808) - 0)]} = 0.6934$$

Determining errors using Equation 3.8:

$$R1 : E1^4 = 0.2387 - 0.7780 = -0.5393$$

$$R2 : E2^4 = 0.2387 - 0.8400 = -0.6013$$

$$R3 : E3^4 = 0.2125 - 0.6934 = -0.4809$$

$$E1^3 = 0.7109 (1.0 - 0.7109)[(0.80 * -0.5393) + (0.90 * -0.6013) + (0.10 * -0.4809)] = -0.2098$$

$$E2^3 = 0.7858 (1.0 - 0.7858)[(0.20 * -0.5393) + (0.40 * -0.6013) + (0.50 * -0.4809)] = -0.0991$$

$$E3^3 = 0.8808 (1.0 - 0.8808)[(0.60 * -0.5393) + (0.80 * -0.6013) + (0.40 * -0.4809)] = -0.1047$$

The errors are assigned to the elements in the second layer as follows:

$$E1^2 = 1.0 (1.0 - 1.0)[(0.40 * -0.2098) + (0.50 * -0.0991) + (0.60 * -0.1047)] = 0$$

$$E2^2 = 1.0 (1.0 - 1.0)[(0.30 * -0.2098) + (0.50 * -0.0991) + (0.80 * -0.1047)] = 0$$

$$E3^2 = 1.0 (1.0 - 1.0)[(0.20 * -0.2098) + (0.30 * -0.0991) + (0.60 * -0.1047)] = 0$$

The weights are updated using the errors associated with each element in the network by using Equation 3.10 so that the network more closely approximates the output. The fourth and third layer connecting weights are updated as follows:

$$w11^3 = 0.8 + 0.30 * -0.5393 * 0.7109 = 0.6850$$

$$w21^3 = 0.2 + 0.30 * -0.5393 * 0.7858 = 0.0729$$

$$w31^3 = 0.6 + 0.30 * -0.5393 * 0.8808 = 0.4575$$

$$\begin{aligned}
w12^3 &= 0.9 + 0.30 * -0.6013 * 0.7109 = 0.7718 \\
w22^3 &= 0.4 + 0.30 * -0.6013 * 0.7858 = 0.2582 \\
w32^3 &= 0.8 + 0.30 * -0.6013 * 0.8808 = 0.6411 \\
w13^3 &= 0.1 + 0.30 * -0.4809 * 0.7109 = -0.0026 \\
w23^3 &= 0.5 + 0.30 * -0.4809 * 0.7858 = 0.3866 \\
w33^3 &= 0.4 + 0.30 * -0.4809 * 0.8808 = 0.2729
\end{aligned}$$

The third layer and the second layer connecting weights are updated as follows:

$$\begin{aligned}
w11^2 &= 0.4 + 0.30 * -0.2098 * 1 = 0.3371 \\
w21^2 &= 0.3 + 0.30 * -0.2098 * 1 = 0.2371 \\
w31^2 &= 0.2 + 0.30 * -0.2098 * 1 = 0.1371 \\
w12^2 &= 0.5 + 0.30 * -0.0991 * 1 = 0.4703 \\
w22^2 &= 0.5 + 0.30 * -0.0991 * 1 = 0.4703 \\
w32^2 &= 0.3 + 0.30 * -0.0991 * 1 = 0.2703 \\
w13^2 &= 0.6 + 0.30 * -0.1047 * 1 = 0.5686 \\
w23^2 &= 0.8 + 0.30 * -0.1047 * 1 = 0.7686 \\
w33^2 &= 0.6 + 0.30 * -0.1047 * 1 = 0.5686
\end{aligned}$$

The second and first layer connecting weights are correct because the associated error is zero. This step is repeated until the minimum acceptance error is reached.

### 5.2.3.3 Fuzzy Rules Induction

The goal of this step is to find the functional relationship between independent variables' membership values and dependent variables (Task Duration) to model the entire relation as shown in Table 5.11. Actual models of the underlying processes are generated. The induction rule creates a knowledge base of fuzzy if-then rules. These rules describe and model one or more behaviors in a large collection of data sets. The functional relationships between independent (i.e. variables affecting the process) and dependent (i.e. process duration) variables are expressed as a set of fuzzy if-then rules. Fuzzy rule induction is as follows:

**a. Generation of the candidate fuzzy rules**

The inputs of this process are fuzzy sets, model variables and the system definitions step, which were discussed earlier. In the generating a candidate rules’ process, fuzzy relations are produced from the dependent and independent variables. Fuzzy relations are converted to “If-Then” rules, as shown in Table 5.12.

**b. Selection of a final rule set**

The final rules will be selected from all of the candidate rules presented in Table 5.12. There are many repeated rules that represent the same duration. The effectiveness of each rule is calculated using Equation 3.11. The highest effectiveness-degree rules are selected, as shown in Table 5.13. For instance, Rule 36 is replaced by Rule 24 due to the degree of effectiveness. The selection of the final rules will decrease the number of rules without affecting the performance of the rules system.

Table 5-11 Fuzzy cluster membership values and task durations

<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>Task Duration(min)</b>
0.5488	0.2387	0.2125	63.0
0.6255	0.2049	0.1696	53.7
0.5461	0.2571	0.1968	59.1
0.7503	0.1273	0.1224	62.7
0.7221	0.1411	0.1368	69.9
0.6182	0.2067	0.1751	51.6
0.4964	0.3138	0.1898	48.9
0.6287	0.2007	0.1706	41.1
0.5443	0.2758	0.1799	51.6
0.5713	0.2433	0.1854	58.8
0.4422	0.3323	0.2255	62.1
0.4686	0.3416	0.1898	66.9
0.4438	0.314	0.2422	67.8
0.4764	0.3072	0.2164	54.0
0.6563	0.1799	0.1638	53.4

#### **5.2.3.4 Fuzzy Knowledge Base (FKB)**

A fuzzy knowledge base includes all the previous steps, i.e. the selected variables, fuzzy sets, and fuzzy rules. As shown in Table 5.14, the FKB is a representation of a particular model of each process that means the FKB of concrete pouring process is separate from the loading process. A collection of FKBs cooperates to form a solution to a concrete pouring operation. The FKB will not be finalized until it has been validated.

#### **5.2.3.5 Validation**

To test the effectiveness of the system's prediction, a validation data set is embedded into the developed FKB to compare its results with actual data. The first test is the actual versus the predicted plots. These plots show that the predicted values are within acceptable limits scattered around the actual values for the response variable (task duration) as shown in Figure 5.3. The results of first validation test are clearly satisfactory. The second test is for mathematical and descriptive validation. Equations 3.12 to 3.15 are used. Equation 3.12 represents the average invalidity percent (AIP), which reveals the prediction error. If the AIP value is closer to 0.0%, a model is sound and a value closer to 1 shows that a model is not appropriate. Similarly, the root mean square error (RMSE) can be estimated using Equation 3.14. If the value of the RMSE is close to 0, the model is sound and vice versa. In addition, the mean absolute error (MAE) is defined as shown in Equation 3.15. The MAE value varies from 0 to infinity. However, the value of the mean absolute error should be close to zero for sound results. Table 5.15 shows the summary of the validation results for the developed knowledge base.



Table 5-12 A sample of candidate rules

<b>Rule 1</b>	<b>If C1 is</b> 0.5488 <b>and C2 is</b> 0.2387 <b>and C3 is</b> 0.2125 <b>Then Time is</b> 63.0
<b>Rule 2</b>	<b>If C1 is</b> 0.6255 <b>and C2 is</b> 0.2049 <b>and C3 is</b> 0.1696 <b>Then Time is</b> 53.7
<b>Rule 3</b>	<b>If C1 is</b> 0.5461 <b>and C2 is</b> 0.2571 <b>and C3 is</b> 0.1968 <b>Then Time is</b> 59.1
<b>Rule 4</b>	<b>If C1 is</b> 0.7504 <b>and C2 is</b> 0.1273 <b>and C3 is</b> 0.1224 <b>Then Time is</b> 62.7
<b>Rule 5</b>	<b>If C1 is</b> 0.7221 <b>and C2 is</b> 0.1411 <b>and C3 is</b> 0.1368 <b>Then Time is</b> 69.9
<b>Rule 6</b>	<b>If C1 is</b> 0.6182 <b>and C2 is</b> 0.2067 <b>and C3 is</b> 0.1751 <b>Then Time is</b> 51.6
<b>Rule 7</b>	<b>If C1 is</b> 0.4964 <b>and C2 is</b> 0.3138 <b>and C3 is</b> 0.1898 <b>Then Time is</b> 48.9
<b>Rule 8</b>	<b>If C1 is</b> 0.6287 <b>and C2 is</b> 0.2007 <b>and C3 is</b> 0.1706 <b>Then Time is</b> 41.1
<b>Rule 9</b>	<b>If C1 is</b> 0.5443 <b>and C2 is</b> 0.2758 <b>and C3 is</b> 0.1799 <b>Then Time is</b> 51.6
<b>Rule 10</b>	<b>If C1 is</b> 0.5713 <b>and C2 is</b> 0.2433 <b>and C3 is</b> 0.1854 <b>Then Time is</b> 58.8
<b>Rule 11</b>	<b>If C1 is</b> 0.4422 <b>and C2 is</b> 0.3323 <b>and C3 is</b> 0.2255 <b>Then Time is</b> 62.1
<b>Rule 12</b>	<b>If C1 is</b> 0.4686 <b>and C2 is</b> 0.3416 <b>and C3 is</b> 0.1898 <b>Then Time is</b> 66.9
<b>Rule 13</b>	<b>If C1 is</b> 0.4438 <b>and C2 is</b> 0.3140 <b>and C3 is</b> 0.2422 <b>Then Time is</b> 67.8
<b>Rule 14</b>	<b>If C1 is</b> 0.4764 <b>and C2 is</b> 0.3072 <b>and C3 is</b> 0.2164 <b>Then Time is</b> 54
<b>Rule 15</b>	<b>If C1 is</b> 0.6563 <b>and C2 is</b> 0.1799 <b>and C3 is</b> 0.1638 <b>Then Time is</b> 53.4
<b>Rule 16</b>	<b>If C1 is</b> 0.4630 <b>and C2 is</b> 0.3096 <b>and C3 is</b> 0.2274 <b>Then Time is</b> 46.2
<b>Rule 17</b>	<b>If C1 is</b> 0.4829 <b>and C2 is</b> 0.3323 <b>and C3 is</b> 0.1848 <b>Then Time is</b> 77.7
<b>Rule 18</b>	<b>If C1 is</b> 0.5289 <b>and C2 is</b> 0.2885 <b>and C3 is</b> 0.1825 <b>Then Time is</b> 75.6
<b>Rule 19</b>	<b>If C1 is</b> 0.6187 <b>and C2 is</b> 0.1751 <b>and C3 is</b> 0.2062 <b>Then Time is</b> 60.3
<b>Rule 20</b>	<b>If C1 is</b> 0.3454 <b>and C2 is</b> 0.2165 <b>and C3 is</b> 0.4381 <b>Then Time is</b> 76.2
<b>Rule 21</b>	<b>If C1 is</b> 0.4806 <b>and C2 is</b> 0.2342 <b>and C3 is</b> 0.2853 <b>Then Time is</b> 75.9
<b>Rule 22</b>	<b>If C1 is</b> 0.5063 <b>and C2 is</b> 0.2639 <b>and C3 is</b> 0.2297 <b>Then Time is</b> 77.4
<b>Rule 23</b>	<b>If C1 is</b> 0.5351 <b>and C2 is</b> 0.2522 <b>and C3 is</b> 0.2127 <b>Then Time is</b> 63.6
<b>Rule 24</b>	<b>If C1 is</b> 0.5727 <b>and C2 is</b> 0.2482 <b>and C3 is</b> 0.1791 <b>Then Time is</b> 54.6
<b>Rule 25</b>	<b>If C1 is</b> 0.4277 <b>and C2 is</b> 0.2974 <b>and C3 is</b> 0.2748 <b>Then Time is</b> 75.9
<b>Rule 26</b>	<b>If C1 is</b> 0.4184 <b>and C2 is</b> 0.3372 <b>and C3 is</b> 0.2443 <b>Then Time is</b> 78.3

The results show that the average validity percent is 92%, the RMSE is 2, and the MAE is 5. Therefore, the developed fuzzy knowledge base is acceptable, robust, and can be recommended for further stages. The reason for setting more than one validation point in this system is to improve the simulation process and give the decision maker a chance to improve the entire modeling process. For instance, if there is a no acceptable validation result in a validation point after the data mining phase, the decision maker can improve

the data sets by increasing the number of data points for each process or by collecting other data sets with different data quality rules. After this validation has been satisfactorily completed, the Knowledge Discovery Stage (KDS) is ready to do its work and present the data to the Simulation Stage (SS). The differences between the data prepared by the KDS and the data and procedures that are used by other available simulation systems are that: (1) the data now is clean and has no missing data and fuzziness; (2) the outliers are separated in special clusters to be used in the simulation process in order to have access to all data patterns and behaviors; (3) the qualitative and quantitative variables are now considered in the prediction of the processes' durations;(4) the duration prediction is dependent upon the historical data not on the approximate probabilities and expert opinions to remove some data patterns and the variables affecting these distributions; (5) the input data now is controlled through a validation procedure to interact with the modeler to improve the modeling and simulation procedure; and (6) the procedure for the durations' prediction is easier and more accurate.

Table 5-13 A sample of the rules' effectiveness degrees

<b>Rules</b>	<b>Degree of effectiveness</b>	<b>Rules</b>	<b>Degree of effectiveness</b>
<b>Rule 1</b>	0.012	<b>Rule 19</b>	0.006
<b>Rule 2</b>	0.007	<b>Rule 20</b>	0.021
<b>Rule 3</b>	0.013	<b>Rule 21</b>	0.016
<b>Rule 4</b>	0.002	<b>Rule 22</b>	0.016
<b>Rule 5</b>	0.003	<b>Rule 23</b>	0.014
<b>Rule 6</b>	0.007	<b>Rule 24</b>	0.011
<b>Rule 7</b>	0.019	<b>Rule 25</b>	0.024
<b>Rule 8</b>	0.007	<b>Rule 26</b>	0.028
<b>Rule 9</b>	0.014	<b>Rule 27</b>	0.022
<b>Rule 10</b>	0.011	<b>Rule 28</b>	0.004
<b>Rule 13</b>	0.024	<b>Rule 31</b>	0.026
<b>Rule 17</b>	0.020	<b>Rule 35</b>	0.017
<b>Rule 18</b>	0.015	<b>Rule 36</b>	0.010

Table 5-14 A sample of a fuzzy knowledge base

Process Variables									Membership Functions			Fuzzy Rules								
V1	V2	V4	V5	V6	V7	V8	V9	Time	C1	C2	C3									
-8.5	46	0	6.6	7	57	2	1	63	0.55	0.23	0.21	Rule 1	If C1 is	0.23	and C2 is	0.23	and C3 is	0.21	Then Time is	63.0
-15	50	0	12	5	60	2	1	53.7	0.63	0.2	0.16	Rule 2	If C1 is	0.20	and C2 is	0.20	and C3 is	0.16	Then Time is	53.7
-14.5	42	0	7.5	11	63	4	1	59.1	0.55	0.25	0.19	Rule 3	If C1 is	0.25	and C2 is	0.25	and C3 is	0.19	Then Time is	59.1
3	97	0	8	11	63	5	1	62.7	0.75	0.12	0.12	Rule 4	If C1 is	0.12	and C2 is	0.12	and C3 is	0.12	Then Time is	62.7
2.5	92	0	6.2	5	60	6	1	69.9	0.72	0.14	0.13	Rule 5	If C1 is	0.14	and C2 is	0.14	and C3 is	0.13	Then Time is	69.9
4.5	86	1	9.1	11	64	6	1	51.6	0.62	0.2	0.17	Rule 6	If C1 is	0.20	and C2 is	0.20	and C3 is	0.17	Then Time is	51.6
-4.5	48	0	14.1	11	64	6	1	48.9	0.50	0.31	0.18	Rule 7	If C1 is	0.31	and C2 is	0.31	and C3 is	0.18	Then Time is	48.9
-18.1	66	0	7	11	64	7	1	41.1	0.63	0.2	0.17	Rule 8	If C1 is	0.20	and C2 is	0.20	and C3 is	0.17	Then Time is	41.1
-6	37	0	19.9	7	57	7	1	51.6	0.55	0.27	0.17	Rule 9	If C1 is	0.27	and C2 is	0.27	and C3 is	0.17	Then Time is	51.6
-7	41	0	7.9	5	60	7	1	58.8	0.57	0.24	0.18	Rule 10	If C1 is	0.24	and C2 is	0.24	and C3 is	0.18	Then Time is	58.8

Table 5-15 A sample of fuzzy knowledge base validation results

C1	C2	C3	Time(min)	Estimated Time (min)	AIP	(Ci - Ei)^2	ABS (Ci - Ei)
0.6255	0.2049	0.1696	53.7	52.9261	0.0144	0.5989	0.7739
0.5461	0.2571	0.1968	59.1	57.3813	0.0291	2.9539	1.7187
0.4422	0.3323	0.2255	62.1	65.3717	0.0527	10.7041	3.2717
0.6187	0.1751	0.2062	60.3	57.8351	0.0409	6.0757	2.4649
0.3454	0.2165	0.4381	76.2	68.3471	0.1031	61.6678	7.8529
0.1668	0.1529	0.6803	71.4	74.8541	0.0484	11.9311	3.4541
0.49	0.2588	0.2512	69.3	56.7075	0.1817	158.5717	12.5925
0.5224	0.2316	0.246	70.2	54.7342	0.2203	239.1914	15.4658
0.5254	0.2468	0.2278	55.8	57.0190	0.0218	1.4861	1.2190
0.4208	0.2456	0.3336	55.5	61.3393	0.1052	34.0975	5.8393
AIP (%)	8	AVP (%)	92	RMSE	2	MAE	5

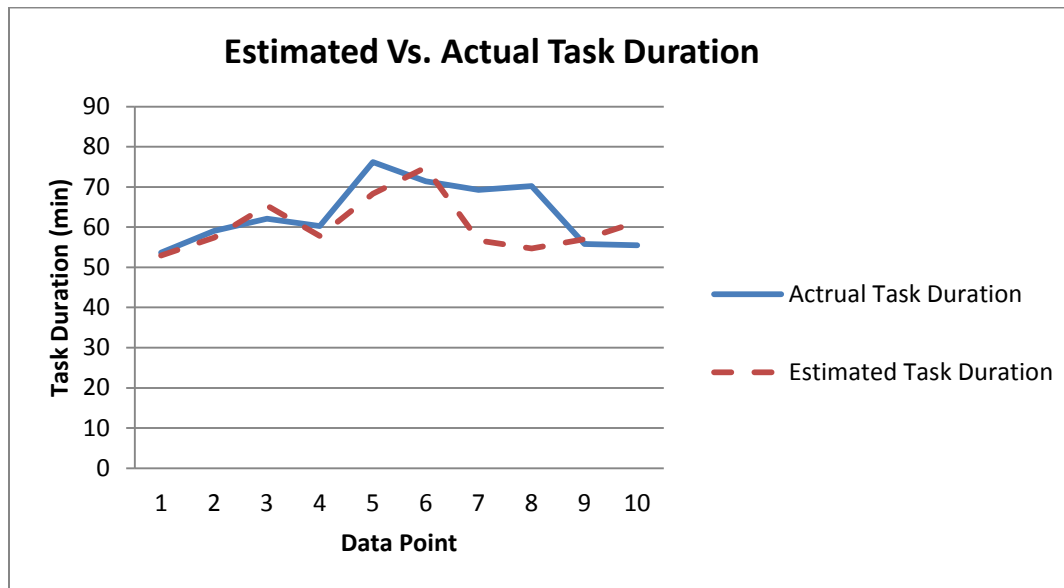


Figure 5-3 Estimated vs. actual task duration

### 5.3 Simulation Stage (SS)

In this stage, the movement of units (i.e. trucks) is modeled through all the processes. The objective of developing this stage is to model and examine the interaction between flow units (trucks) and the idle times of resources in order to flag bottlenecks and estimate

productivity of the developed system. In this stage, the simulation engine is also designed and validated.

### **5.3.1 Simulation Engine Design and Building Phase**

The simulation engine is designed and built in this phase. The framework of the engine is implemented within the “during construction” and “planning” stages. The inputs of this engine are the outputs of Data Mining Engine, fuzzy knowledge base and project database. The project is considered in two cases to verify the system performance under both the “during construction” and “planning” stage conditions. In planning stage, the input variables that affect duration are identified, studied, and transformed into fuzzy membership functions via data mining engine. The fuzzy knowledge base is then used to predict task durations. During construction stage, variables and durations are collected from real operations. The data mining engine is then used to build the environment of simulation process. Thereafter, it interacts with the discrete event simulation. The simulation engine design and building are carried out via the following steps:

#### **5.3.1.1 During the planning stage**

In planning stage, the variables that affect process duration are identified and defined using upper and lower bounds as shown in Table 5.16. Those limits are dependent upon the expected variables’ range in the project under study. For instance, the temperature in the project area could be -15 to -25 °C at the expected time of construction. The data mining engine will transform the data sets (variables only) to fuzzy membership functions. The fuzzy knowledge base (FKB) is then used to predict task durations, as shown in Table 5.17, since the fuzzy membership functions are now available as input for the FKB. These durations, in turn, are the input for the discrete event simulation. The

discrete event simulation system starts forming the simulation event list at the TNOW (the time at the first run) equal to zero point in the first run. The simulation clock keeps the simulation time. If the work task cannot achieve this condition, the simulation clock will be advanced and another process will be checked. Thereafter, the procedure will be as follows: (1) moving units are moved to another process, (2) the predicted durations are used, and (3) the end event time is calculated by adding the durations to the TNOW and the end of event time (EET) is recorded in the event list. The chronological list is produced based on the events list, as shown in Table 5.18. The productivity in this case is calculated as:  $60 / 140.37$  equal to 0.472 cycle / hour.

Table 5-16 The range of variables affecting the concrete pouring process

Temperature °C	Humidity %	Precipitation	Wind speed (K/h)	Gang size (workers)	Labor Percentage %	Floor level	Method
(15) to 25	30 to 85	0 to 2	0 to 25	5 to 15	55 to 70	1 to 15	1 to 2

Table 5-17 Predicted task durations during planning stage

Cluster 1	Cluster 2	Cluster 3	Duration (min)
0.6255	0.2049	0.1696	52.93
0.5461	0.2571	0.1968	57.38
0.4422	0.3323	0.2255	65.37
0.6187	0.1751	0.2062	57.84
0.3454	0.2165	0.4381	68.35
0.1668	0.1529	0.6803	74.85
0.4900	0.2588	0.2512	56.71
0.5224	0.2316	0.2460	54.73
0.5254	0.2468	0.2278	57.02
0.4208	0.2456	0.3336	61.34

### 5.3.1.2 During the construction stage

In the construction stage, the variables that affect the process duration are identified and measured at the site. The combinations of variables and durations which measured at several times for the proposed process are shown in Table 5.19. The concrete pouring

process durations are predicted using these cases. The data mining engine is trained using the process states, depending on the cases that are measured from the site. The data mining engine predicts process durations, as shown in Table 5.20, without using the fuzzy knowledge base. After transforming the variables to fuzzy clusters, the data mining engine uses the clusters to build fuzzy rules (IF-THEN) to form the simulation environment. The discrete event simulation system starts to form the simulation event list at the TNOW equals to zero point in the first run.

Table 5-18 Simulation event list

Process	TNOW	Durations (min)	E.E.T	Cycles	
3	0	52.92	52.92		
3	0	57.38	57.38		
3	0	65.37	65.37		
4	52.92	20	72.92		
4	57.38	20	77.38		
4	65.37	20	85.37		
5	72.92	30	102.92		
5	77.38	30	107.38		
5	85.37	30	115.37		
7	102.92	25	127.92		Cycle 1
7	107.38	25	132.38		Cycle 2
7	115.37	25	140.37		Cycle 3

Table 5-19 Concrete pouring process states

Temperature °C	Humidity %	Precipitation	Wind speed (K/h)	Gang size (workers)	Labor Percentage %	Floor level	Method	Duration (min)
-15	50	0	12	5	60	2	1	53.7
-14.5	42	0	7.5	11	63	4	1	59.1
-0.05	68	0	7.2	11	63	8	1	62.1
25	83	1	10	10	60	12	2	60.3
24	82	0	8	11	55	12	2	76.2
18	79	0	10	9	56	14	2	71.4
18	71	0	19	8	63	14	2	69.3
20	73	0	23	9	56	15	2	70.2
13	64	0	19	8	63	16	2	55.8
14	81	0	13	7	57	16	2	55.5

Table 5-20 Predicted task duration during construction

Cluster 1	Cluster 2	Cluster 3	Duration (min)
0.6287	0.2007	0.1706	56.24
0.4630	0.3096	0.2274	54.68
0.4829	0.3323	0.1848	61.59
0.5290	0.2885	0.1825	63.63
0.3454	0.2165	0.4381	68.35
0.4278	0.2974	0.2748	74.71
0.4185	0.3372	0.2443	86.40

If the work task cannot achieve this condition, the simulation clock will be advanced and another process will be checked. Thereafter, similar procedure to the planning stage is implemented. The chronological list is then produced based on the events list as shown in Table 5.21. The productivity in this case is calculated as:  $60 / 136.59$  equal to 0.439 cycle / hour.

Table 5-21 Simulation event list

Process	TNOW	Durations (min)	E.E.T	Cycles
3	0	56.24	56.24	
3	0	54.68	54.68	
3	0	61.59	61.59	
4	56.24	20	76.24	
4	54.68	20	74.78	
4	61.59	20	81.59	
5	76.24	30	106.24	
5	74.78	30	104.78	
5	81.59	30	111.59	
7	106.24	25	131.24	Cycle 1
7	104.78	25	129.78	Cycle 2
7	111.59	25	136.59	Cycle 3

To validate the simulation process, the results of different simulation methodologies are compared. The developed simulation stage's results, productivity and cost per unit time, are compared to those based on standard simulation methodology and deterministic calculations.



**i. Productivity based on simulation methodology**

Simulation models are adopted for the analysis of concrete pouring operation as shown in Figure 5.4. The duration of each process and resource information, activity and resources, is given as shown in Tables 5.22 and 5.23. The code for construction operation is built with the Web cyclone system and presented in Figure 5.5. The result of running a simulation model with all of the information described in this section comes to 0.374 Cycles/hour as indicated in Figure 5.6. A sensitivity analysis, shown in Table 5.24, is conducted by changing one resource while fixing all other resources. This change will affect productivity and cost per unit time for each alternative in turn. WebCyclone and Ezstrobe are used both to model the operation and to analyze the sensitivity. This solution is then compared to the results of developed system.

**ii. Productivity based on deterministic calculations**

To find the productivity based on the deterministic model, the time durations are assumed to be fixed or constant values that do not vary over time. Any variability in this method is assumed to be small or insignificant. The productivity of the concrete pouring operation can then be determined as follows:

Truck Cycle Time:

Pouring Concrete	62.3 min
Hauling	20.0 min
Loading	30.0 min
Return	25.0 min

Total Truck Time = 137.3

Productivity =  $60/137.3 = 0.437$  Cycle/hour

Labor Cycle Time:

Pouring Concrete            62.3 min

Productivity =  $60/62.3 = 0.963$  Cycle/hour

The truck cycle time will control the productivity at 0.437cycle/hour. The balance point is:

Balance point: B.P:  $0.963/0.437 = 3$  Trucks

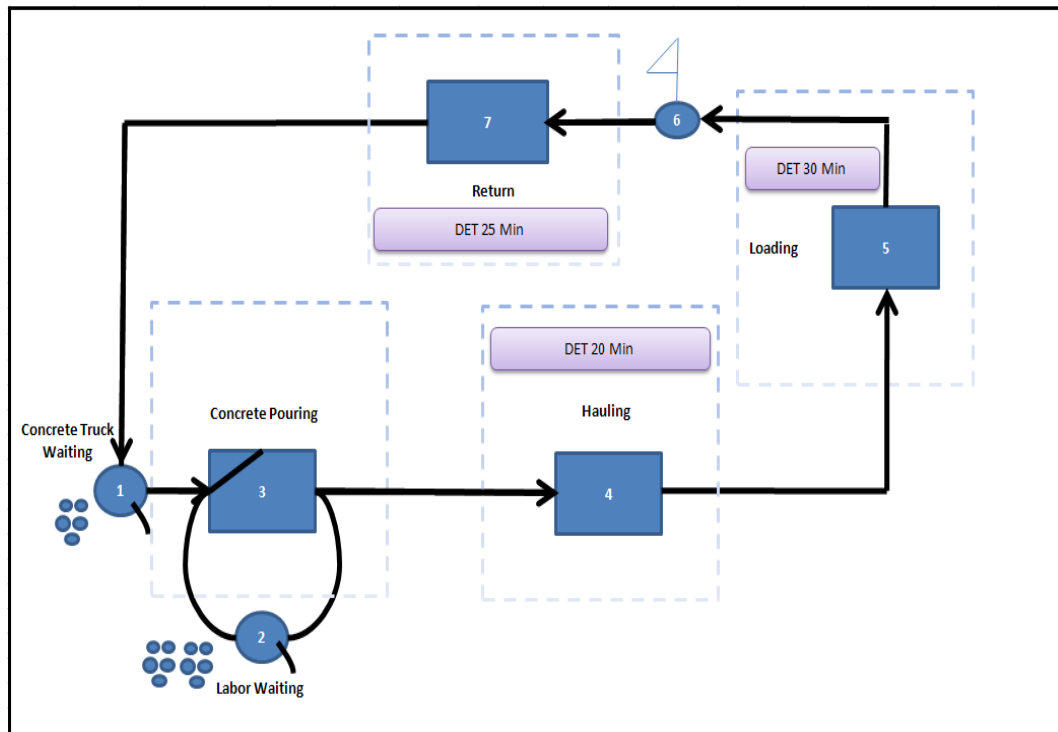


Figure 5-4 Concrete pouring operation model

**5.3.2 Validation Phase**

The goal of this phase is to examine the system modeling effectiveness in order to validate the developed simulation models. In the ‘during construction’ stage, the simulation results will be compared to the actual results of productivity and of queue waiting times. If the percent of difference is acceptable, the model can be considered successful; otherwise the model is rejected.

Table 5-22 Process duration information

<b>Node</b>	<b>Activity</b>	<b>Duration (min)</b>
3	Concrete pouring	Triangle (37.20, 55.20, 94.49)
4	Hauling	Deterministic (20)
5	Loading	Deterministic (30)
7	Return	Deterministic (25)

Table 5-23 Resources information

<b>Node</b>	<b>Resource</b>	<b>Quantity</b>
1	Trucks	5
2	Labor	10

The validation decision depends on project’s circumstances. Depending on the above results and as shown in Figure 5.7, the results of developed system are very close to the real collected productivity data, as indicated by a 1.39 % difference, in the ‘during construction stage’, and there is a relatively small difference during the planning stage at 9.00 %, compared to 13.63 % differences in other simulation systems. Therefore, the system is more precise than the available systems, which means that the simulation efficiency is better at modeling construction operations.

The simulation model is now ready to be processed through the optimization stage. The simulation model is more accurate than the other available systems because: (1) the data quality used to build this model is better; (2) the simulation engine and data mining engine integration provides a near-real project environment that gives more accurate durations instead of assumed or fitted probabilities; (3) the simulation is built based on training data sets that are real and contain all the data patterns; and (4) the validation

points, which present an interaction between the model builder and the system, improved the modeling process itself.

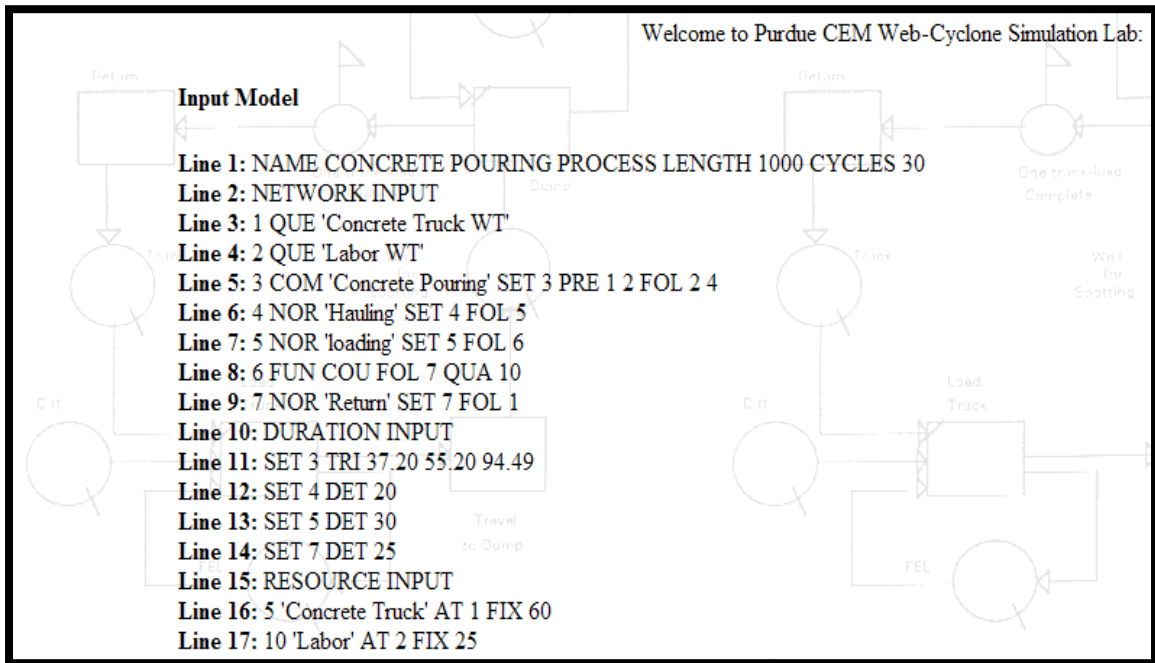


Figure 5-5 Simulation model code

#### 5.4 Optimization Stage (OS)

The first goal of this stage is to evaluate the effect of changing the variables on the existing state and on the output of the concrete pouring operation. The second goal is to rank and select the optimum solutions. This stage consists of two steps. In the first step, the sensitivity analysis at the project level is carried out on the operation to determine the effect of each variable and resource on the system output that there after present the optimum solution(s), as well as other feasible solutions. In the second step, the selection and ranking of optimal solutions is done using a rank assignment algorithm.

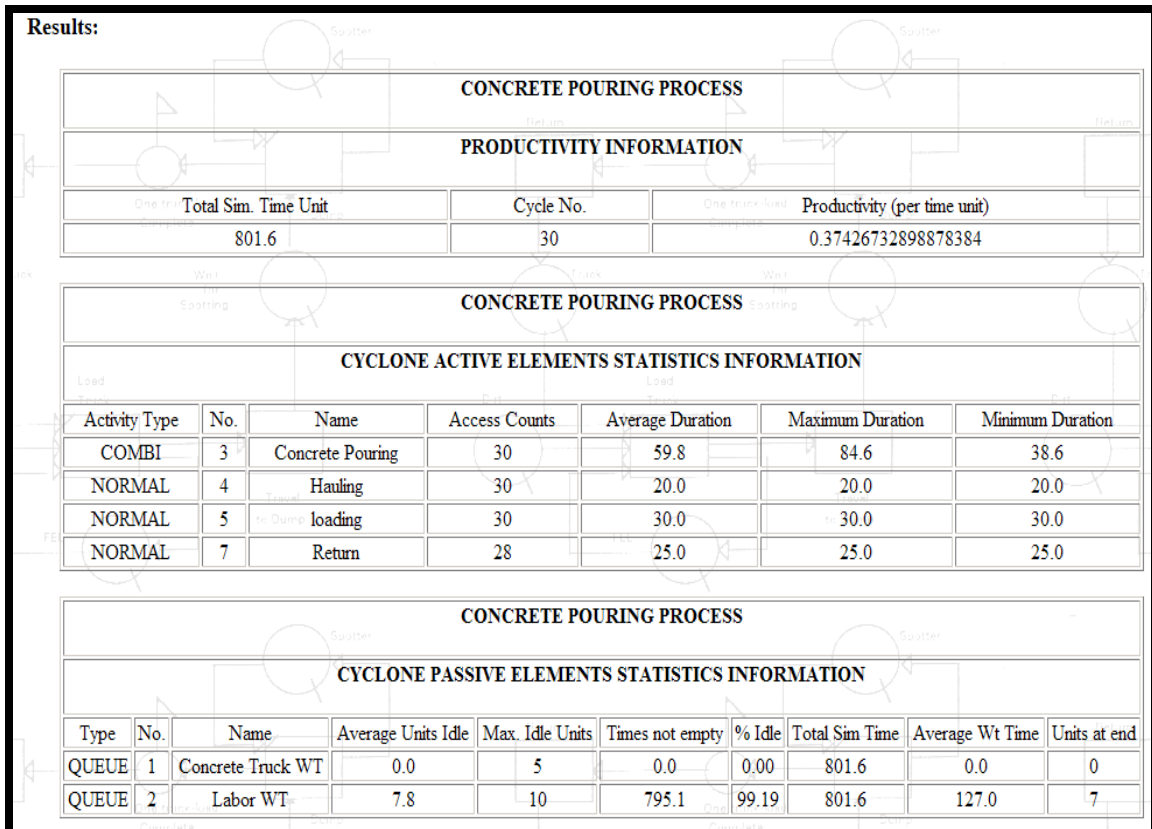


Figure 5-6 Simulation process results

#### 5.4.1 Sensitivity Analysis on Project Level Phase

In this phase, as shown in Table 5.25, the system is run under different conditions of temperature, floor level and labor percentage, changing only one variable at a time while fixing all other variables and resources. This phase indicates the effect of each variable on the feasible solutions and presents the optimum solution(s), showing the validity of the optimum solution on all of the system states. As shown in Figures 5.8 to 5.10, it is clear that these variables have an effect on productivity and on the optimal solutions. The temperature has a positive effect on productivity results in the concrete pouring process; wherein the productivity is more affected by temperature changes from -15 to 0 °C and thereafter the affect is more moderate from 0 °C and up.

Table 5-24 Sensitivity analysis for Web cyclone and Ezstrobe

Resources Information		Productivity Information					
		Web Cyclone			Ezstrobe		
LOADER	labor	Productivity Per Unit Time	Cost Per Unit Time	Cost Per Prod. Unit	Productivity Per Unit Time	Cost Per Unit Time	Cost Per Prod. Unit
1	1	0.0729	1.5046	20.6429	0.0693	1.8730	27.0269
1	2	0.0755	1.8974	25.1429	0.0794	2.6961	33.9555
1	3	0.0730	2.2541	30.8571	0.0612	2.9294	47.8656
1	4	0.0719	2.7926	38.8571	0.0544	2.6287	48.3210
1	5	0.0721	3.2401	44.9286	0.0722	4.0407	55.9658
1	6	0.0751	3.6052	48.0000	0.1532	2.9945	19.5462
1	7	0.0701	4.0024	57.0714	0.1615	3.8023	23.5435
1	8	0.0703	4.4418	63.1429	0.1401	3.5160	25.0965
1	9	0.0725	5.0201	69.2143	0.2141	6.6340	30.9855
1	10	0.0711	5.3538	75.2857	0.1414	5.2128	36.8657
1	11	0.0711	5.7859	81.3571	0.0653	2.8029	42.9235
1	12	0.0706	6.1706	87.4286	0.0834	4.6592	55.8654
1	13	0.0703	6.5720	93.5000	0.0825	5.2772	63.9658
1	14	0.0732	6.8583	93.7143	0.0725	4.4215	60.9856
1	15	0.0714	7.5430	105.6429	0.0734	5.1864	70.6589
1	16	0.0727	8.1231	111.7143	0.1528	4.2125	27.5687
1	17	0.0716	8.4355	117.7857	0.1488	4.7108	31.6589
1	18	0.0729	8.5003	116.5714	0.1421	5.0810	35.7566
1	19	0.0723	9.3957	129.9286	0.1563	6.3550	40.6587
1	20	0.0726	9.8782	136.0000	0.1402	5.8270	41.5621
2	1	0.1311	2.4867	18.9615	0.0698	3.9064	55.9658
2	2	0.1421	2.9332	20.6429	0.0866	5.3752	62.0698
2	3	0.1410	3.3394	23.6786	0.0699	5.5619	79.5687

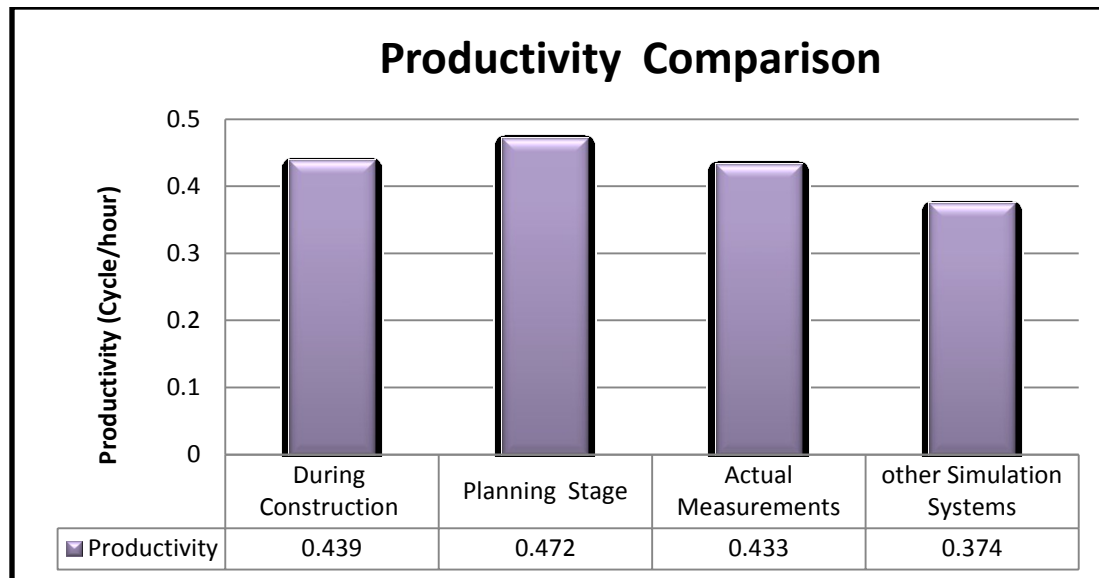


Figure 5-7 Productivity comparison between proposed system and other systems

The floor level has two phases: the first, positive phase is from floors 1-8, and from floors 8-15 there is a negative effect. It is logical that the productivity in the 1-8-floors' phase is increasing because the learning curve is higher. The second phase, for floors 8-15 will have a negative affect because the productivity will be less than that of the lower floors because of safety and handling issues. The labor percentage has no effect on productivity, because pouring concrete is not a high technology activity and so is not affected by a scarcity of skilled workers in the crew. Therefore, a sensitivity analysis on the project level has a significant role in the selection of optimum solutions.

As shown in Table 5.26, a comparison is made between the sensitivity analysis of the proposed system and that of the Web Cyclone method. The results show the significant affect of the variable changes on the simulation results. Some specific differences are obvious, for instance, the temperature variable changes the productivity of the proposed system, while this does not change the productivity with Web Cyclone.

Table 5-25 Sensitivity analysis on the project level

Variables Information			Resources Information			Productivity Information		
Temperature °C	Floor Level	Labor percentage %	Trucks	labor	loader	Productivity Per Unit Time	Cost Per Unit Time	Cost Per Production Unit
-15	6	65	3	3	1	0.4022	3.08	0.077
0	6	65	3	3	1	0.4061	3.08	0.076
25	6	65	3	3	1	0.4068	3.08	0.076
10	1	65	3	3	1	0.4124	3.08	0.75
10	8	65	3	3	1	0.4131	3.08	0.75
10	15	65	3	3	1	0.4098	3.08	0.75
10	6	55	3	3	1	0.4023	3.08	0.77
10	6	60	3	3	1	0.4023	3.08	0.77
10	6	70	3	3	1	0.4023	3.08	0.77
10	6	65	1	1	1	0.405	1.58	3.90
10	6	65	2	1	1	0.402	2.08	5.17
10	6	65	3	1	1	0.301	2.58	8.57
10	6	65	1	1	1	0.405	1.58	3.90
10	6	65	1	2	1	0.405	1.83	4.52
10	6	65	1	3	1	0.405	2.08	5.14
10	6	65	1	1	1	0.405	1.58	3.90
10	6	65	1	1	2	0.405	2.42	5.98
10	6	65	1	1	3	0.405	3.25	8.02



Table 5-26 Comparison between the sensitivity analysis of the proposed system and that of web cyclone

Variables Information			Resources Information			Productivity Per unit Time	
Temperature °C	Floor Level	Labor percentage %	Trucks	labor	loader	The Proposed System Cycles/hour	Web Cyclone Cycles/hour
-15	6	65	3	3	1	0.4022	0.4350
0	6	65	3	3	1	0.4061	0.4350
25	6	65	3	3	1	0.4068	0.4350
10	1	65	3	3	1	0.4124	0.4350
10	8	65	3	3	1	0.4131	0.4350
10	15	65	3	3	1	0.4098	0.4350
10	6	55	3	3	1	0.4023	0.4350
10	6	60	3	3	1	0.4023	0.4350
10	6	70	3	3	1	0.4023	0.4350
10	6	65	1	1	1	0.4050	0.4230
10	6	65	2	1	1	0.4020	0.4330
10	6	65	3	1	1	0.3010	0.4340
10	6	65	1	1	1	0.4050	0.4230
10	6	65	1	2	1	0.4160	0.4335
10	6	65	1	3	1	0.4351	0.4344
10	6	65	1	1	1	0.4050	0.4230
10	6	65	1	1	2	0.4050	0.4230
10	6	65	1	1	3	0.4050	0.4230

This means that the decision maker using Web Cyclone is making decisions while neglecting the effects of some variables that do affect the process. The changes in the results are more or less comparable to those of Web Cyclone, as the resource distribution is the same. So, the results here as well as those shown above are more accurate and give the decision maker the ability to study the project under different conditions.

#### **5.4.2 Selection and Ranking of the Optimal Solutions Phase**

In this phase, the optimal solution(s) is/are selected from the candidate solutions shown in Table 5.27, which contains all of the feasible solutions. The ideal case is to find one solution (the dominated solution), but in most cases a set of solutions are presented (non-dominated – Pareto optimal). To model the optimal solution model, there will be two objective functions. The first objective function (f1) maximizes productivity and the second objective function (f2) minimizes cost. To select the dominated solution, non-dominated solutions are removed first. To find the non-dominated solution, a comparison between all feasible solution points is made. This comparison is presented by a pair of symbols, each of which can take three values, +, -, or =, according whether a solution is better than, worse than or equal to the others. As shown in Table 5.28, non-dominated solutions are extracted as follows:

**Consider point A:** Is not-dominated in intersection (A-B), (A-C), and (A-D). For (A-B) the cost of A is more than B, and the productivity of A is greater than the productivity of B. For (A-C) the cost of A is more than C, and the productivity of A is greater than the productivity of C. For (A-D) the cost of A is more than D, and the productivity of A is greater than the productivity of D. So, point A will continue in the competition due to its higher productivity.

**Consider point B:** Is not-dominated in intersection (B-A), (B-C), and (B-D). For (B-A) the cost of B is less than B, and the productivity of B is less than the productivity of A. For (B-C) the cost of B is less than C, and the productivity of B is more than the productivity of C. For (B-D) the cost of B is less than D, and the productivity of B is less than the productivity of D. This means the point C loses against point B and it will be removed from the selection pool.

Rank one will be assigned to the first group of non-dominated solutions and then temporarily removed from the population to complete the comparison process. The second rank of classification is shown in Table 5.29. None of the variables were dominated for a classification of solutions rank 3. The process follows:

**Consider point A:** Is not dominated in intersection (A-C) because the cost of A is more than C, and the productivity of A is greater than the productivity of C. The point loses because it is dominated by point C, which was removed in the previous ranking step.

**Consider point D:** Is dominated in intersection (D-C) because point C was already removed from the competition.

Rank two is assigned to the first group of non-dominated solutions and they are then temporarily removed from the population to complete the comparison process. The third rank of classification is shown in Table 5.30; the table is empty of non-dominated solutions because none were found. Therefore, the process will be stopped because all non-dominated solutions were removed and the points B and D are selected as the optimum solutions.

Table 5-27 Candidate solutions

Point	Productivity (f1) Cycle/hour	Cost (f2) (\$/minute)
A	0.4098	3.08
B	0.4050	1.58
C	0.3010	2.58
D	0.4060	1.83

Table 5-28 Candidate solutions comparison (Rank 1)

	A	B	C	D
A		(+,-)	(+,-)	(+,-)
B	(-,+)		(+,-)	(-,+)
C	(-,+)	(-,-)		(-,-)
D	(-,+)	(+,-)	(+,+)	

Table 5-29 Classification of solutions (Rank 2)

	A	B	D
C	(-,+)	(-,-)	(-,-)

Table 5-30 Classification of solutions (Rank 3)

	B	D
C	(-,-)	(-,-)

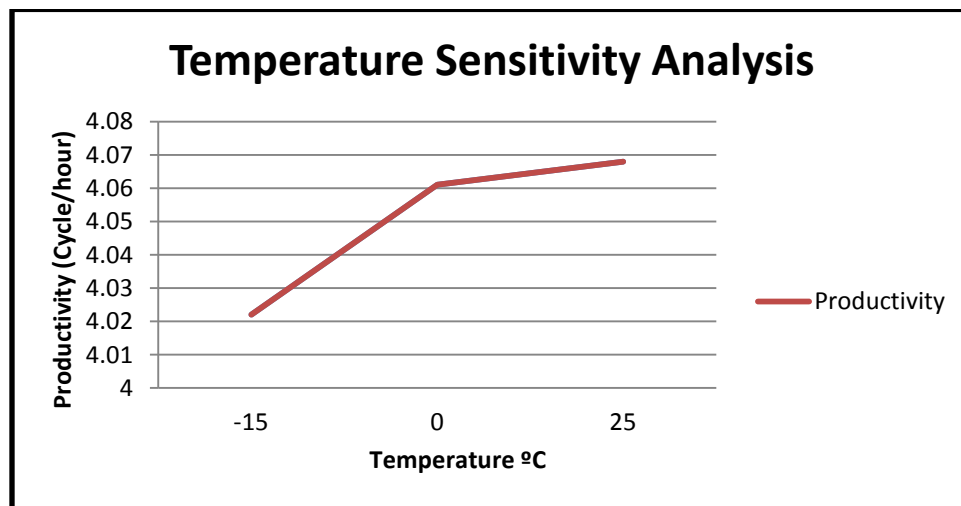


Figure 5-8 Temperature sensitivity analysis

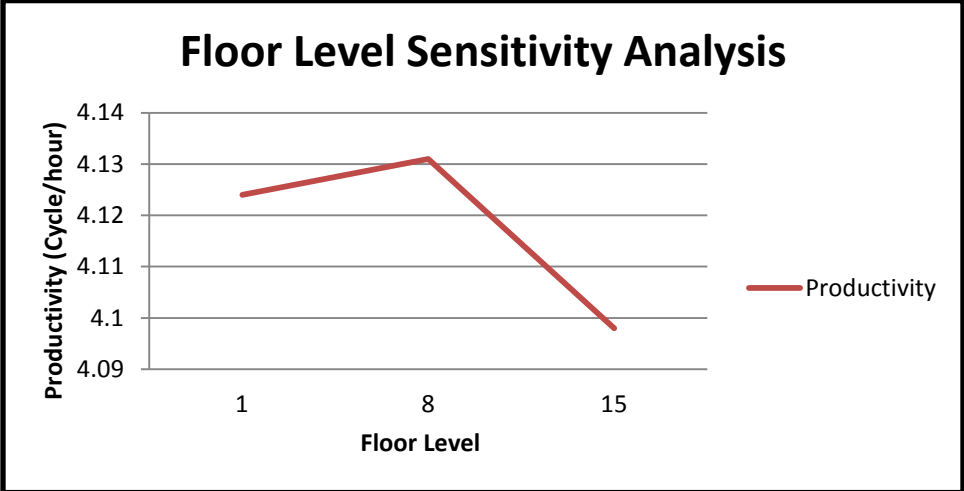


Figure 5-9 Floor level sensitivity analysis

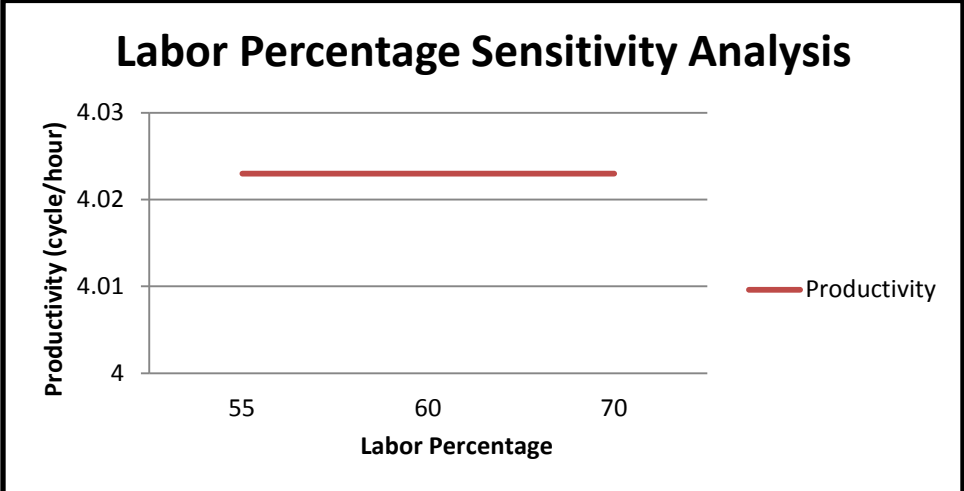


Figure 5-10 Labor percentage sensitivity analysis

# Chapter 6: KEYSTONE Language and System Automation

## 6.1 Chapter Overview

As shown in Figure 6.1, this chapter consists of six sections that provide a detailed explanation of the KEYSTONE language development and system automation. It begins with the KEYSTONE language's core, which shows the components of the developed language and how it interacts with the construction operation simulation and modeling. The second section presents the selection criteria for this programming tool. The third section details the system's architecture, which consists of four models: training model, simulation model, reporting model and graphical user interface (GUI). The fourth section shows the data flow of the developed system. The fifth section is the (GUI), which shows the interaction between the user and the developed system. The final section presents a complete case study to show the procedure of applying the implemented software on a complete construction operation.

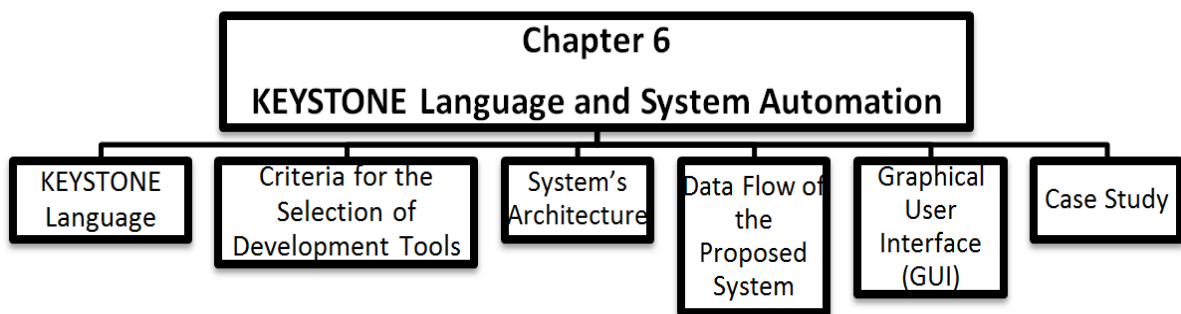


Figure 6-1 KEYSTONE language and system automation chapter's organization chart

## 6.2 KEYSTONE Language's Core

This section describes KEYSTONE, a new construction simulation language that allows user to model construction operations with unique ease. KEYSTONE is the acronym for

a Knowledge discovery based construction simulation system. The KEYSTONE language includes most of the features that are desirable in a general purpose simulation language. KEYSTONE is designed to write and generate object-oriented simulation code. Construction operations are automatically translated into C# after defining by user. Graphical user interfaces are automatically generated for various operating systems. The sequential instructions written in this language allow the computer to create a platform of the construction model that is accurate in time, space, and appearance; and which shows the interaction between the project, operations, processes, servers, and auxiliaries.

This section presents the building and modeling features of the developed language. As shown in Figure 6.2, the language's core consists of four classes. These classes present and contain all of the data and the model's information. The first class is the project, which represents the second level of construction management, for example, building an administration building. The second class is an operation level that represents the third level of construction management, for example, planning that administration building's foundations. The third class is a process level that represents the fourth level of construction management, for example, pouring the concrete for foundations. The fourth class is an auxiliary that represents the moving objects that are shared between all of model's processes. Each class consists of three objects: fields, properties, and methods. Fields contains all of the data types in the project. Properties contain all of the assigned values for all of the models. Methods encompass all of the processing that can occur on all types of data in the system.

As shown in Figure 6.3, the project consists of operations, processes, servers and their related components. The operations and servers are in the same level so that the servers

can be sheared between more than one operation. The interaction between all of the components is carried out by the moving units or auxiliaries that represent the simulation model. The KEYSTONE language deals with the model as classes, not as components, which gives it the ability to model and to extend the simulation model to be more than merely for an operation and more even, than a single project.

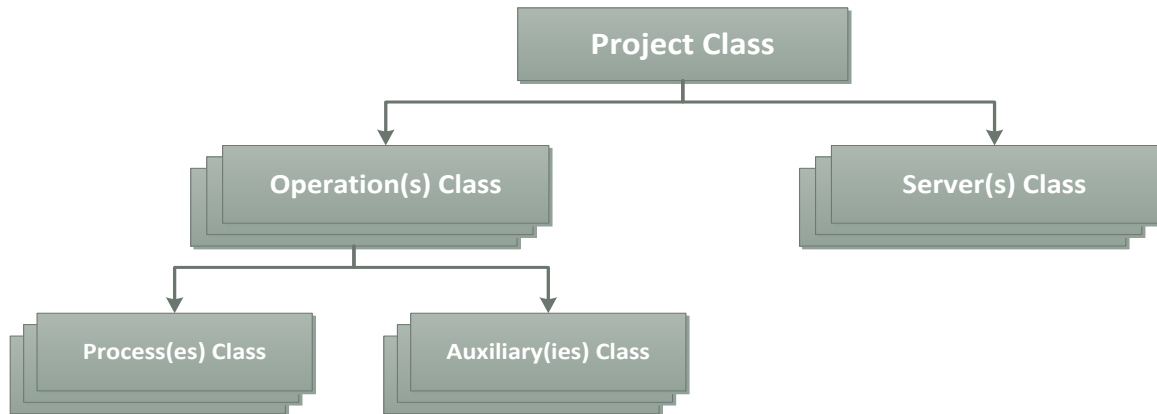


Figure 6-2 KEYSTONE core language framework

### 6.2.1 Project Class

Project classes contain and present information pertaining to data and models. As shown in Figure 6.4, project class consists of **12** fields, **11** properties and **17** methods. Fields are used to store and exhibit assigned data in the project class, for example: duration log, simulation log, and productivity log. Properties data is either entered by model builder or calculated by the system. Project class takes the following form:

```
Class Project Class
```

```
{
```

```
// Fields, Properties and methods go here
```

```
}
```



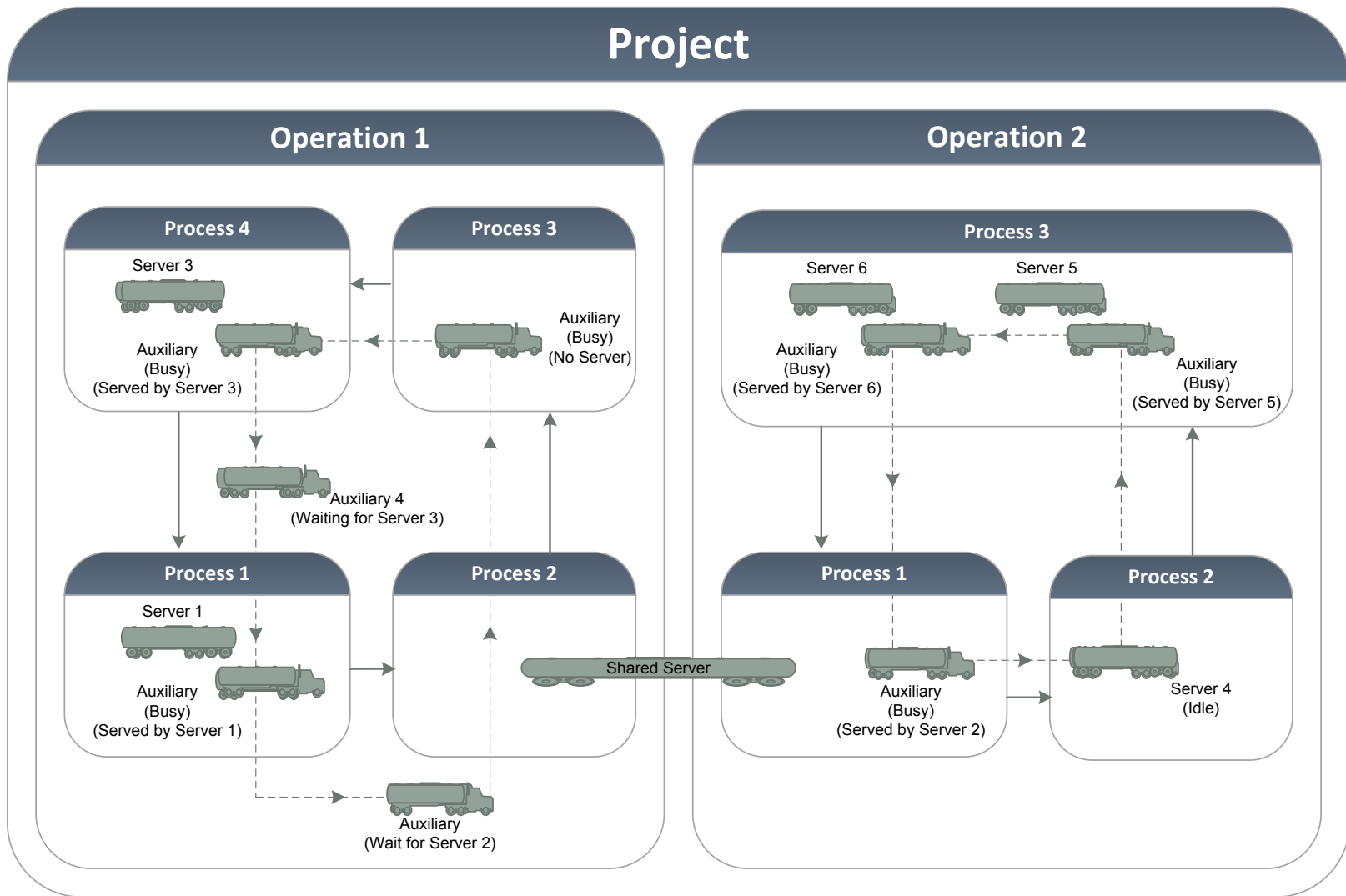


Figure 6-3 System component interaction

### 6.2.1.1 Methods

Methods fully represent processing operation on the entire array of data in the system.

There are 17 methods as enumerated below:

#### 1- Add Operation Method

This method is used to create all project operations, for example: pouring concrete, loading and etc. It takes the following form:

```
public void Add Operation(COperation Operation)
{
    _Operations.Add(Operation);
}
```

#### 2- Add Server Method

This method is used to add a server to the system, which is assigned later on to the processes, for example: adding loader or pouring pump. It takes the following form:

```
public void Add Server(CServer Type Server)
{
    Server Types.Add(Server);
}
```

#### 3- Clone Method

This method is used to provide a copy of all objects in the system, which is used as a backup copy of the entire project information. It takes the following form:

```
public object Clone()
{
    return this.Member wise Clone();
}
```

#### 4- Project Method

This method is used to generate a copy of resource distribution assigned to all processes in each operation to re-run the sensitivity analysis and ranking at every cycle. It takes the following form:

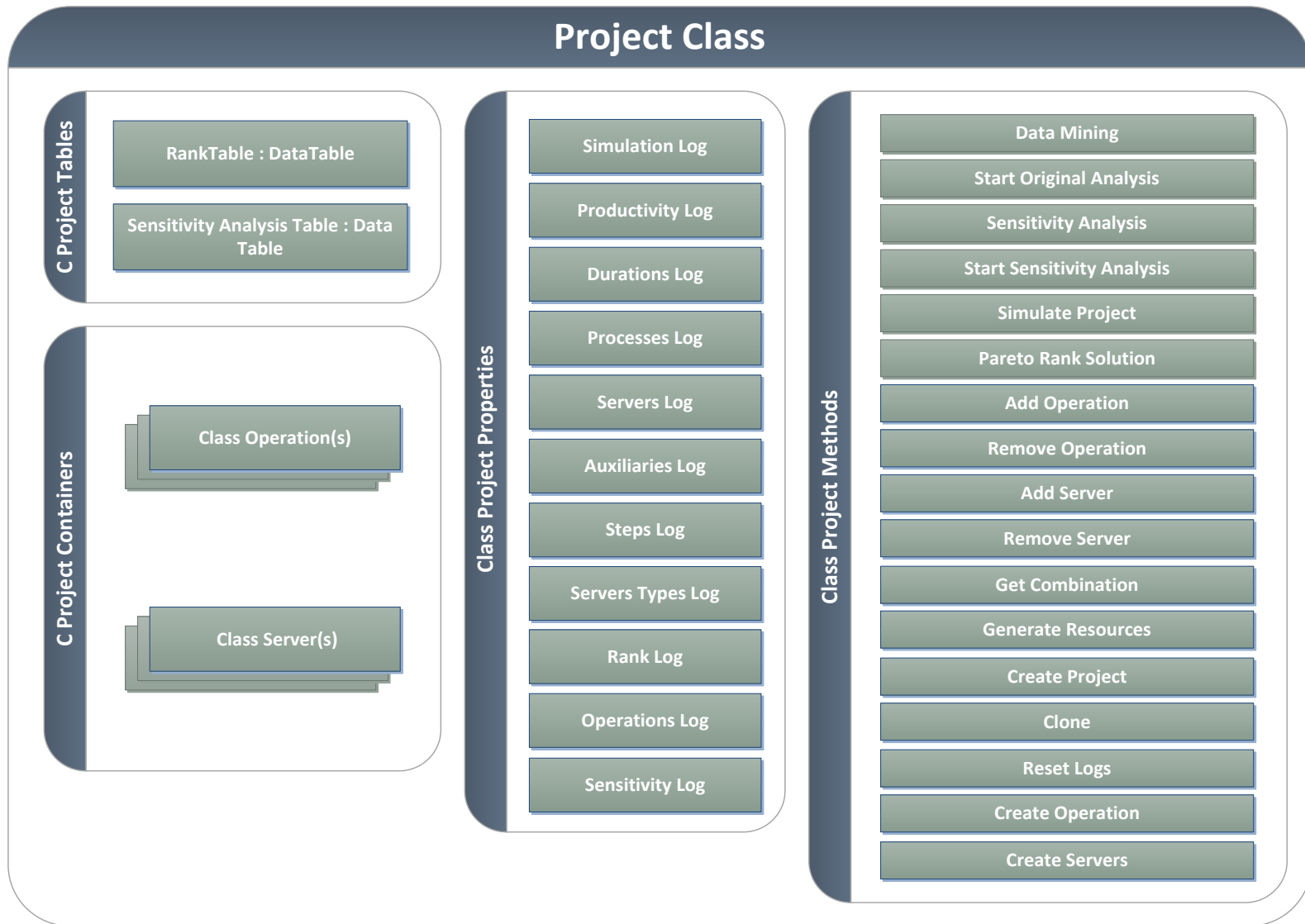


Figure 6-4 Project class components

```

public CProject()
{
    Main Sensitivity Analysis Table = new Data Table();
    Rank Table = new Data Table();
}

```

### 5- Create Operation Method

This method is used to create a database for all operations, assign names and avoid repeating the same operation name. It takes the following form:

```

public void Create Operation(string Name)
{
    bool Operation Exists = false;

    for each (COperation o in this.Operations)
        if (o.Name == Name)
        {
            Operation Exists = true;
            break;
        }
    if (Name != null && Name != "" && !Operation Exists)

        COperation op = new COperation();
        op.Name = Name;

        Operations.Add(op);
    }

    if (Operation Exists)
        System.Windows.Forms.MessageBox.Show("Operation Already Exists.
Nothing will be created.");
}

```

### 6- Create Server Method

This method is used to create all servers, assign names, and avoid repeating the same server name. The server is defined by name, cost, and original number. Also, sensitivity limits are defined in this method. It takes the following form:

```

public void Create Server(string Name, double Cost, int Original, int Minimum, int
Maximum)
{
    bool Server Exists = false;
}

```

```

foreach (CServer Type server in _ServerTypes)
    if (server.Name == Name)
    {
        Server Exists = true;
        break;
    }

if (Name != null && Name != "" && !Server Exists)
{
// Create Server Information
CServer Type s = new CServer Type();
s.Name = Name;
s.Cost = Cost;
s.Original Number Of nstances = Original;
s.Minimum Number Of Instances = Minimum;
s.Maximum Number Of Instances = Maximum;
// Message for Existing Name
if (Server Exists)
    System.Windows.Forms.Message Box. Show("Server Already Exists. Nothing
will be created.");
}

```

## 7- Data Mining Method

This method is used to create clusters and clean data. Data cleaning step attempts to fill in missing values, smooth out noise while identifying outliers, and rectify inconsistencies in the data. Fuzzy K-means Clustering is used to fill the missing data attributes using the predefined characteristic of the fuzzy engine. This method is repeated depending on the number of cycles that are applied in the sensitivity analysis. It takes the following form:

```

private static void Data Mining(CProject Current Project, int Number of Cycles,
CFKM_Settings FuzzySettings)
{
    foreach (COperation oper Current Project.Operations)
        foreach (CProcess proc in oper.Processes)
        {
            proc.Set Number of Cycles(Number of Cycles);

            for (int i = 0; i < Number of Cycles; ++i)
// Starting the Mining Process
proc.Start Process Data Mining (FuzzySettings);
}
}

```

## 8- Generate Resources Method

This method is used to generate combination of resources for sensitivity analysis. These combinations present feasible solutions that are generated by changing in one resource while fixing all other resources. It takes the following form:

```
public void Generate Resources(List<int> Combination Pattern, bool For Sensitivity Analysis)
{
    if (Sensitivity Analysis)
    {
        int Pattern Index = 0;

        // Define Auxiliary Types in each operation First
        foreach (COperation oper in this.Operations)
            foreach (CAuxiliary Type Aux in oper.Auxiliary Types)
        // Define Auxiliary Types in each operation First Analysis
            foreach (CServerType Ser in this.ServerTypes)
                Ser.Generate Instances(Combination Pattern[PatternIndex++]);
    }
    else
    {
        // Populate Auxiliary Types in each operation First
        foreach (COperation oper in this.Operations)
            foreach (CAuxiliaryType Aux in oper.AuxiliaryTypes)
                Generate Instances(Aux.Original Number Of Instances);

        // Populate AuxiliaryTypes in each operation FirstAnaly
        foreach (CServerType Ser in this.ServerTypes)
            Ser.Generate Instances(Ser.Original Number Of Instances);
    }
}
```

## 9- Get Combination Method

This method is used to generate the resources matrix for sensitivity analysis. This matrix is formed depending on the resources combination. It takes the following form:

```
private static List<string> GetCombination(int a, List<Array> x)
{
    List<string> retval = new List<string>();

    if (a == x.Count)
    {
```

```

        retval.Add("");
        return retval;
    }
    foreach (Object y in x[a])
    {
        foreach (string x2 in Get Combination(a + 1, x))
        {
            retval.Add(y.ToString() + "," + x2.To String());
        }
    }
    return retval;
}

```

### 10- Pareto Ranking Solution Method

This method is used to utilize a ranking system to assign equal probability of reproduction of the best individuals in the population. Rank one is assigned to the first group of non dominated solutions and temporarily removed from the population. Then the next group of non-dominated individuals is assigned rank two, and so on. The rank assigned to each individual represents its fitness level. It takes the following form:

```

private List<CSensitivity Pair> Pareto_Rank_Solutions()
    int Current Rank = 1;
    int N = _Productivity Vs. Cost Pairs.Count;

// Store Original Indices
    for (int i = 0; i < Productivity Vs. Cost Pairs.Count; i++)
        Productivity Vs. Cost Pairs[i].OriginalIndex = i;

    List<CSensitivity Pair> Temp = new List<CSensitivity Pair>(Productivity Vs. Cost
Pairs.Count);

    foreach (CSensitivity Pair item in Productivity Vs. Cost Pairs)
        Temp.Add(item.Deep Clone());

    List<CSensitivity Pair> Selected Set = new List<CSensitivity Pair>();

    while (N != 0)
        for (int i = 0; i < Temp.Count; ++i)
            if (Temp [i] .Is Non Dominated (Temp))
                Temp [i]. Rank = Current Rank;

```

```

bool Is Pair Removed = false;

for (int i = 0; i < Temp.Count; ++i)
    if (Temp[i].Rank == CurrentRank)
        Selected Set.Add(Temp[i]);
        Temp.Remove (Temp[i]);
        Is Pair Removed = true;
        i--;
        N--;
if(Is Pair Removed)
    Current Rank++;
if (Temp.Count == 1)
    Temp [0]. Rank = Current Rank;
    Selected Set.Add(Temp[0]);
    break;
return SelectedSet;

```

## 11- Sensitivity Analysis Method

This method is used to combine all related methods that were previously discussed. It also checks on all requirements of sensitivity analysis, prepares tables for ranking method, and reports this progress for the reporting method. This method checks on processes, servers, and auxiliaries. It takes the following form:

```

public bool Prepare For Sensitivity Analysis()

// Report Progress
This.Report Progress(4, "Starte the Sensitivity Analysis");
This.Report Progress(4, "Checking if Processes need training");
// Check if any Process still needs Training
// Check for Auxiliaries
This BGW.Report Progress(5, "Checking if Processes need training");
    foreach (COperation oper in this. Operations)
        if (oper.Auxiliary Types.Count == 0)
            {
                System.Windows.Forms.Message Box.Show("Operation : " + "\"" +
oper.Name + "\" has NO Auxiliaries !", "No Auxiliaries Found"),
System.Windows.Forms.Message Box Buttons.OK, System.Windows.Forms.Message
Box Icon.Error);
                return false;
            }
// Warn if there are no Servers
// Calculate the Productivity

```



```

Main Sensitivity Analysis Table = this.Start Sensitivity Analysis (Number Of Cycles,
Fuzzy Settings, ref bgw);
// Copy Table
    Rank Table = MainSensitivity Analysis Table.Copy();
    Rank Table.Rows.Clear();
    Rank Table.Accept Changes();

// Create Rank Table
    bgw.Report Progress (7, "Creating Rank Table");

// Find Best Solution
    List <CSensitivity Pair> Sorted List = Pareto_Rank_Solutions();
// List in Rank Table
    for (int i = 0; i < SortedList.Count; i++)
    {
        DataRow r =
        _MainSensitivityAnalysisTable.Rows[SortedList[i].OriginalIndex];
//Rank Table.NewRow();
        Rank Table.Import Row(r);
        Rank Table.Accept Changes();
// Report Progress
    return true;

```

## 12- Remove Operation Method

This method is used to remove any defined operation without a trace or affect on the modeling and simulation processes. It takes the following form:

```

public void Remove Operation(C Operation Operation)
{
    Operations.Remove (Operation);
}

```

## 13- Remove Server Method

This method is used to remove any server without any traces and affect on modeling and simulation processes. It takes the following form:

```

public void RemoveServer(CServer Type Server)
{
    Server Types. Remove (Server);
}

```

## 14- Reset Logs Method

This method is used to reset and erase project's logs. It erases simulation log, productivity log, steps' log, durations log, auxiliaries' log, servers' log and processes' log. The explanation behind this method is to start another model and reset the memory in order to ease the use of system. It takes the following form:

```
public void ResetLogs()
{
    Simulation Log = "";
    Productivity Log = "";
    Steps Log = "";
    Durations Log = "";
    Auxiliaries Log = "";
    Servers Log = "";
    Processes Log = "";

    return true;
}
```

### 15- Simulate Project Method

This method models the movement of units in the project. The objective of developing this stage is to model the interaction between flow units and the idle times of the resources to flag the bottlenecks and estimate the productivity of the proposed system. This phase is the main phase in the project class that controls most of output and affects most of outputs in the project. It takes the following form:

```
public static void Simulate Project
{
    Current Project.Reset Logs();

    if (Log)
    {
        // Auxiliaries Moving in the Model
        // Check if All Auxiliaries Finished
        // Reset Server
        // Check Sequence , Operation and Process
        // Begin Original Simulation with Selected Auxiliaries and Server Types
        // Reset All Auxiliary Types at the beginning of each operation
        // Not Working and Waiting for Current Process
    }
}
```

```

// Begin Serving
// Wait for Next Process to be Free
// If currently Working
// Continue Working if Current Process is Still Busy
// Finishing Current Process to Proceed to Another one
// For each Auxiliary
CurrentProject._SimulationLog += Environment.NewLine + "At Time : " + Current
Time.To String("N1")
+ " Auxiliary \"\" + CurrentAuxiliary.Name + "\"\" + " has FINISHED its Job" +
Environment.NewLine;

// For Current Process, Calculate Waiting Time for all Attached Server Types
if (CurrentProcess.AttachedServer != null)
foreach (CServerInstance server in Current Process.Attached Server. Instances)
{
if (!server.IsBusy())
{
server.Wait(Time Increment);
}
}

// Increment Time
CurrentTime += Time Increment;

// Final Check for finishing
// Auxiliaries
// After All Auxiliaries Entered Last Process, Calculate Cycle Time
// Productivity Calculations for all Auxiliaries)
double Productivity Per Auxiliary = (Aux. Average Waiting Time() + Aux.Average
WorkingTime()) / Aux Type.Instances.Count; Sum += Productivity Per Auxiliary;Cost
Per Unit Time += AuxType.Cost / 60;
// Processes
// Calculate Average waiting time at each process
foreach (CProcess proc in Current Operation.Processes)
Current Project._Simulation Log += "Average Waiting Time
for process : \"\" + proc.Name + "\" is " + proc. Average Waiting Time().To String("N1")
+ " \" + Environment.NewLine + Environment.NewLine;
// Servers
foreach (CServerType ServerType in Current Project._Server Types)
foreach (CServer Instance server in Server Type. Instances)
// Processes

```

### 16- Start Original Analysis Method

This method is used to start the original simulation model. It checks the data mining process to estimate process durations. This method implements certain checks before

starting the original analysis. Those checks include auxiliary and server types in each operation. It creates only one combination of auxiliaries and servers for the original model. It takes the following form:

```

public bool Start Original Analysis(int Number Of Cycles, CFKM_Settings Fuzzy
Settings)
{
// Data Mining to Estimate Process Durations
CProject.Data Mining. Project (this, Number of Cycles, Fuzzy Settings);
// Reset SimulationLog
// Populate Auxiliary Types in each Operation First
// Populate ServerTypes in each Operation First
// Create Combination (One Combination in Case of Original Simulation)
    foreach (string x in Get Combination (0, my List))
    {

        string[] split = x.Split(new Char[] { ',' });
        List<int> Combinations Pattern = new List<int>();

        foreach (string s in split)
            if (s.Trim () != "")
            {
                int current;
                int.TryParse(s, out current);

                Combinations Pattern .Add (current);
            }
// Generate Resources based on Current Pattern
this.Generate Resources (Combinations Pattern, true);
// Run Simulation
        double Operation Productivity = 0;
        double Cost Per Unit Time = 0;
        CProject.Simulate Project(this, Number of Cycles, true, Fuzzy Settings, out
Operation Productivity, out Cost Per Unit Time);
    }
}

```

### 17- Start Sensitivity Analysis Method

This method is used to start the sensitivity analysis. It generates sensitivity analysis table columns that include adding columns for each auxiliary, server, results, productivity, and column of productivity. This method builds a framework for the table to be filled by sensitivity analysis process. It takes the following form:

```

private Data Table Start Sensitivity Analysis ( )

// Data Mining to Estimate Process Durations
    CProject.Data Mining Project (this, Number of Cycles, Fuzzy Settings);

    # region Prepare Sensitivity Analysis Table
// Generate Sensitivity Table Columns
// Add Columns for each Auxiliary
// Add columns for Each Server
// Add Results columns
// Populate Auxiliary Types in each operation
// Clear Previous Pairs
// Recursively Create Combinations , Fill the Tables and Run simulation
// Fill in Table
// Fill The Table Row
// Make a copy of the project
// Generate Resources based on Current Pattern
// Run Simulation
// Write Down Simulation Results in Sensitivity Table
// Add Productivity and Cost Pairs
// Finally Return DataTable

```

## 6.2.2 Operation Class

Operation Class contains and presents all operations relate data such as auxiliary types, names, processes, and cycles. As shown in Figure 6.5, Operation class consists of 4 fields, 3 properties, and 9 methods. Project class takes the following form:

Class operation Class

```

{
// Fields, Properties and methods go here
}

```

### 6.2.2.1 Methods

Methods represent any processing that has occurred on all types of system data in that relates to the operations. There are 9 methods. The methods are as following:

#### 1- Add Process Method

This method is used to create all operation processes such as hauling, loading and etc. this method also checks for duplicate names and null names. It takes the following form:

```

        public void Append Process (ref CProcess Process)
    {
        bool Is Already there = false;

        foreach (CProcess proc in _Processes)
            if (proc. Name == Process.Name && proc. Name != null)
            {
                Is Already there = true;
                break;
            }
        if (!Is Already there Processes.Add(Process));
        else
            System.Windows.Forms.MessageBox.Show: ("The Process \"\" +
        Process.Name + "\" already Exists !");
    }

```

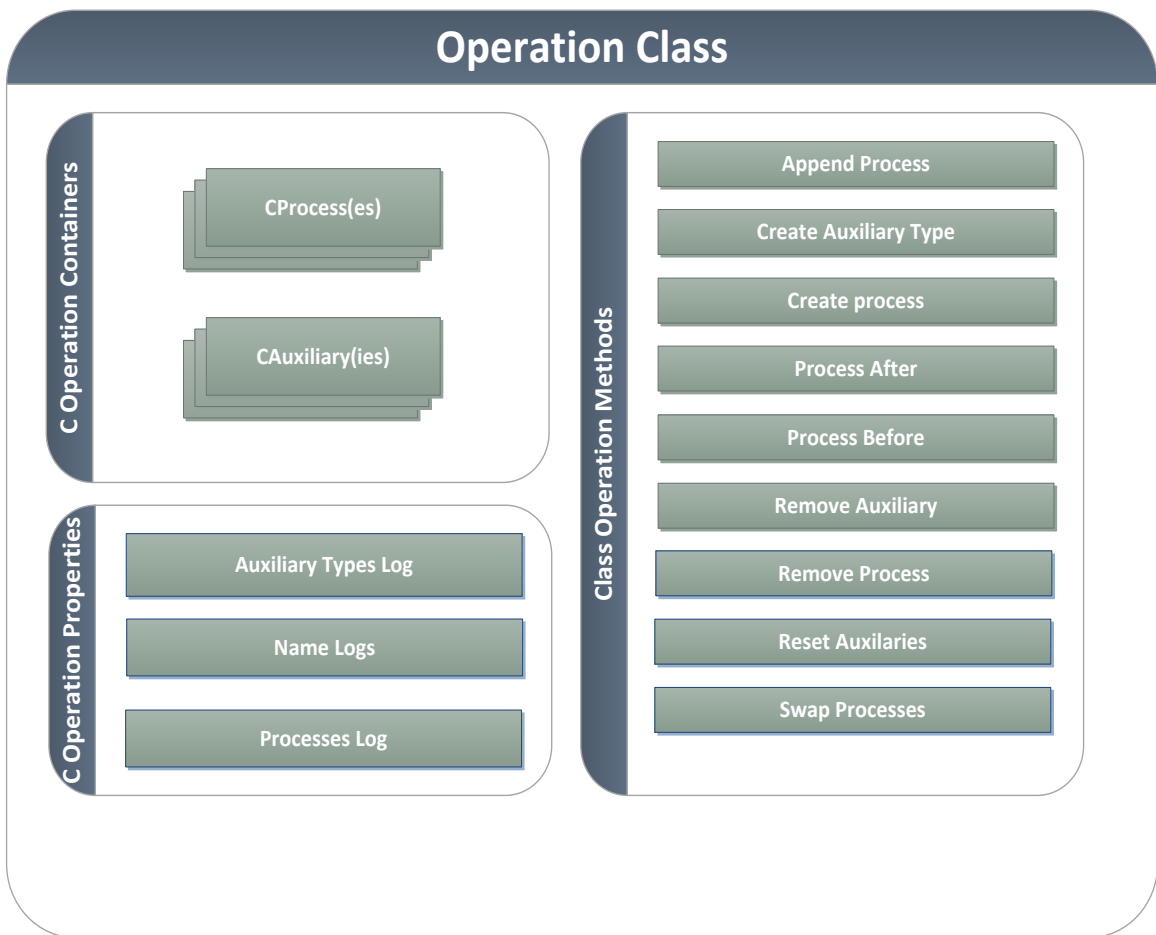


Figure 6-5 Operation class components

## 2- Create Auxiliary Types Method

This method is used to create all auxiliaries like trucks. This method also checks the duplicate name, and null names. This method assigns different types of auxiliaries depending on the name, e.g. there is a difference between Truck\_1 and truck\_2 in everything including capacity. It takes the following form:

```
public bool Create Auxiliary Type ()
{
    foreach (CAuxiliary Type Aux in this. Auxiliary Types)
    {
        if (Aux.Name == Name)
            System. Windows. Forms. Message Box. Show: ("An Auxiliary Type Having the Same
            Name Already Exists !");
        return false;
    }
}
```

## 3- Create Predefined Process Method

This method is used to create templates from any defined process. It takes an image from any given process to be used in the future models. The image includes all related variables and definitions. It takes the following form:

```
public void Create Pre defined Process(string Process Name)
{
    if (Process Name == "")
    {
        CProcess Current = new CProcess(this);

        Current.Name = "";

        for (int i = 0; i < 8; ++i)
        {
            CProcess Parameter Param = new CProcess Parameter (ref Current);
            Current. Append Process Parameter (Param, true);
        }
        // Add to Processes
        Append Process(ref Current);
    }

    # endregion

    # region Hauling
```

```

if (Process Name == "")
{
    CProcess Current = new CProcess(this);

    Current.Name = "";

    for (int i = 0; i < 6; ++i)
    {
        CProcess Parameter Param = new CProcess Parameter(ref Current);
        Current. Process Parameters. Add (Param);
    }
}

```

#### 4- Process After/Before Method

Those two methods are used to locate the process in the sequence. It mimics the graphical representation to be sequence. It takes the following form:

```

public CProcess Process After(CProcess This Process, bool Cyclic)
{
    int index = Processes.Index of (This Process);

    if (index < Processes. Count - 1)
        return Processes [index + 1];
// Return Next Process in List
    else
    {
        if (Cyclic)
            return Processes[0];
// Return First Process if "This Process" is the Last one
    }
    else
        return This Process;
}

```

#### 5- Remove Auxiliary Method

This method is used to remove any assigned auxiliary without affecting the original model. It takes the following form:

```

public void Remove Auxiliary (ref CAuxiliary Type Auxiliary)
{
    bool Is Already there = false;
    if (Auxiliary != null)
        foreach (CAuxiliary Type aux in Auxiliary Types)

```



```

if(Auxiliary.Name == aux.Name && aux.Name != null)
{
    Is Already there = true;
    break;
}

```

### 6- Remove Process Method

This method is used to remove any defined process without any affect on the original model. It takes the following form:

```

public void Remove Process (ref CProcess Process)
{
    bool Is Already there = false;

    if( Process != null )
    foreach (CProcess proc in Processes)
    if (proc. Name == Process. Name && proc. Name != null)
    {
        Is Already there = true;
        break;
    }
}

```

### 7- Reset Auxiliary Method

This method is used to reset the auxiliary information log to start another cycle. It facilitates the handling of memory with the development platform. It takes the following form:

```

public void Reset Auxilaries (int Number of Cycles)
{
    this. Number of Cycles = Number of Cycles;

    foreach (CAuxiliaryType Aux Type in this.Auxiliary Types)
    foreach (CAuxiliary Instance Aux in Aux Type. Instances)
    {
        Aux.Reset (Number of Cycles);
        Aux.Current Process = null;
    }
}

```

### 8- Swap Process Method

This method is used to swap processes to change the system consideration for each process. It directs the operation to control related processes. The control defines the

manner in which auxiliaries and servers deal with each process. It takes the following form:

```
public bool Swap Processes ( )

if (proc1 == proc2)
    return false;

if (proc1 == null || proc2 == null)
    return false;

CProcess First Process, Second Process;

if (Processes. Index of (proc1) < Processes. Index of (proc2))
{
    First Process = proc1;
    Second Process = proc2;
}
else
{
    First Process = proc2;
    Second Process = proc1;
}

int First Index = Processes. Index of (First Process);
int Second Index = Processes. Index of (Second Process);

return true;
```

### 6.2.3 Process Class

Process Class presents and contains all Process related data such as validation process, training, simulation type (planning stage or construction stage), and parameter information. As shown in Figure 6.6, process class consists of **29** fields, **16** properties, and **13** methods. Project class takes the following form:

Class Process Class

```
{

// Fields, Properties and methods go here

}
```

### 6.2.3.1 Methods

Methods represent any processing happen on all types of data in the system that related to processes. There are 13 methods. The methods are as following:

#### 1- Create Parameter Method

This method is used to create parameters to the related process, for example: temperature, humidity, etc. this method will also check for duplicate names and null names. This method adds the proposed parameters to the table of parameters and durations. It takes the following form:

```
public bool Append Process Parameter ()
{
    if (Parameter == null)

        return false;

// Add Paramter to Main Training Table
// Add Paramter to Lower Bound Training Table
// Add Paramter to Upper Bound Training Table
// Add Paramter to History Data Table
```

#### 2- Attach Server Method

This method is used to attach predefined server or servers to a specific process such as attaching a loader for loading process. Attach Server Method assigns different servers to the same process and the same server to different processes. It takes the following form:

```
public void Attach Server (CServer Type Server)
{
    Attached Server Type = Server
```

#### 3- Average Waiting Time Method

This method is used to calculate the average waiting time for each process. The average waiting time for the process signals that the process is not busy with any work. It also

indicates that all servers in this process are idle. It calculates the waiting time from the status of the process. It takes the following form:

```
public double Average Waiting Time()
{
    int Total Auxiliaries = 0;
    foreach (CAuxiliary Type aux in My Operation. Auxiliary Types)
        Total Auxiliaries += aux. Original Number of Instances;
    double AWT = Total Waiting Time / (this.My Operation. Number Of Cycles *
TotalAuxiliaries);

    return AWT;
```

#### 4- Serving Auxiliary Method

This method is used to command the servers to start serving the auxiliaries. This method also measures the time before and after the serving starts. It also checks the integration and shearing of server(s) with other processes and operations. It takes the following form:

```
public string Begin Serving Auxiliary()
{
    string Log String = "";

    if (!Auxiliary To Be Served. Entered Last Process())
    {

// Fill Auxiliary Properties
        Auxiliary To Be Served. Is Waiting = false;
// Working
        Auxiliary To Be Served. Current Process = this;
        Auxiliary To Be Served. Next Process = Operation. Process After (this, true);
        Auxiliary To Be Served. Number of Processes Entered++;

// Count Processes
        Log String += " Auxiliary " + Auxiliary To Be Served.Name + " Started Process \" +
this.Name + "\" in its Cycle Number: " + (Auxiliary To Be Served.Current Cycle() +
1).ToString() + " After Waiting for " + Auxiliary To Be Served.Last Waiting Duration().
ToString ("N2") + " Minutes" + Environment. New Line;
        break;
```

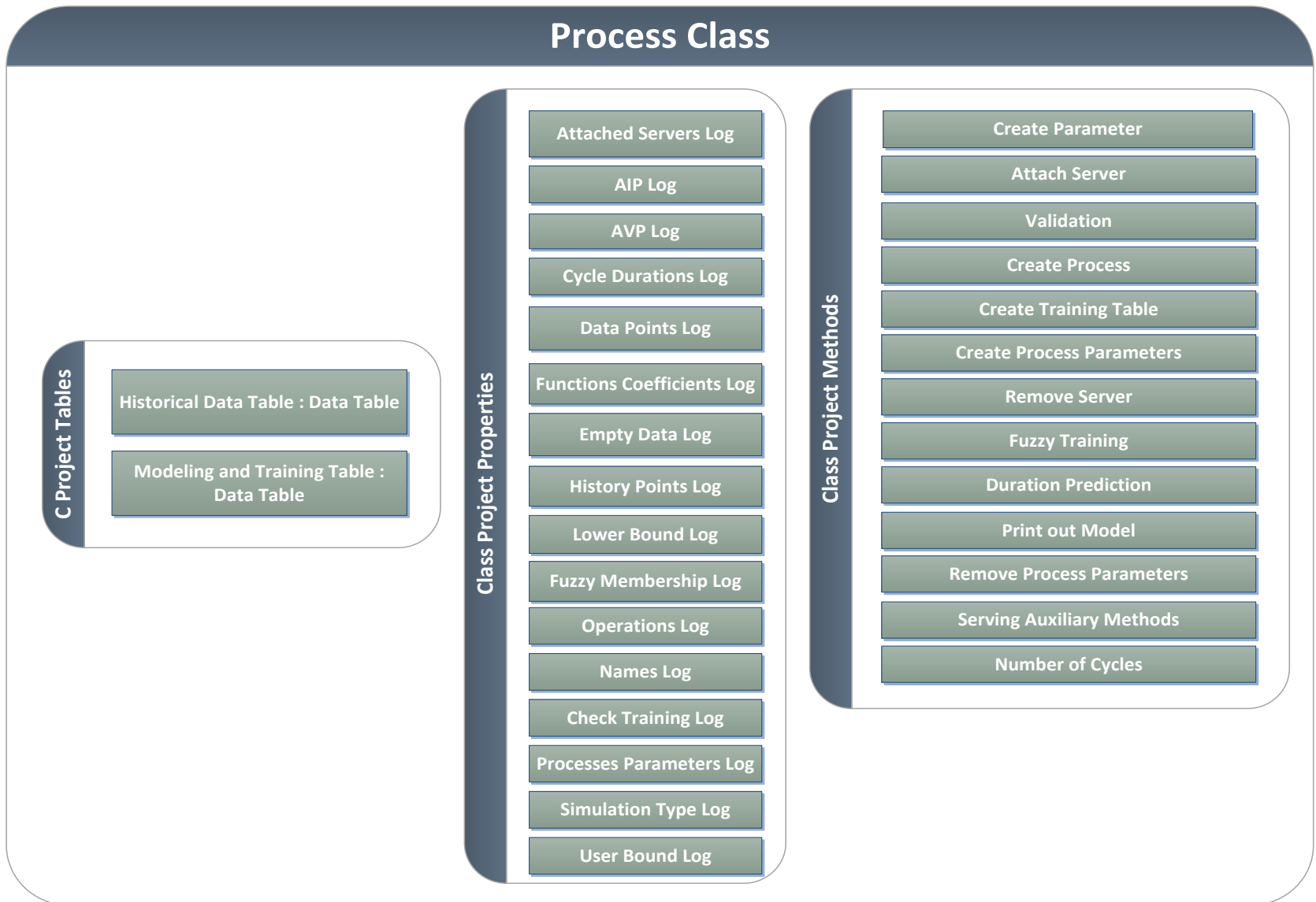


Figure 6-6 Process class components

## 5- Training Validation Method

This method is used to calculate the validation percentage for each process. The validation percentage gives an indication to the decision maker to decide about future actions. This validation depends on the efficiency of this model to predict the future durations. This method calls on the calculated membership functions, and validation and modeling data. It takes the following form:

```
private void Calculate Validation( )
// STEP 1 : Get Membership Matrix for ALL Data Sets
// Read Process Data
// Delete All Column but the Last One in "Durations"
// Input to Fuzzy Module
// STEP 2 : Split Actual modeling Data for validation
// Read Process Data
// Copy Input table and remove last column
// Remove Only Last Column in "DataPoints"
// Delete All Column Except "Durations"
// Begin Validation
    AIP = Get Average Validity Index (Settings);
    AVP = 100 - AIP;
```

## 6- Create Training Tables Method

This method is used to create the training tables. These tables are used in the future duration prediction depending on the validated models. It takes the following form:

```
public void Create Training Table From Parameters()
{
    Data Table dt = new Data Table();

    foreach (CProcess Parameter param in _Parameters)
    {
        DataColumn column = new Data Column();
        column. Data Type = typeof (float);
```

```

        column. Column Name = param. Name;
        column. Caption = param. Name;
        column. Auto Increment = false;
        column. Read Only = false;
        column. Unique = false;
// Finally Add One Empty Data Row
// Copy to All Tables
        Full Data Points Table = dt.Copy();
        Lower Bound Table = dt.Copy();
        Upper Bound Table = dt.Copy();
        History Points Table = dt.Copy();
        Selected Simulation Data = dt.Copy();

```

#### **7- Remove Attached Servers Method**

This method is used to remove the attached servers without removing the server itself from the system. After removing the attached server, the model is not affected and continues without this server and recalculates everything. It takes the following form:

```

public void Deattach Server()
{
    Attached Server Type = null;
}

```

#### **8- Predict Duration Method**

This method is used to select the final duration values for each process. This method integrates with the other methods which are used to calculate, train, and check all required procedures for the process. It takes the following form:

```

private double Predicted Duration()
{
    double Duration = 0;
}

```

```

for (int i = 0; i < Simulation Function Coefficients . Count; i++)
{
    double value = (double) My Simulation Membership Matrix Table. Rows[0][i];

    Duration += value * Simulation Function Coefficients[i];
}

return Duration;
}

#endregion

```

### 9- Print out the Simulation Model Method

This method is used to print the simulation models including all processes. This method will also import the graph to Visio graphical software to make inside/outside modifications. It takes the following form:

```

public string Print Out Data For Visio()
{
    string Data;

    if (this. Attached Server Type != null)
    {
        Data = Name + "\n( Server : " +Attached Server Type.Name + ") \n" +
Attached
        Server Type. Cost + "$ / Unit Time";
    }
    else
    {
        Data = Name + "\n ( No Servers Assigned )";
    }
    return Data;
}

public bool Has No Servers()
{
    if (this. Attached Server Type != null)
        return false;
    else
        return true;
}

```



### **10- Remove Parameters Method**

This method is used to remove the defined parameters that were allocated previously on the processes. It takes the following form:

```
public void Remove Process Parameter ( )  
// Remove Parameter
```

### **11- Remove Servers Method**

This method is used to remove the defined servers. It removes the server completely from the system and its assigned processes. It takes the following form:

```
public void RemoveServers()  
{  
    Attached Server Type = null;  
    Is Busy = false;  
}
```

### **12- Rename Parameter Method**

This method is used to rename parameters' names. The rename is applied also in all related files. It takes the following form:

```
public void Rename Process Parameter()  
{  
    Parameter. Name = New Name;  
}
```

### **13- Count Numbers of Cycles Method**

This method is used to count simulation cycles and issue a command to the system to stop at the required number of simulation cycles. It takes the following form:

```
public void Set Number of Cycles(int Number of Cycles)  
{  
    Current Cycle = 0;
```

```
Cycle Durations.Clear ();  
  
for (int i = 0; i < Number of Cycles; ++i)  
    Cycle Durations.Add (0.0);
```

#### **6.2.4 Auxiliary Instance Class**

Auxiliary Class contains and presents all Auxiliary Instance related data as average waiting time, average working time, and cycles' information. As shown in Figure 6.7, Auxiliary Instance class consists of **11** fields, **8** properties, and **5** methods. Project class takes the following form:

Class Auxiliary Instance Class

```
{  
  
// Fields, Properties and methods go here  
  
}
```

##### **6.2.4.1 Methods**

Methods represent any processing job on all types of data in the system that relates to auxiliary instance. There are 5 methods. The methods are as following:

###### **1- Average Waiting Time Method**

This method is used to calculate the average waiting time for each auxiliary. The average waiting time for the auxiliary signals that process is busy with another auxiliary, while the other auxiliary is waiting. It also means that the auxiliary is idle in a queue. It calculates the waiting time from the status of the auxiliary and process, because the system is checking the status of each server and auxiliary to conclude on the status of the process. It takes the following form:

```

public double Average Waiting Time()
{
    double Sum = 0;

    // for Each cycle
    for (int i = 0; i < Number of Cycles * Operation. Processes. Count; i++)
    {
        Sum += Waiting Durations [i];
    }

    return Sum / Number Of Cycles;
}

```

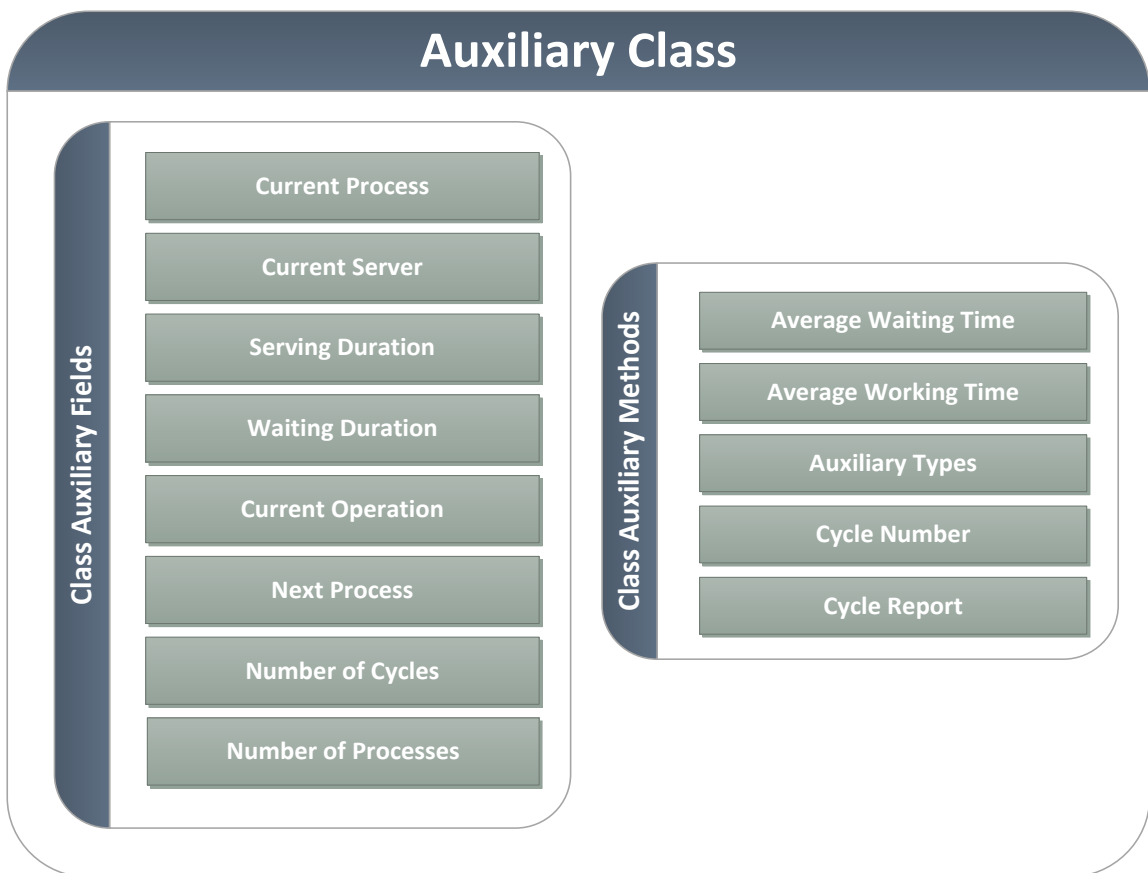


Figure 6-7 Auxiliary class components

## 2- Average Working Time Method

This method is used to calculate the average working time for each auxiliary. The average working time for the auxiliary means that the process is either busy with that

auxiliary or the auxiliary is in the cycle in another process. The system recognizes the place of the auxiliary by checking the status of the auxiliary at every fixed interval and duplicates this action for the same auxiliary motion. It takes the following form:

```
public double Average Working Time()
{
    double Sum = 0;

    // for Each cycle
    for (int i = 0; i < _Number of Cycles * Operation. Processes. Count; i++)
    {
        Sum += Working Durations [i];
    }

    return Sum / Number of Cycles;
```

### 3- Auxiliary Type Method

This method is used to distinguish between different types of the same auxiliary. Accordingly, if there are two identical trucks that only differ in their names, this method can distinguish between them. It takes the following form:

```
public CAuxiliary Instance (CAuxiliary Type Auxiliary Type)
{
    Auxiliary Type = Auxiliary Type;
    Operation = Auxiliary Type. Operation;
    Current Process = null;
    Next Process = null;
    Current Server = null;
    Current Cycle = 0;
    Number of Processes Entered = 0;
    Is Waiting = true;
    Name = My Auxiliary Type. Name + "( " + (Auxiliary Type. Instances. Count + 1).ToString() + ")";
```

### 4- Cycle Number Method

This method is used to determine the cycle time from the auxiliary motion, meaning that the auxiliary counts the cycle number starting from the first through the last process.

Depending on this number, the Cycle Number Method reports its motion to the system log. It takes the following form:

```
public int Current Cycle()
{
    int ans = Number of Processes Entered / Operation. Processes. Count;

    if (Number of Processes Entered % Operation. Processes. Count == 0 & ans != 0)
        ans - = 1;

    return ans;
```

### 5- Cycles Report Method

This method is used to report the outputs of different system elements that relate to auxiliaries namely as waiting times, working times, and productivity. It calculates the number of cycles per time unit depending on the motion of the auxiliaries. It takes the following form:

```
public override string Print Log (int Number of Cycles)
{
    string Log = "";
    Log += this. Auxiliary Type. Name + " ( " + (this. Auxiliary Type. Index(this) + 1).To String( ) + " ) " + Environment. New Line + Environment. New Line;
    Log += "Average Waiting Time = " + this. Average Waiting Time().To String("N1") + " Minutes " + Environment. New Line;
    Log += "Average Working Time = " + this. Average Working Time().To String("N1") + " Minutes " + Environment. New Line;
    Log += "Productivity Per Auxiliary = " + ((this. Average Waiting Time() + this. Average Working Time()) / this. Auxiliary Type. Instances. Count).To String("N3") + Environment. New Line + Environment. New Line;

    return Log;
```

### 6.3 Criteria for the Selection of Programming Development Tools

In developing the proposed system, different programming languages can be used. The programming language selection process considers certain language features such as the availability, the ability to integrate with other software systems, the ability to handle a complex computation in a short time, and to provide a user-friendly interface. The C#

language was selected to develop the proposed system. C# is a new language with the power of C++ and the smoothness of Visual Basic. C# offers the potential of being available across many platforms. It is a very powerful high-level language; for example, to develop something that C requires 100 lines of code would require only 10 lines of code in C#. C# is an object-oriented programming language encompassing imperative, declarative, functional, generic, and component-oriented programming languages. C# is one of the programming languages that have been designed for the Common language Infrastructure and as such will give the proposed system the capability of being developed and integrated with other systems and platforms in the future.

#### **6.4 System's Architecture**

As stated in Chapter 3 and as shown in Figure 6.8, the proposed system consists of three main stages presented in four module plus a graphical user interface (GUI) . The first module is the training module, which defines and collects the project description and data, the operation description and data, the process description and data, the variables that affect the processes, the servers and auxiliary units. This module utilizes data mining techniques to prepare the project data and to train the system to predict process durations. This module ends with a validation. The second module is the simulation model, which models the movement of the construction operation based on the information from the training module. The third module is the optimization module that develops the sensitivity analysis and selects and ranks the optimum solution(s). The fourth module is the reporting module, which generates reports of the training process and the simulation process.

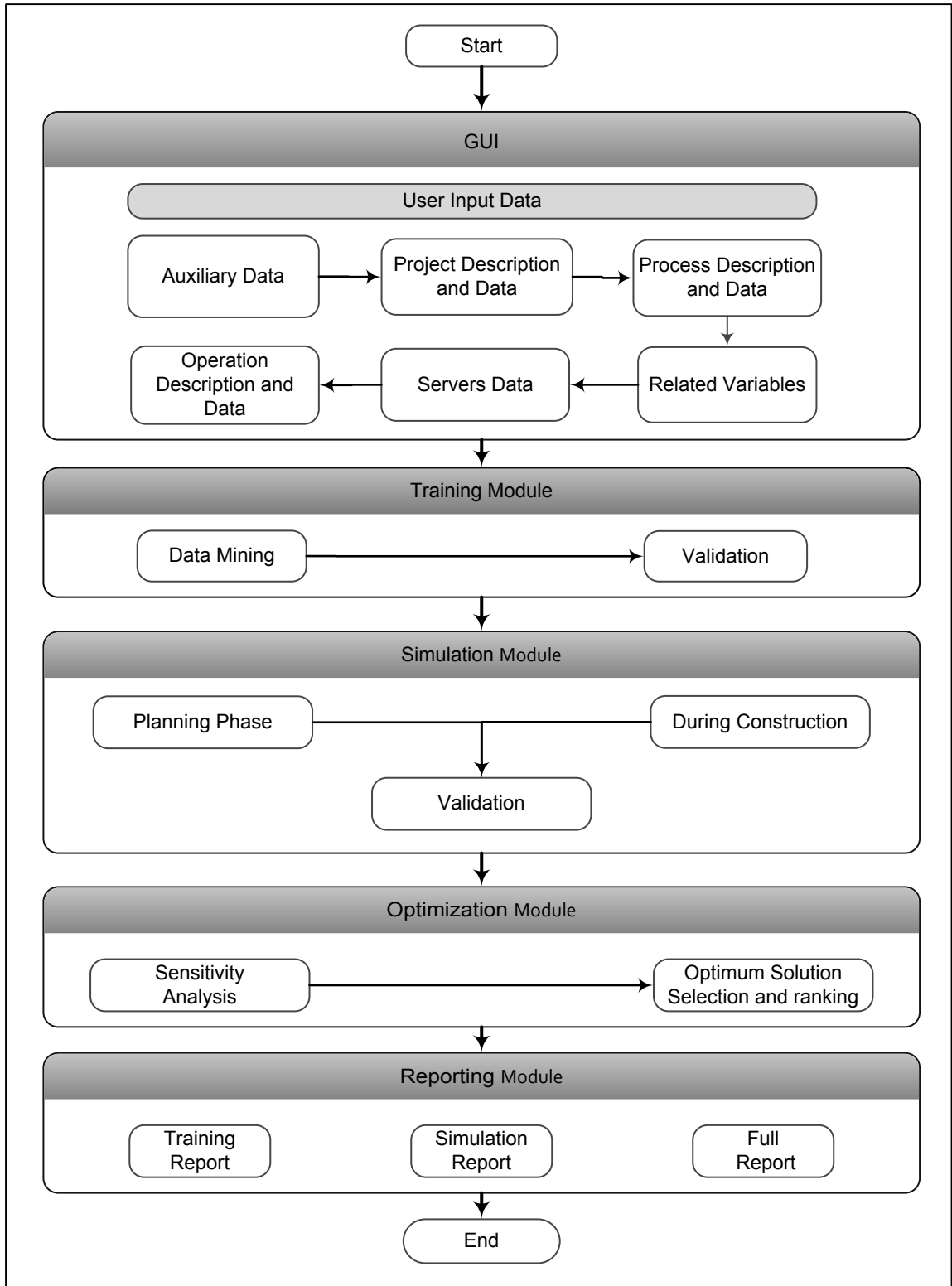


Figure 6-8 System architecture and data flow

The graphical user interface (GUI) is a human-computer interface that uses windows, icons and menus and can be manipulated by a mouse (and often, to a limited extent, by a keyboard as well) to create an interactive environment between the user and the system. The flexibility of integration between all of the module and the GUI was taken into consideration in designing the proposed system architecture.

### **6.5 Data Flow of the Developed System**

The system begins by receiving the project data from the user through the graphical user interface (GUI) using an interactive set of dialog window interfaces. After checking the required data, the output will be the input of the training module. The training module is where the data are cleaned, a data mining module is built and the relation between qualitative and quantitative variables is modeled. The validation results guides the user as to whether he/she will continue or not. The output of the training module becomes the input of the simulation module. The simulation module is carried out during a planning or a construction phase. The difference between the two phases is the availability of actual data or predicted ranges. The simulation module is validated by user measurements and/or judgments. Depending on the user's decision, the optimization module will then begin. The optimization module consists of two phases. The first is a sensitivity analysis that evaluates the effect of variability among the variables that affect the system and the assigned resources and optimum solution/s. The second phase is the ranking phase wherein the optimum solution or solution(s) are selected and ranked. The last module is the reporting module, which can produce three types of reports. The first is a training report that graphically and analytically presents the details of the training process and validation. The second type is a simulation report that presents the details of the



simulation procedure and a chronological analysis. It includes details of the system status and all its components, such as servers' working and idle times, as well as the status of the other system components. The third type of report is an integrated full report.

## **6.6 Graphical User Interface (GUI)**

The graphical user interface incorporates menus, toolbars, spreadsheets, and dialog windows with an object-oriented programming technique using C#. The system has been designed to help the user, facilitating data entry and the validation of input data. The system is a standalone type, which means the system does not depend on any other program or core program. Its standalone status makes the program easy to use and it can be compatible with copyrights regulations. Figure 6.9 depicts the graphical user interface of the developed system, and contains all the menus, toolbars and various models.

## **6.7 Applying the Developed System on a Case Study**

As discussed in Chapter 4, the data that were recorded and measured from the jobsites and from the Internet provided consistent observations for analyzing the event times of the concrete pouring process: hauling, loading and return. In Chapter 5 the system was only applied on the concrete pouring process, but in this chapter the system will be applied on the whole concrete pouring operation.

### **6.7.1 Input the Project Data**

As shown in Figure 6.9, all of the project parameters are described. The project data is uploaded from spread sheets or manually for each variable. The server (Concrete Pump) is defined as shown in Figure 6.10, by Name, Cost/hour, and number of entities in the original operation. The minimum and maximum numbers of entities are defined for

sensitivity analysis purposes. The auxiliary (Truck) is defined as shown in Figure 6.11, by Name, Cost/hour, and number of entities in the original operation. The minimum and maximum numbers of entities are defined for sensitivity analysis purposes. The servers will be assigned on the processes. The system is run under different variable conditions, taking into consideration the minim and maximum bounds of variables and resources while fixing all other variables and resources. The definition of each variable defines the feasible solutions area and the selection of optimum solution(s). The system can graphically provide the model illustration and all project details as shown in Figure 6.12. This window is shared using MS Visio to indicate its powerful features. In a case of standalone status, the model's graph will be unmodified.

### **6.7.2 Training Module**

This module can begin after the input data has been validated by the system. Here the processed data is prepared to be modeled and further validated using the validation data. This module is the key component for preparing the data for the simulation module. The training module utilizes a data mining engine to perform multiple functions such as modeling the processed data, preparing a fuzzy knowledge base, and simulating the predicted environment for future projects. This module consists of a set of functional steps: variable selection, forming fuzzy sets, fuzzy rule inductions, and building a fuzzy knowledge base. The training module defines and selects the variables that affect the output variable (work task duration) using the fuzzy average method in two steps; the production of the fuzzy curve followed by the ranking process.

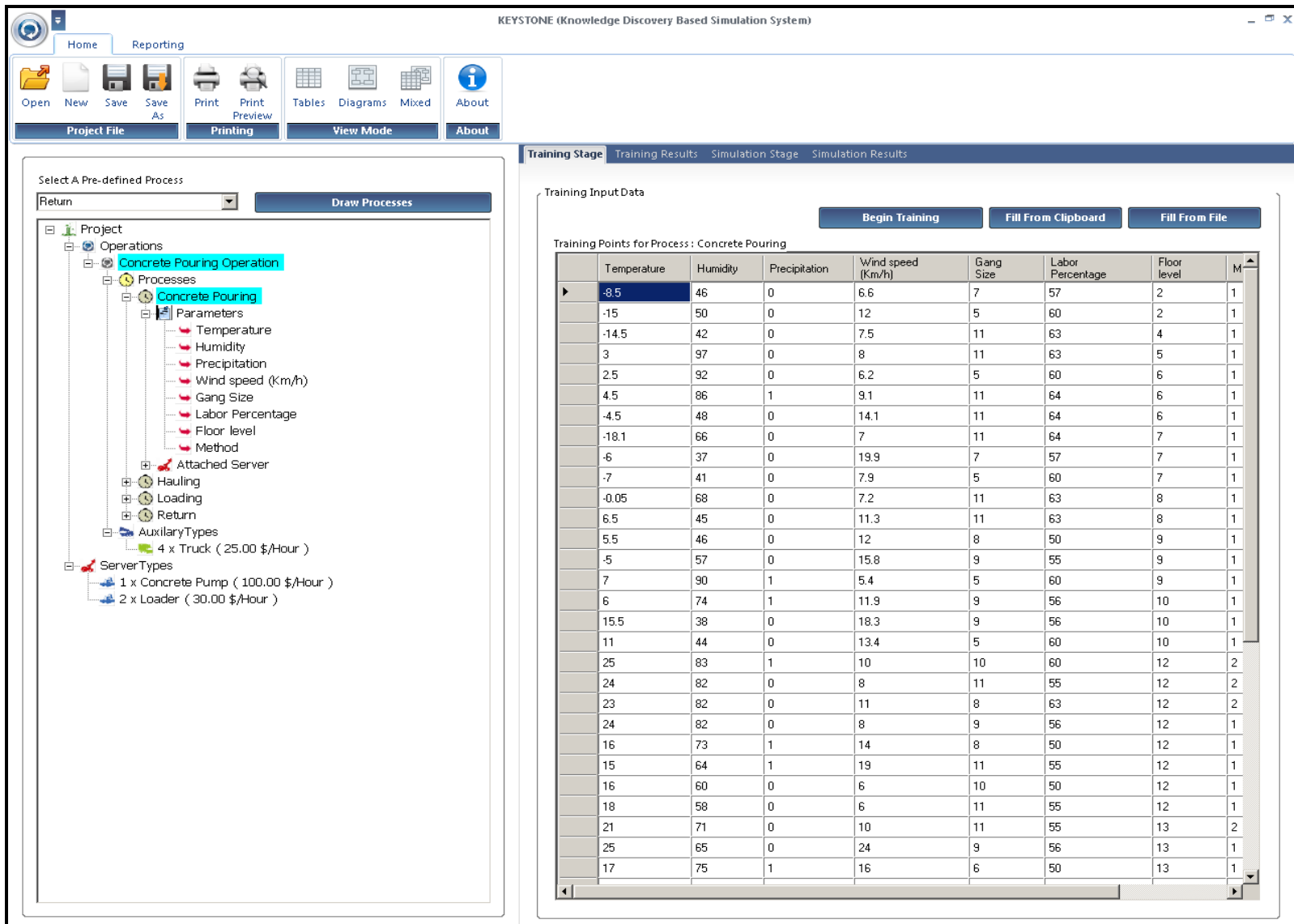


Figure 6-9 Graphical user interface (GUI)

After selecting the critical variables, fuzzy sets are formed, through which crisp quantities are converted into fuzzy sets. The conversion to fuzzy values is represented by the membership functions. The Fuzzification process involves assigning membership values for the given crisp quantities. Membership values are assigned using the neural network technique. Fuzzy rule induction, representing the functional relationships between the independent variables' membership values and the values of the dependent variables (Task Duration) is conducted next. The set of all the past steps is known as the Fuzzy Knowledge Base (FKB). Before the FKB can be considered finalized, the validation process is started. The goal of this validation is to test the system's prediction effectiveness. The output of the training module is the actual versus the predicted plot validation test, as shown in Figure 6.13. and the validity and invalidity percent test. The AVP is 77% and AIP is 33%. The modeler decides to accept the results (or not) to continue to other steps.

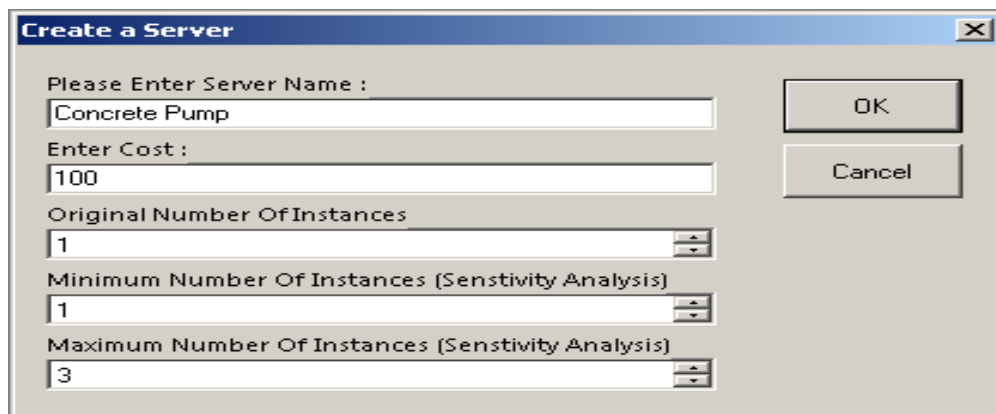


Figure 6-10 Create and define a server

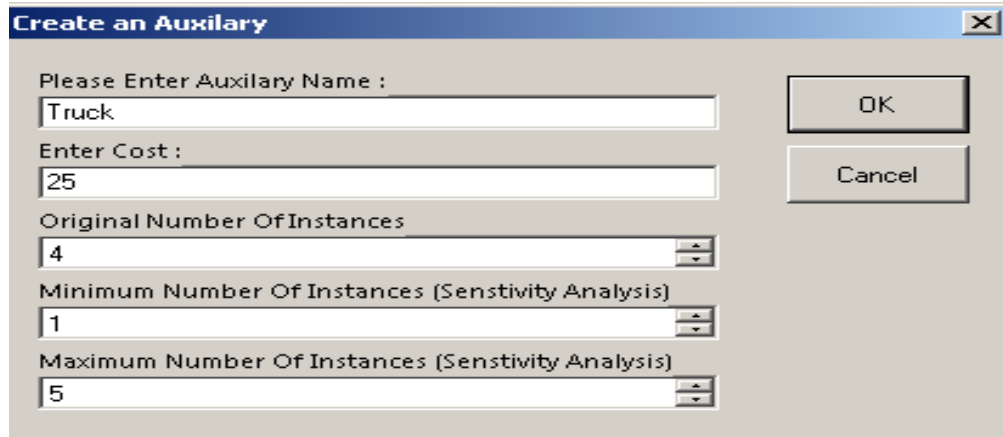


Figure 6-11 Create and define an auxiliary

### 6.7.3 Simulation Module

In this simulation module, the movements of units (Trucks) in concrete pouring operation are modeled. This module examines the interaction between flow units and the resource idle times to flag the bottlenecks and estimate the productivity of the proposed operation. This module starts by selecting the proper stage, either a planning or during construction stage. In this case, the planning stage is studied. As shown in Figure 6.14, the module requires the number of simulation cycles and the upper and lower bounds of all variables to be provided. In the planning stage, the input variables that affect the duration are identified from the previous module, studied, and transformed into fuzzy membership functions via the data mining engine. The fuzzy knowledge base will then be used to predict task durations.

### 6.7.4 Optimization Module

This module starts as an output of the simulation module. The first goal of this module is to evaluate the effect of changing the variables on the existing state and on the system output. The second goal is to rank and select the optimum solutions.

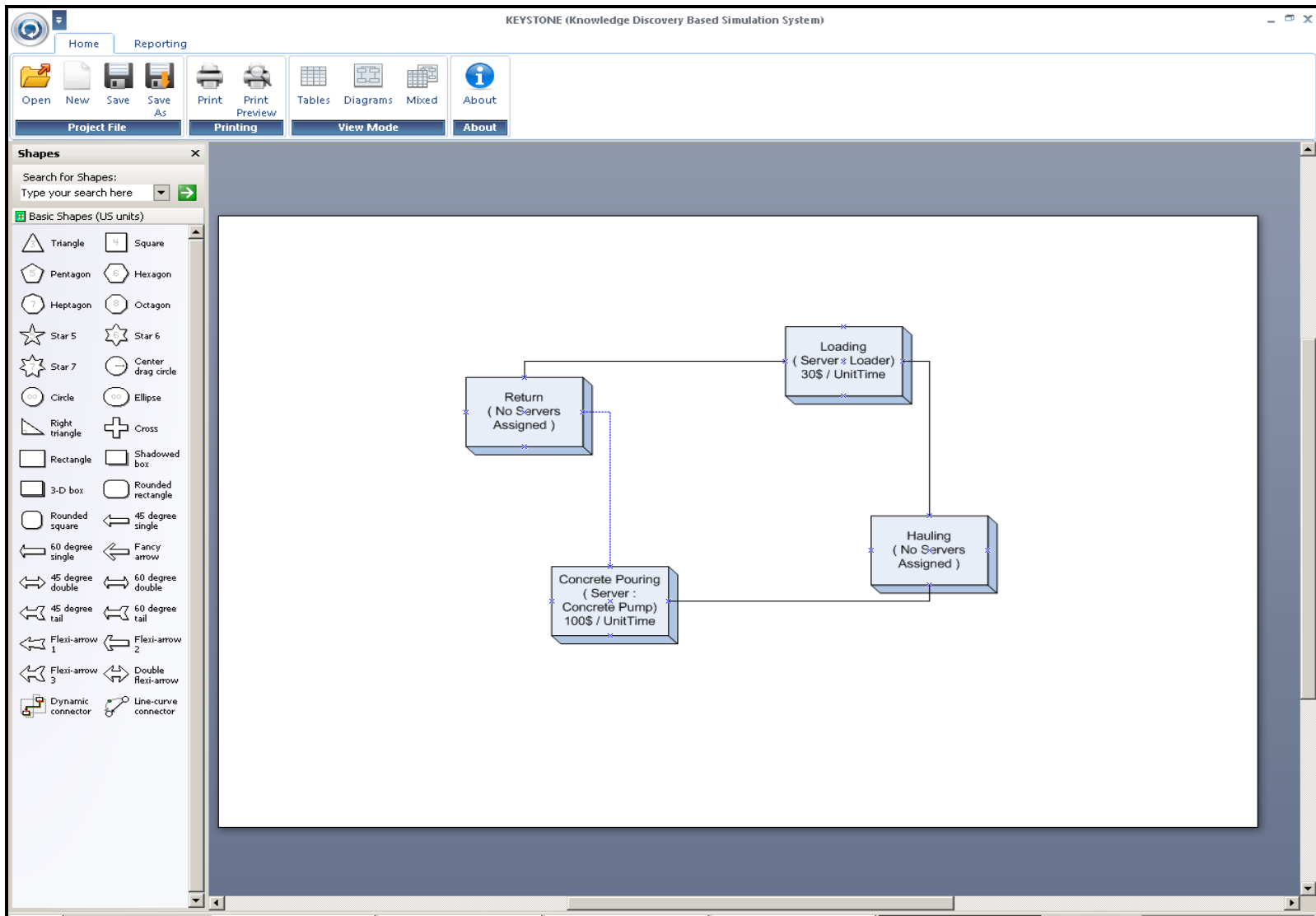


Figure 6-12 Graphical representation of the project

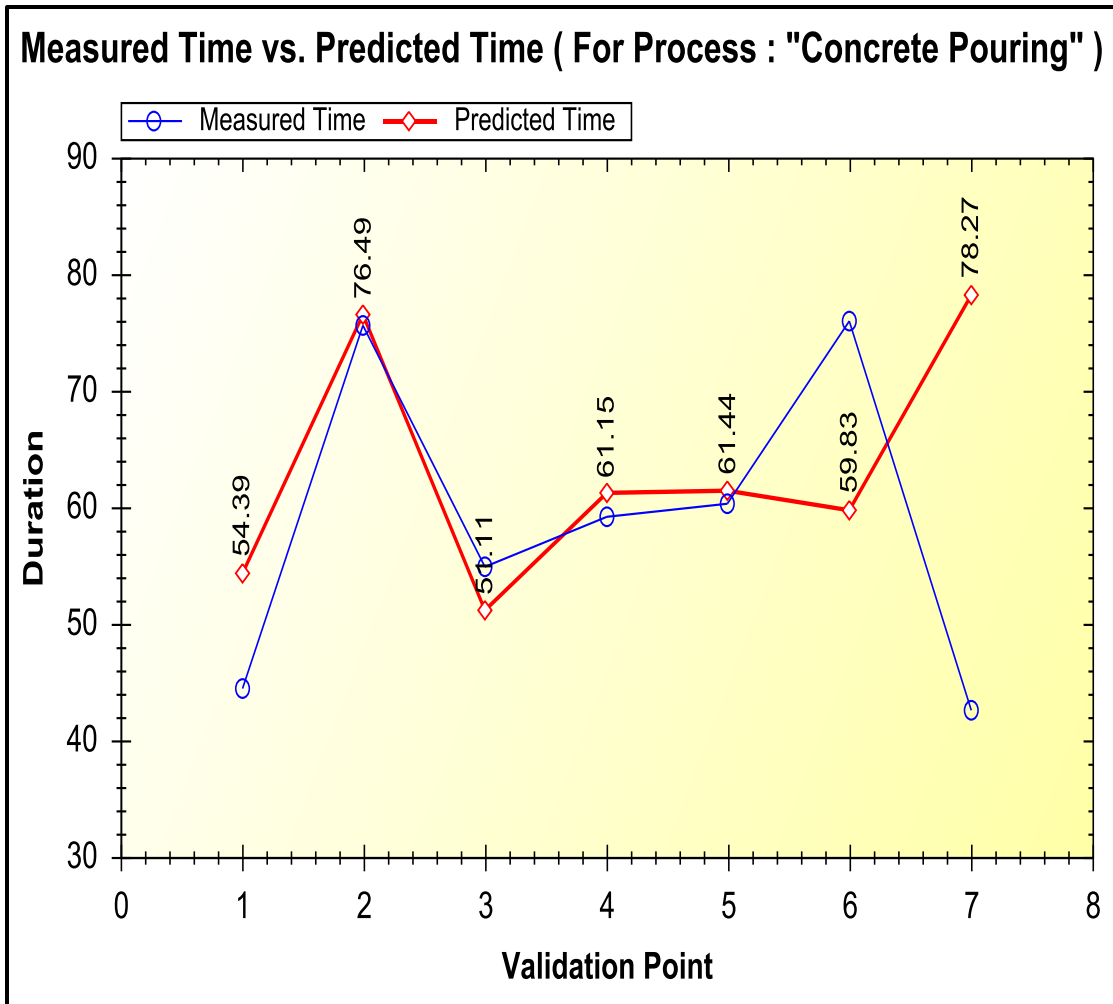


Figure 6-13 Actual vs. predicted concrete pouring durations

This module works in three steps. The first step only conducts the sensitivity analysis at the resource level. In the second step, sensitivity analysis at the project level is carried out to find the effect of each variable on the optimum solution(s), as well as other feasible solutions. The selection and ranking of optimal solutions is done in the third step, using a rank assignment algorithm. The first output is the sensitivity analysis, as shown in figure 6.15. This output shows the validity of the optimum solution on all system states and gives the decision maker a wide overview of the project. The decision maker can see the

project under the best and worst examples of weather conditions and also for resource allocation. The second output, the ranking and selection of the optimum solution(s), is shown in Figure 6.16. The ranking results give the decision maker a group of feasible solutions ranked in descending depending on the unit cost and productivity. For example, as shown at points one and two, the productivity is equal to 0.4402 cycle/minute, but they have different costs of 2.58 (\$ / minute) and 3.08 (\$ / minute), respectively. The system has ranked point one before point two and so on for all other points as it develops the optimum ranking solution(s).

### **6.7.5 Reporting Module**

This module presents the results of different modules and a final result report. As shown in Figure 6.17, the simulation report consists of six groups of data. The first group is project productivity that contains the operation productivity per unit time (0.242 cycle / hour) and the cost per unit time (4.33 \$ / minute). The second group is the calculated process durations. In this group, all the estimated durations are shown and compared at each cycle. The third group contains the chronological steps for the simulation process and shows the status of the system at all cycle times. The fourth group is the auxiliary waiting and working times. The fifth group is made up of the servers' waiting and working times. The sixth is the empty and busy process times. All of the reports are presented in PDF format files to be easy for presentation, to require minimal a disk space, and to be resistant to tampering.



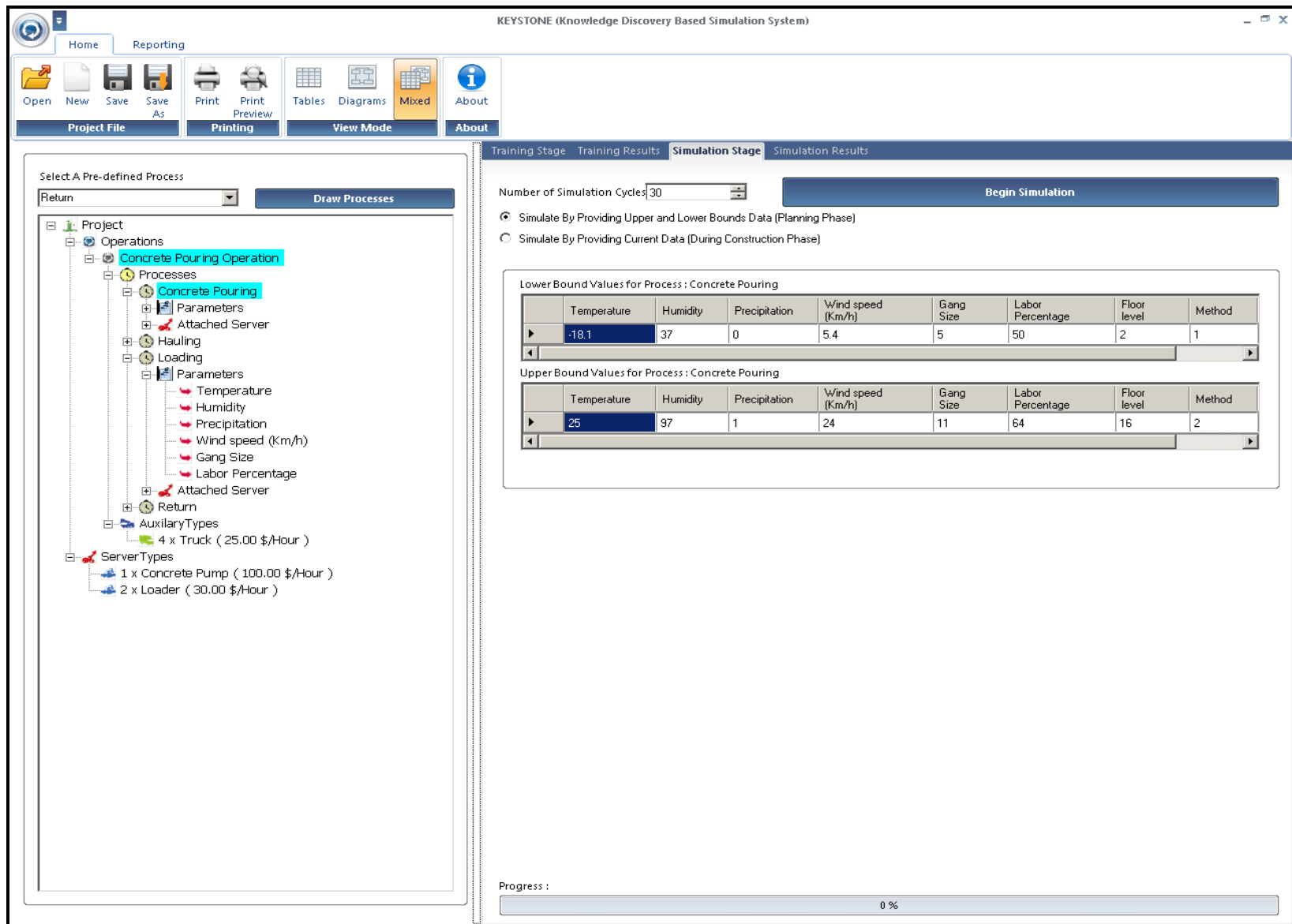


Figure 6-14 Simulation module in the planning stage

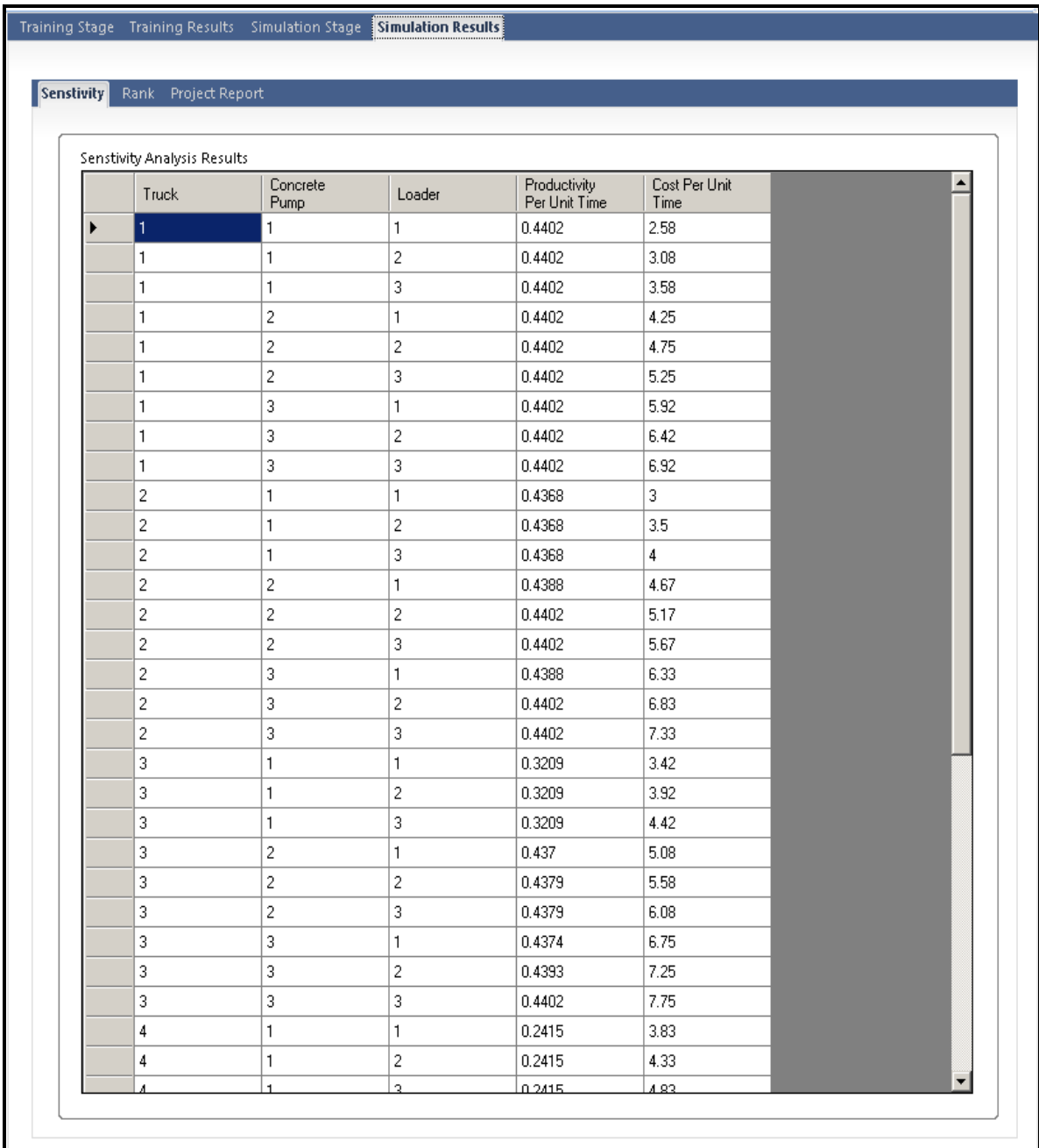


Figure 6-15 Sensitivity analysis results

Training Stage Training Results Simulation Stage **Simulation Results**

Sensitivity **Rank** Project Report

Rank Results

	Truck	Concrete Pump	Loader	Productivity Per Unit Time	Cost Per Unit Time
▶	1	1	1	0.4402	2.58
	1	1	2	0.4402	3.08
	2	1	1	0.4368	3
	1	1	3	0.4402	3.58
	2	1	2	0.4368	3.5
	3	1	1	0.3209	3.42
	1	2	1	0.4402	4.25
	2	1	3	0.4368	4
	3	1	2	0.3209	3.92
	4	1	1	0.2415	3.83
	1	2	2	0.4402	4.75
	2	2	1	0.4388	4.67
	3	1	3	0.3209	4.42
	4	1	2	0.2415	4.33
	5	1	1	0.1936	4.25
	2	2	2	0.4402	5.17
	3	2	1	0.437	5.08
	4	1	3	0.2415	4.83
	5	1	2	0.1936	4.75
	1	2	3	0.4402	5.25
	2	2	3	0.4402	5.67
	3	2	2	0.4379	5.58
	4	2	1	0.4354	5.5
	5	1	3	0.1936	5.25
	1	3	1	0.4402	5.92
	1	3	2	0.4402	6.42
	2	3	1	0.4388	6.33
	3	2	3	0.4379	6.08
	4	2	2	0.4368	6

Figure 6-16 Ranking and selecting the optimum solution(s)



 <h1 style="text-align: center;">SIMULATION REPORT</h1> 
<p><b><u>Project Productivity :</u></b></p> <p><b><u>Calculated Durations For Processes :</u></b></p> <p><b><u>Chronological Steps :</u></b></p> <p><b><u>Auxiliaries :</u></b></p> <p><b><u>Servers :</u></b></p> <p><b><u>Waiting Time for Processes :</u></b></p>
Simulation Results

Figure 6-17 Simulation results report (Template)

# Chapter 7: Conclusions and Recommendations

## 7.1 Summary and Conclusions

This thesis has presented a new knowledge discovery based construction simulation system for construction processes. The system handles the problem of input data sets, utilizes the massive amount of available historical data, models the effect of qualitative and quantitative variables, and optimizes the simulation system output to enhance the simulation and modeling of construction processes.

The developed system consists of three stages. The first is the Knowledge Discovery Stage (KDS) where raw data are prepared for the subsequent simulation stage. Patterns are extracted, which implicitly represent knowledge that is stored or captured in large databases. This stage models the effect of qualitative and quantitative variables on the construction process. The KDS includes data cleaning, integration, selection, transformation, and mining as well as pattern evaluation and knowledge presentation. The cleaning step includes identifying outliers and filling missing data. To predict the missing values, K-means, Fuzzy K-means, and the average method are used to validate the developed method. The AVP calculations show that the best method is the Fuzzy K-means method with a score of 85%. The worst method is the average method, with an average validity percent (AVP) of 70%. The data models are compared before and after filling in missing data and removing outliers where it is found that cleaning framework improved data pattern detection by 10%. The results of data mining engine in selecting the variables that affect construction process using Fuzzy average method comes to an  $R^2$  equals to 91.3%. The Artificial Neural Networks and Regression techniques produced  $R^2$

values of 88.8% and 88%, respectively, which shows that the fuzzy average method is better in selecting the variables that affect the construction processes. The end result of the KDS is fuzzy knowledge base, which is validated using the AVP test, producing a 92 % validation result.

The second is the Simulation Stage (SS) where the movement of system units and resources are modeled. The SS then examines the interaction between the flow units and resource idle times to discover bottlenecks and estimate productivity and unit cost of the proposed operation. This stage consists of two phases; simulation engine design and system validation. Using deterministic methodology, the developed system is shown to have made an improvement, with results that are closer to the real collected data by 4% to 11%. This improvement is the result of using data mining techniques to improve the quality of data, of including the qualitative and quantitative variables affecting the construction process, and of utilizing historical data to improve the duration prediction.

The third is the Optimization Stage (OS) where optimal productivity and cost solutions are discovered and ranked. The first goal of this stage is to evaluate the effect of changing the variables on the existing state and system outputs. The second goal is to rank and select the optimum solutions using the Pareto ranking assignment algorithm. This stage gives the decision maker the ability to see the system under the best and worst conditions of resource combinations that maximizes productivity and minimizes cost.

The system is developed based on the design and assembly of a general purpose Construction simulation language (KEYSTONE), an acronym for **KnowlEdgE discoverY baSed construcTiON simulatiON systEm**, to build a simulation environment platform for modeling construction operations. The KEYSTONE language includes most of the

features that are desirable to a general purpose simulation language. The KEYSTONE language is designed to write and generate object-oriented simulation code. Construction operations are automatically translated into C#. The system is then developed under the development platform of KEYSTONE and using C#. The system platform considers and promotes certain features, such as the ability to integrate with other software systems, the capability of handling a complex computation in a short time, and the potential of providing a user friendly interface.

The language's core consists of four classes (i.e. levels): project, operation, process, and auxiliary. Each class consists of three objects: fields, properties, and methods. The fields contain all data types in the project. The project consists of operations, processes, servers and their related components. The operations and servers are in the same level so that the servers can be shared among operations. The interaction between all of the components is carried out by the moving units or auxiliaries that represent the simulation model. The KEYSTONE language deals with the model as classes not as components, which gives the modeler the ability to extend the simulation model over more than one operation and even more than a single project. The developed system is tested using a construction case study with satisfactory results. The developed system with KEYSTONE allows researchers and practitioners to effectively utilize historical construction data and to improve simulation and modeling processes.

## **7.2 Research Contributions**

This research presents the following contributions to the state of the art in construction simulation:

- Develops a strategy and methodology to manage the uncertainties in a project and the associated data in the planning and construction stages.
- Designs a data mining engine to find hidden data patterns and to simulate the predicted environment of future projects.
- Integrates the enormous amount of historical data and design into a fuzzy knowledge base that can be used to predict process' durations instead of approximated probability distribution.
- Designs the integrated framework of discrete event simulation and a fuzzy knowledge base.
- Prioritizes the optimum resource combinations of any process while considering both the qualitative and quantitative variables that affect this process.
- Designs and builds an interactive general purpose construction simulation language (KEYSTONE) and programming environment to model construction processes and provides effective user interaction during the simulation process.

### **7.3 Research Limitations**

The developed system and the general purpose construction simulation language (KEYSTONE) entail the following limitations:

- 1- The system is limited to cyclic construction operations, i.e. does not consider non-cyclic operations. It is also not suitable to do analysis in the project or organization levels.
- 2- This research only considers the factors that affect duration of construction operations; however, it doesn't consider the factors that affect cost elements of such operation.



- 3- Although the automated system is standalone as simulation software, it doesn't have the capabilities to make an interface with external database management systems and doesn't support web based technology.

#### **7.4 Recommendations for Future Research**

Below is a list of subjects that can be considered for future work in this area to enhance and extend the developments made in this thesis:

##### **I. Current study enhancement areas:**

- 1- This study has only modeled the relation between qualitative/quantitative variables and the construction process. It is recommended to also study the relation between variables affecting the project and their effect on the construction operation. For instance, the relation between the efficiency of the project organization and that of the construction operation.
- 2- It is crucial to address the relation between the number of variables and the number of data points to enhance process duration prediction. It is probable that if data size is increased, the process duration prediction might be improved. However, in some cases, data sets are limited. Therefore, it is recommended to build indexes and graphs to guide model builders with a reliable threshold of the number of variables and data sets in addition to their interaction with the validation percentage.
- 3- The developed simulation methodology can be utilized as part of a project control system by updating the states of processes based on actual performance, which improves forecasting project performance. The simulation model can be embedded in the project control system with an interactive interface. Instead of

using the developed system only in the planning and during construction stages; it can be updated with the construction schedule to help the control team precisely update their progress and take informed decisions.

- 4- Visualization and animation are two features that enable the decision maker to see the system during work simulation and effectively highlight bottlenecks.
- 5- Provide some process templates to save time during system modeling. Even though it is difficult to prepare some complete process templates because of the unique nature of the construction industry, it would be better to design and build them for specific disciplines or even types of projects, such as tunnels or bridges. The developed system provides the ability to modify and integrate these templates.

## **II. Current study extension areas:**

- 1- Cost historical data is important to improve the cost per unit time prediction of the developed system. There are many factors that affect unit cost. These factors should be identified and studied to be included during simulation of construction operations. These factors help the user to determine and overcome the problems of cost fluctuation.
- 2- Non-cyclic networks are essential to solve many construction problems. Scheduling non-cyclic networks, for example, are extensively used to determine project's duration. It is however crucial to expand the capabilities of the developed system so that it will be able to address wide range of construction operations, i.e. cyclic and non-cyclic.

- 3- The developed system has the ability to save and retrieve the presented models any time; however, designing a data management system will increase the system ability of self-learning. Moreover, it will offer more features of data analysis and storage techniques, such as query language and structured way to retrieve data sets.
- 4- The developed system considers an equal priority for all model branches but it would improve the modeling process if different priorities are allowed. For instance, if there is an auxiliary in the cycles with two options (branches) to go for process one or two, the developed system will give an equal priority for both options, but it is recommended to assign different priorities.
- 5- It is recommended to develop a web-based version for the developed simulation system to make it easy to access and upgrade. The KEYSTONE language can be integrated with any other development platform, e.g. Java. A web-based application will present clear benefits for users and developers. For users, it would be quite simple to access, upgrade, and store. For developers, it could present an interactive environment that collects feedback from other parties to improve the simulation system's capabilities.
- 6- The data import engine in the developed system should be improved to transform the data from different data sources, e.g. GPS systems. The data import engine improvement will extend the capability of the developed system to interact with other data sources and also other construction software tools, such as the Primavera and Pertmaster packages.

## References

- AbouRizk, S, M. (1990). Input modeling for construction simulation. *Doctoral dissertation, Purdue University, W. Lafayette, In.*
- AbouRizk, S., & Mohamed, Y. (2000). Symphony: An integrated environment for construction simulation. *Proceedings of the 32nd Conference on Winter Simulation, 1907-1914.*
- Anand, S., Patterson, D., Hughes, J., & Bell, D. (1998). Discovering case knowledge using data mining. *Proceedings of the conference on Research and Development in Knowledge Discovery and Data Mining, 25-35*
- Bao, Yang and Zhang, LuJing (2010). Decision support system based on data warehouse. *Journal of World Academy of Science, Engineering and Technology, 71, 172-176.*
- Berkan, R. C., & Trubatch, S. (1997). Fuzzy system design principles *Wiley-IEEE Press.*
- Bishop, J. L., & Balci, O. (1990). General purpose visual simulation system: A functional description. *Proceedings of the Conference on Winter Simulation, 504-512.*
- Bock, H. H. (1974). Automatische Klassifikation. Göttingen: *Vandenhoeck and Ruprecht.* An introduction and survey for clustering methods and their mathematical and statistical bases.
- Boulicaut, J. F., & Jeudy, B. (2010). Constraint-based data mining. *Data Mining and Knowledge Discovery Handbook, 339-354.*

- Branke, J., Deb, K., & Miettinen, K. (2008). Multiobjective optimization: Interactive and evolutionary approaches *Springer-Verlag New York Inc.*
- Buchheit, R., Garrett Jr, J., Lee, S., & Brahme, R. (2000). A knowledge discovery framework for civil infrastructure: A case study of the intelligent workplace. *Journal of Engineering with Computers, 16(3), 264-274.*
- Cedeño, W., Vemuri, V. R., & Slezak, T. (1994). Multi niche crowding in genetic algorithms and its application to the assembly of DNA restriction-fragments. *Journal of Evolutionary Computation, 2(4), 321-345.*
- Chapman, P., Clinton, J., Khabaza, T., Reinartz, T., & Wirth, R. (1999). The CRISP-DM process model. *The CRIP-DM Consortium, 310.*
- Christopher M. T. (2009). A multi-tiered genetic algorithm for data mining and hypothesis Refinement. *Doctoral dissertation, the University of Kansas, USA.*
- Chau, K., Cao, Y., Anson, M., & Zhang, J. (2003). Application of data warehouse and decision support system in construction management. *Journal of Automation in Construction, 12(2), 213-224.*
- Cho, S. Y., Chow, T. W. S., & Leung, C. T. (1999). A neural-based crowd estimation by hybrid global learning algorithm. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions, 29(4), 535-541.*
- Corder, G. W., & Foreman, D. I. (2009). Nonparametric statistics for non-statisticians: A step-by-step approach *John Wiley & Sons Inc.*
- Cox, E. (2005). Fuzzy modeling and genetic algorithms for data mining and exploration *Morgan Kaufmann Publishers, Elsevier.*

- Deogun, J. Spaulding, W. Shuart, B., Li, D. (2004). Towards missing data imputation: A study of fuzzy K-means clustering method. *Journal of Rough Sets and Current Trends in Computing*, 573-579.
- De Raedt, L., Blockeel, H., Dehaspe, L., & Van Laer, W. (2001). Three companions for data mining in first order logic. Conference of *Relational Data Mining*, 3, 105-139.
- Detlef Kutsche, R. and Milanovic, N. (2008). Model-based software and data integration: *First international workshop, MBSDI 2008, Proceedings Springer-Verlag New York Inc.*
- Dikmen, I., Birgonul, M. T., & Kiziltas, S. (2005). Prediction of organizational effectiveness in construction companies. *Journal of Construction Engineering and Management*, 131, 252.
- Du, K., Swamy, M., MyiLibrary, L., & SpringerLink (Online service). (2006). *Neural networks in a soft computing framework Springer.*
- Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, 17(3), 37.
- Fisher, Deborah J. (1997) .The knowledge process. Conference of *Lean construction.4*, 17-23.
- Fonseca, C. M., & Fleming, P. J. (1993). Genetic algorithms for multi objective optimization: Formulation, discussion and generalization. *Proceedings of the Fifth International Conference on Genetic Algorithms*, 2, 423 416–423.
- Fonseca, C. M., & Fleming, P. J. (1995). An overview of evolutionary algorithms in multi objective optimization. *Journal of Evolutionary Computation*, 3(1), 1-16.

- Fung, B., Wang, K., Wang, L., & Debbabi, M. (2008). A framework for privacy-preserving cluster analysis. *Intelligence and Security Informatics, 2008, ISI 2008. IEEE International Conference on, 46-51.*
- Hajjar, D., and AbouRizk, S. (2002). Unified modeling methodology for construction simulation. *Journal of Construction Engineering and Management, 128, 174.*
- Halpin, D. W. (1973). An investigation of the use of simulation networks for modeling construction operations. *Doctoral dissertation, The University of Illinois at Urbana-Champaign, Illinois.*
- Halpin, D. W., & Martinez, L. H. (1999). Real world applications of construction process simulation. *Proceedings of the 31st Conference on Winter Simulation: Simulation-a Bridge to the Future-Volume 2, 956-962.*
- Huang, R. Y., Grigoriadis, A. M., & Halpin, D. W. (1994). Simulation of cable-stayed bridges using DISCO. *Proceedings of the 26th Conference on Winter Simulation, 1130-1136.*
- Huang, R. Y., & Halpin, D. W. (1994). Visual construction operation simulation: The DISCO approach. *Journal of Computer-Aided Civil and Infrastructure Engineering, 9(3), 175-184.*
- Ioannou, P. G., & Martinez, J. (1996). Animation of complex construction simulation models. *Journal of Computing in Civil Engineering (1996), 620-626.*
- Han, J., & Kamber, M. (2006). Data mining: Concepts and techniques. *Second Edition, Elsevier Inc. San Francisco, CA*

- Kamat, V. R., & Martinez, J. C. (2003). Validating complex construction simulation models using 3D visualization. *Journal of Systems Analysis Modeling and Simulation, 43(4), 455-467.*
- Khan, Z. (2005). Modeling and parameter ranking of construction labor productivity. *Master's thesis, Concordia University, Montreal, Canada.*
- Kim, H. (2000). Knowledge discovery and machine learning in a construction project database. *Doctoral dissertation, University of Illinois, Urbana-Champaign, Urbana, IL*
- Kim, G., & Fishwick, P. A. (1997). A method for resolving the consistency problem between rule-based and quantitative models using fuzzy simulation. *Procedure of the conference on Enabling Technology for Simulation Science, Part of SPIE AeroSense, 4, 97 22-24.*
- Knorr, E. M., & Ng, R. T. (1996). Finding aggregate proximity relationships and commonalities in spatial data mining. *Knowledge and Data Engineering, IEEE Transactions on, 8(6), 884-897.*
- Kelton, W. D., & Law, A. M. (2000). Simulation modeling and analysis. *3rd edition, McGraw Hill.*
- Levine, D. M., Berenson, M. L., & Stephan, D. (1999). Statistics for managers using Microsoft excel *Prentice Hall.*
- Lee, D. E., Yi, C. Y., Lim, T. K., & Arditi, D. (2010). Integrated simulation system for construction operation and project scheduling. *Journal of Computing in Civil Engineering, 24, 557.*



- Lin, Y., & Cunningham III, G. A. (1995). A new approach to fuzzy-neural system modeling. *Fuzzy Systems, IEEE Transactions on*, 3(2), 190-198.
- Marzouk, M., & Moselhi, O. (2004). Fuzzy clustering model for estimating haulers' travel time. *Journal of Construction Engineering and Management*, 130, 878.
- Melhem, H. G., & Cheng, Y. (2003). Prediction of remaining service life of bridge decks using machine learning. *Journal of Computing in Civil Engineering*, 17, 1.
- Moselhi, O., Gong, D., & El-Rayes, K. (1997). Estimating weather impact on the duration of construction activities. *Journal of Canadian Civil Engineering*, 24(3), 359-366.
- Odeh, A. M., Tommelein, I. D., & Carr, R. I. (1992). Knowledge-based simulation of construction plans. *Computing in Civil Engineering and Geographic Information Systems Symposium*, 1042-1049.
- Radivojevic, Z., Cvetanovic, M., Milutinovic, V., & Sievert, J. (2003). Data mining: A brief overview and recent IPSI research. *Journal of Annals of Mathematics, Computing, and Teleinformatics*, 1(1), 84-91.
- Reffat, R. M., Gero, J. S., & Peng, W. (2006). Improving the management of building life cycle: A data mining approach. *Clients Driving Innovation International Conference*, 25 – 27.
- Ross, T. J. (2004). Fuzzy logic with engineering applications 2nd Edition, John Wiley & Sons Ltd.
- Ruwanpura, J.Y. (2001). Special purpose simulation for tunnel construction operations. *Doctoral dissertation, University of Alberta, Edmonton, Alberta, Canada.*

- Sadiq, R., Kleiner, Y., & Rajani, B. (2010). Fuzzy cognitive maps for decision support to maintain water quality in ageing water mains. *DMUCE 4, 4th International Conference on Decision Making in Urban and Civil Engineering, 1-10.*
- Sawhney, A., Bashford, H., Walsh, K., & Mulky, A. (2003). Agent-based modeling and simulation in construction. *Proceedings of the winter simulation conference, 2 1541-1547 vol. 2.*
- Sawhney, A., & Mund, A. (2002). Adaptive probabilistic neural network-based crane type selection system. *Journal of Construction Engineering and Management, 128, 265.*
- Shaheen, A. A., Fayek, A. R., & AbouRizk, S. (2005). A framework for integrating fuzzy expert systems and discrete event simulation. *Proceedings of the Construction Research Congress, 2005, 1373-1383.*
- Shen, Z., Issa, R. R. A., & O'Brien, W. (2004). A model for integrating construction design and schedule data. *Proceedings of the Fourth Joint International Symposium on Information Technology in Civil Engineering, 1-13.*
- Soibelman, L. and Hyunjoo, K. (2002). Data preparation process for construction knowledge generation through knowledge discovery in databases. *Journal of Computing in Civil Engineering, 16, 39.*
- Soibelman, L., Liu, L. Y., & Wu, J. (2004). Data fusion and modeling for construction management knowledge discovery. *International Conference on Computing in Civil and Building Engineering, Weimar, Germany.*

- Srinivas, N., & Deb, K. (1994). Multi objective optimization using non-dominated sorting in genetic algorithms. *Journal of Evolutionary Computation*, 2(3), 221-248.
- Takagi, H., & Hayashi, I. (1991). NN-driven fuzzy reasoning. *International Journal of Approximate Reasoning*, 5(3), 191-212.
- Tucker, S., Lawrence, P., & Rahilly, M. (1998). Discrete-event simulation in analysis of construction processes. *CIDAC Simulation Paper*, 5, 1-14.
- Valente de Oliveira, J., Pedrycz, W. (2007). Advances in fuzzy clustering and its applications. *Wiley Online Library*.
- Wang, L. X., & Mendel, J. M. (1992). Generating fuzzy rules by learning from examples. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(6), 1414-1427.
- Wang, SW. Y. (2005). Simulation experiment for improving construction processes. *Doctoral dissertation, Purdue University, West Lafayette, IN*.
- Wang, F. (2005). On-Site labor productivity estimation using neural networks. *Master's thesis, Concordia University, Montreal, Canada*.
- Wasserman, Larry (2007). All of nonparametric statistics *Springer, ISBN: 0387251456*.
- Werbos, P. J. (1994). The roots of back propagation: From ordered derivatives to neural networks and political forecasting *Wiley-Interscience*.
- Wilson, J. R. (1984). Statistical aspects of simulation. *Operational Research, Elsevier Science publishers*, 921-937.

- Yang, Q., Zhang, W., Liu, C., Wu, J., Yu, C., Nakajima, H., & Rishe, N. D. (2001). Efficient processing of nested fuzzy SQL queries in a fuzzy database. *IEEE Transactions on Knowledge and Data Engineering*, 13, 884-901.
- Ying-Chao Zhang, Wang Hui, Xiao-Ling Ye, (2009). Fuzzy SQL queries with weights based on fuzzy logic conjunctions. *Global Congress on Intelligent Systems conference*, 470-473.
- Zayed, T. M., & Halpin, D. W. (2005). Pile construction productivity assessment. *Journal of Construction Engineering and Management conference*, 131, 705-714.
- Zhang, Y. C., Hui, W., & Ye, X. L. (2009). Fuzzy SQL queries with weights based on fuzzy logic conjunctions. *Global Congress on Intelligent Systems conference*, 470-473.

## **Appendix A**

### **The Developed Fuzzy Curves**

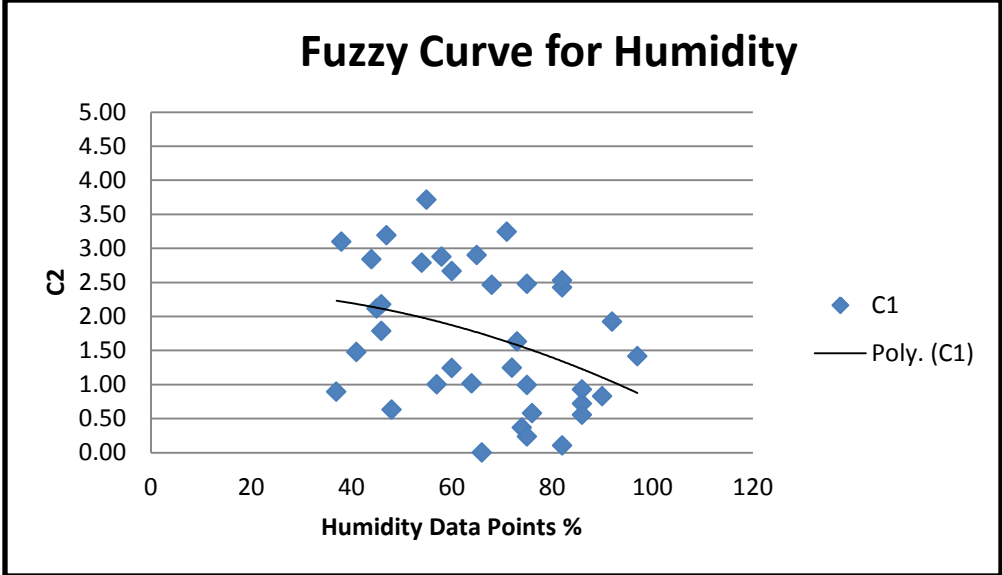


Figure A-1 Fuzzy curve for humidity

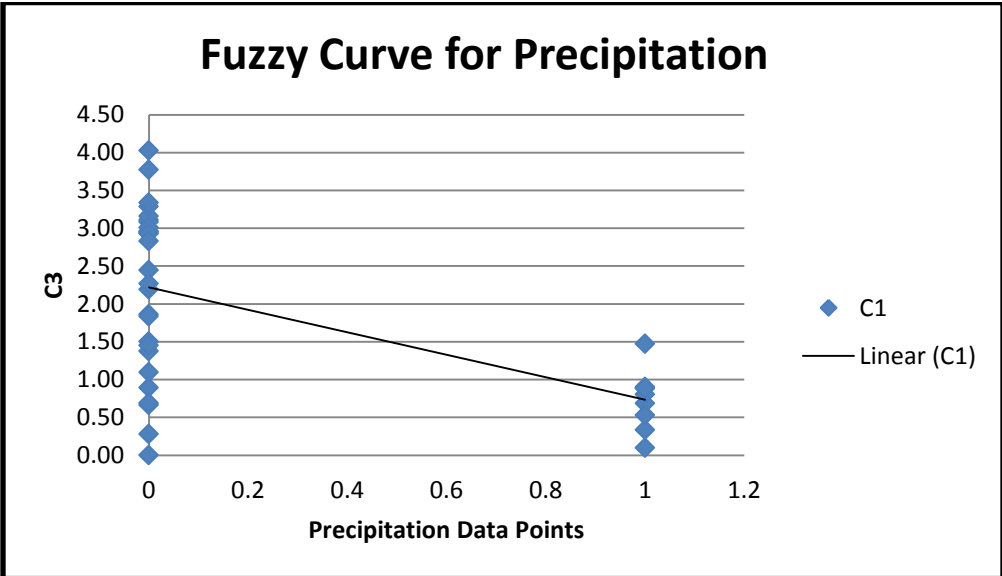


Figure A-2 Fuzzy curve for precipitation

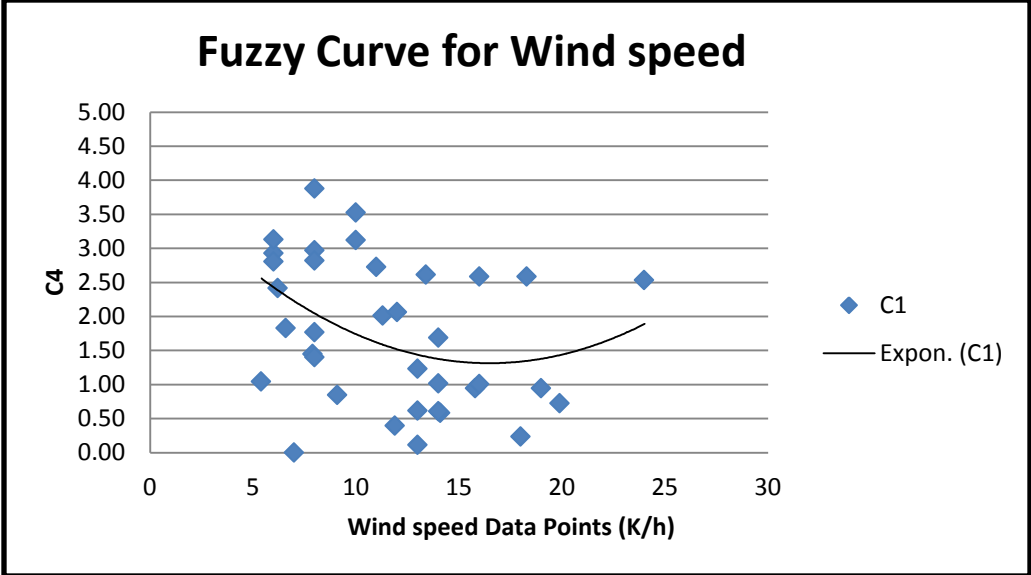


Figure A-3 Fuzzy curve for wind speed

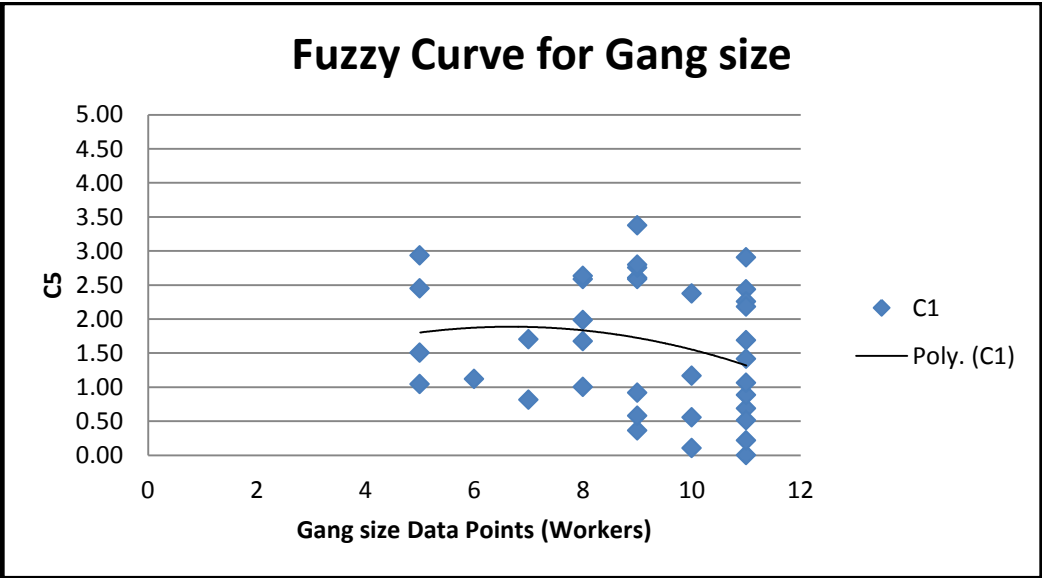


Figure A-4 Fuzzy curve for gang size

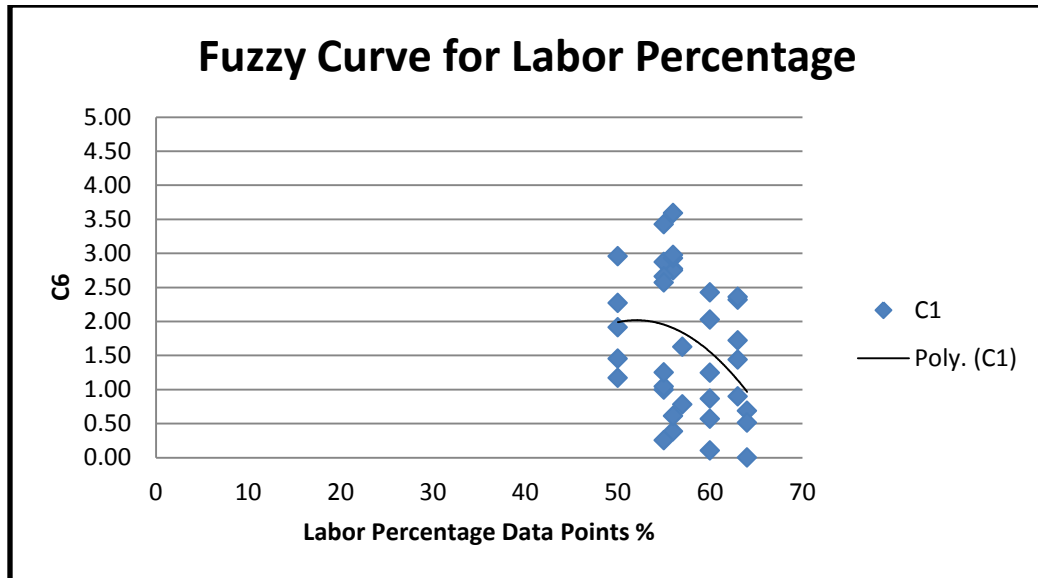


Figure A-5 Fuzzy curve for labor percentage

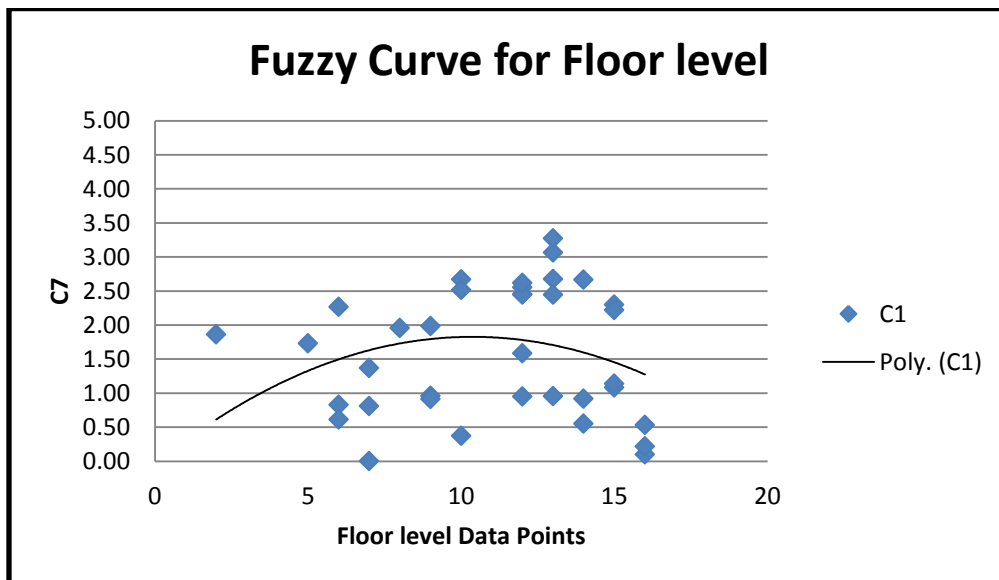


Figure A-6 Fuzzy curve for floor level



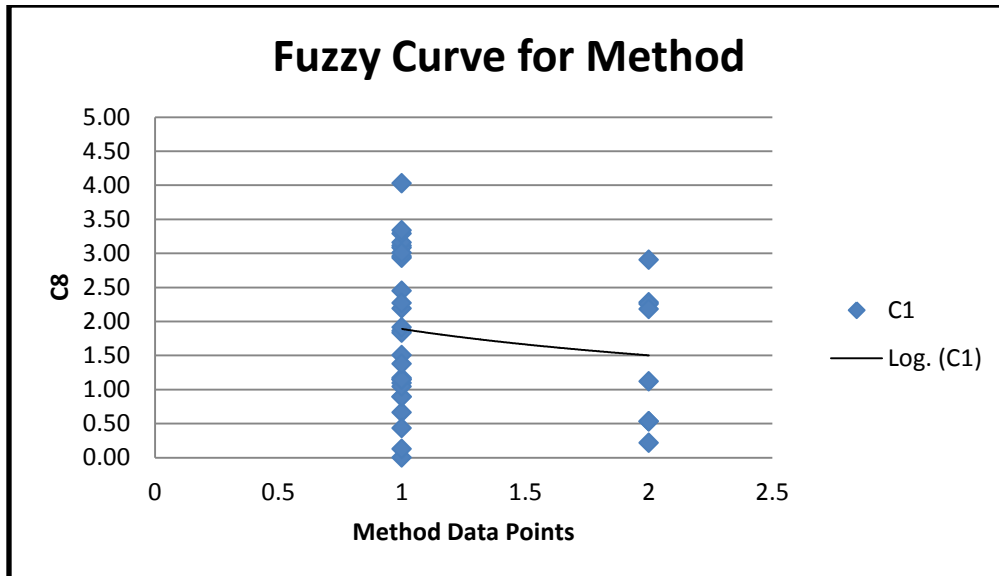


Figure A-7 Fuzzy curve for method

## **Appendix B**

### **Mean Square Error Calculations**

Table B-1 Fuzzy curve and mean square error calculations for humidity

<b>Humidity %</b>						
<b>X2</b>	<b>X2 Normalized</b>	<b>Y</b>	<b>Y Normalized</b>	<b>Mi,k(Xi)</b>	<b>C2</b>	<b>(Ci(xi,k)-yk)^2</b>
46	0.15	63.00	0.46	3.87	1.79	11.58
97	1.00	62.70	0.46	3.10	1.41	7.00
92	0.92	69.90	0.61	3.17	1.92	6.55
86	0.82	51.60	0.22	3.25	0.72	9.16
48	0.18	48.90	0.16	3.83	0.63	13.44
66	0.48	41.10	0.00	3.54	0.00	12.51
37	0.00	51.60	0.22	4.03	0.89	14.48
41	0.07	58.80	0.37	3.95	1.48	12.82
45	0.13	66.90	0.54	3.88	2.11	11.15
46	0.15	67.80	0.56	3.87	2.18	10.90
57	0.33	54.00	0.27	3.68	1.00	11.61
90	0.88	53.40	0.26	3.19	0.83	8.61
74	0.62	46.20	0.11	3.42	0.37	10.95
38	0.02	77.70	0.77	4.01	3.10	10.47
44	0.12	75.60	0.73	3.90	2.84	10.07
82	0.75	75.90	0.73	3.30	2.42	6.60
82	0.75	77.40	0.77	3.30	2.53	6.44
73	0.60	63.60	0.47	3.43	1.63	8.75
64	0.45	54.60	0.28	3.57	1.02	10.78
60	0.38	75.90	0.73	3.63	2.67	8.40
58	0.35	78.30	0.78	3.66	2.88	8.29
71	0.57	85.50	0.94	3.46	3.24	6.38
65	0.47	79.80	0.82	3.55	2.90	7.49
75	0.63	54.90	0.29	3.40	0.99	9.68
54	0.28	76.50	0.75	3.73	2.79	8.90
55	0.30	88.50	1.00	3.71	3.71	7.36
47	0.17	80.40	0.83	3.85	3.19	9.11
86	0.82	49.20	0.17	3.25	0.55	9.46
86	0.82	54.60	0.28	3.25	0.92	8.78
60	0.38	57.30	0.34	3.63	1.24	10.82
75	0.63	75.60	0.73	3.40	2.48	7.15
72	0.58	58.20	0.36	3.45	1.24	9.52
68	0.52	74.40	0.70	3.51	2.46	7.86
82	0.75	42.60	0.03	3.30	0.10	10.70
75	0.63	44.40	0.07	3.40	0.24	11.11
76	0.65	49.20	0.17	3.39	0.58	10.35
<b>MSE2</b>	<b>9.86</b>					

Table B-2 Fuzzy curve and mean square error calculations for precipitation

<b>Precipitation</b>						
<b>X3</b>	<b>X3 Normalized</b>	<b>Y</b>	<b>Y Normalized</b>	<b>Mi,k(Xi)</b>	<b>C3</b>	<b>(Ci(xi,k)-yk)^2</b>
0	0.00	63.00	0.46	4.03	1.86	12.71
0	0.00	62.70	0.46	4.03	1.83	12.75
0	0.00	69.90	0.61	4.03	2.45	11.69
1	1.00	51.60	0.22	3.10	0.69	8.30
0	0.00	48.90	0.16	4.03	0.66	14.92
0	0.00	41.10	0.00	4.03	0.00	16.21
0	0.00	51.60	0.22	4.03	0.89	14.48
0	0.00	58.80	0.37	4.03	1.50	13.35
0	0.00	66.90	0.54	4.03	2.19	12.13
0	0.00	67.80	0.56	4.03	2.27	12.00
0	0.00	54.00	0.27	4.03	1.10	14.10
1	1.00	53.40	0.26	3.10	0.80	8.08
1	1.00	46.20	0.11	3.10	0.33	8.97
0	0.00	77.70	0.77	4.03	3.11	10.59
0	0.00	75.60	0.73	4.03	2.93	10.88
0	0.00	75.90	0.73	4.03	2.96	10.84
0	0.00	77.40	0.77	4.03	3.08	10.63
1	1.00	63.60	0.47	3.10	1.47	6.90
1	1.00	54.60	0.28	3.10	0.88	7.94
0	0.00	75.90	0.73	4.03	2.96	10.84
0	0.00	78.30	0.78	4.03	3.16	10.51
0	0.00	85.50	0.94	4.03	3.77	9.55
0	0.00	79.80	0.82	4.03	3.29	10.31
1	1.00	54.90	0.29	3.10	0.90	7.90
0	0.00	76.50	0.75	4.03	3.01	10.76
0	0.00	88.50	1.00	4.03	4.03	9.16
0	0.00	80.40	0.83	4.03	3.34	10.22
1	1.00	49.20	0.17	3.10	0.53	8.59
1	1.00	54.60	0.28	3.10	0.88	7.94
0	0.00	57.30	0.34	4.03	1.38	13.58
0	0.00	75.60	0.73	4.03	2.93	10.88
0	0.00	58.20	0.36	4.03	1.45	13.44
0	0.00	74.40	0.70	4.03	2.83	11.05
1	1.00	42.60	0.03	3.10	0.10	9.43
0	0.00	44.40	0.07	4.03	0.28	15.66
0	0.00	49.20	0.17	4.03	0.69	14.87
<b>MSE3</b>	11.49					

Table B-3 Fuzzy curve and mean square error calculations for wind speed (K/h)

<b>Wind speed (K/h)</b>						
<b>X4</b>	<b>X4 Normalized</b>	<b>Y</b>	<b>Y Normalized</b>	<b>Mi,k(Xi)</b>	<b>C4</b>	<b>(Ci(xi,k)-yk)^2</b>
6.6	0.06	63.00	0.46	3.96	1.83	12.21
8	0.14	62.70	0.46	3.88	1.77	11.70
6.2	0.04	69.90	0.61	3.98	2.42	11.37
9.1	0.20	51.60	0.22	3.81	0.85	12.91
14.1	0.47	48.90	0.16	3.55	0.58	11.48
7	0.09	41.10	0.00	3.93	0.00	15.47
19.9	0.78	51.60	0.22	3.28	0.73	9.34
7.9	0.13	58.80	0.37	3.88	1.45	12.31
11.3	0.32	66.90	0.54	3.70	2.01	9.93
12	0.35	67.80	0.56	3.66	2.06	9.58
15.8	0.56	54.00	0.27	3.47	0.94	10.22
5.4	0.00	53.40	0.26	4.03	1.04	14.19
11.9	0.35	46.20	0.11	3.66	0.39	12.65
18.3	0.69	77.70	0.77	3.35	2.59	6.65
13.4	0.43	75.60	0.73	3.59	2.61	8.18
11	0.30	75.90	0.73	3.71	2.73	8.87
8	0.14	77.40	0.77	3.88	2.97	9.67
14	0.46	63.60	0.47	3.56	1.69	9.50
19	0.73	54.60	0.28	3.32	0.95	9.20
6	0.03	75.90	0.73	3.99	2.93	10.61
6	0.03	78.30	0.78	3.99	3.13	10.28
10	0.25	85.50	0.94	3.77	3.53	8.00
24	1.00	79.80	0.82	3.10	2.53	5.22
16	0.57	54.90	0.29	3.46	1.01	10.03
16	0.57	76.50	0.75	3.46	2.58	7.35
8	0.14	88.50	1.00	3.88	3.88	8.27
10	0.25	80.40	0.83	3.77	3.12	8.62
13	0.41	49.20	0.17	3.61	0.62	11.81
14	0.46	54.60	0.28	3.56	1.01	10.71
13	0.41	57.30	0.34	3.61	1.23	10.67
8	0.14	75.60	0.73	3.88	2.82	9.91
8	0.14	58.20	0.36	3.88	1.40	12.36
6	0.03	74.40	0.70	3.99	2.80	10.82
13	0.41	42.60	0.03	3.61	0.11	12.79
18	0.68	44.40	0.07	3.36	0.23	10.85
14	0.46	49.20	0.17	3.56	0.61	11.47
<b>MSE4</b>	<b>10.72</b>					

Table B-4 Fuzzy curve and mean square error calculations for gang size

<b>Gang size (Workers)</b>						
<b>X5</b>	<b>X5 Normalized</b>	<b>Y</b>	<b>Y Normalized</b>	<b>Mi,k(Xi)</b>	<b>C5</b>	<b>(Ci(xi,k)-yk)^2</b>
7	0.33	63.00	0.46	3.68	1.70	10.36
11	1.00	62.70	0.46	3.10	1.41	7.00
5	0.00	69.90	0.61	4.03	2.45	11.69
11	1.00	51.60	0.22	3.10	0.69	8.30
11	1.00	48.90	0.16	3.10	0.51	8.63
11	1.00	41.10	0.00	3.10	0.00	9.62
7	0.33	51.60	0.22	3.68	0.82	11.96
5	0.00	58.80	0.37	4.03	1.50	13.35
11	1.00	66.90	0.54	3.10	1.69	6.54
8	0.50	67.80	0.56	3.52	1.98	8.76
9	0.67	54.00	0.27	3.37	0.92	9.62
5	0.00	53.40	0.26	4.03	1.04	14.19
9	0.67	46.20	0.11	3.37	0.36	10.67
9	0.67	77.70	0.77	3.37	2.60	6.77
5	0.00	75.60	0.73	4.03	2.93	10.88
8	0.50	75.90	0.73	3.52	2.59	7.77
9	0.67	77.40	0.77	3.37	2.58	6.80
8	0.50	63.60	0.47	3.52	1.67	9.29
11	1.00	54.60	0.28	3.10	0.88	7.94
10	0.83	75.90	0.73	3.23	2.37	6.25
11	1.00	78.30	0.78	3.10	2.43	5.37
11	1.00	85.50	0.94	3.10	2.91	4.69
9	0.67	79.80	0.82	3.37	2.75	6.54
6	0.17	54.90	0.29	3.85	1.12	12.65
8	0.50	76.50	0.75	3.52	2.63	7.70
9	0.67	88.50	1.00	3.37	3.37	5.63
9	0.67	80.40	0.83	3.37	2.80	6.47
10	0.83	49.20	0.17	3.23	0.55	9.38
8	0.50	54.60	0.28	3.52	1.00	10.48
11	1.00	57.30	0.34	3.10	1.06	7.62
11	1.00	75.60	0.73	3.10	2.26	5.64
10	0.83	58.20	0.36	3.23	1.17	8.25
11	1.00	74.40	0.70	3.10	2.18	5.76
10	0.83	42.60	0.03	3.23	0.10	10.25
11	1.00	44.40	0.07	3.10	0.22	9.19
9	0.67	49.20	0.17	3.37	0.58	10.26
<b>MSE5</b>	<b>8.92</b>					

Table B-5 Fuzzy curve and mean square error calculations for labor percentage

<b>Labor Percentage %</b>						
<b>X6</b>	<b>X6 Normalized</b>	<b>Y</b>	<b>Y Normalized</b>	<b>M<sub>i,k</sub>(X<sub>i</sub>)</b>	<b>C6</b>	<b>(C<sub>i</sub>(x<sub>i,k</sub>)-y<sub>k</sub>)<sup>2</sup></b>
57	0.50	63.00	0.46	3.52	1.63	9.36
63	0.93	62.70	0.46	3.16	1.44	7.30
60	0.71	69.90	0.61	3.33	2.02	7.43
64	1.00	51.60	0.22	3.10	0.69	8.30
64	1.00	48.90	0.16	3.10	0.51	8.63
64	1.00	41.10	0.00	3.10	0.00	9.62
57	0.50	51.60	0.22	3.52	0.78	10.89
60	0.71	58.80	0.37	3.33	1.24	8.76
63	0.93	66.90	0.54	3.16	1.72	6.83
50	0.00	67.80	0.56	4.03	2.27	12.00
55	0.36	54.00	0.27	3.66	1.00	11.46
60	0.71	53.40	0.26	3.33	0.86	9.44
56	0.43	46.20	0.11	3.59	0.39	12.12
56	0.43	77.70	0.77	3.59	2.77	7.93
60	0.71	75.60	0.73	3.33	2.43	6.79
63	0.93	75.90	0.73	3.16	2.32	5.87
56	0.43	77.40	0.77	3.59	2.75	7.97
50	0.00	63.60	0.47	4.03	1.91	12.62
55	0.36	54.60	0.28	3.66	1.04	11.37
50	0.00	75.90	0.73	4.03	2.96	10.84
55	0.36	78.30	0.78	3.66	2.87	8.25
55	0.36	85.50	0.94	3.66	3.43	7.40
56	0.43	79.80	0.82	3.59	2.93	7.69
50	0.00	54.90	0.29	4.03	1.17	13.95
63	0.93	76.50	0.75	3.16	2.36	5.81
56	0.43	88.50	1.00	3.59	3.59	6.70
56	0.43	80.40	0.83	3.59	2.98	7.62
60	0.71	49.20	0.17	3.33	0.57	10.00
63	0.93	54.60	0.28	3.16	0.90	8.25
55	0.36	57.30	0.34	3.66	1.25	10.99
55	0.36	75.60	0.73	3.66	2.66	8.58
50	0.00	58.20	0.36	4.03	1.45	13.44
55	0.36	74.40	0.70	3.66	2.57	8.73
60	0.71	42.60	0.03	3.33	0.11	10.90
55	0.36	44.40	0.07	3.66	0.25	12.87
56	0.43	49.20	0.17	3.59	0.61	11.68
<b>MSE6</b>	<b>9.67</b>					

Table B-6 Fuzzy curve and mean square error calculations for floor level

<b>Floor level</b>						
<b>X7</b>	<b>X7 Normalized</b>	<b>Y</b>	<b>Y Normalized</b>	<b>Mi,k(Xi)</b>	<b>C7</b>	<b>(Ci(xi,k)-yk)^2</b>
2	0.00	63.00	0.46	4.03	1.86	12.71
5	0.21	62.70	0.46	3.80	1.73	11.18
6	0.29	69.90	0.61	3.73	2.26	9.73
6	0.29	51.60	0.22	3.73	0.83	12.29
6	0.29	48.90	0.16	3.73	0.61	12.69
7	0.36	41.10	0.00	3.66	0.00	13.37
7	0.36	51.60	0.22	3.66	0.81	11.80
7	0.36	58.80	0.37	3.66	1.37	10.78
8	0.43	66.90	0.54	3.59	1.95	9.27
9	0.50	67.80	0.56	3.52	1.98	8.76
9	0.50	54.00	0.27	3.52	0.96	10.56
9	0.50	53.40	0.26	3.52	0.91	10.65
10	0.57	46.20	0.11	3.46	0.37	11.22
10	0.57	77.70	0.77	3.46	2.67	7.21
10	0.57	75.60	0.73	3.46	2.52	7.45
12	0.71	75.90	0.73	3.33	2.45	6.75
12	0.71	77.40	0.77	3.33	2.55	6.59
12	0.71	63.60	0.47	3.33	1.58	8.17
12	0.71	54.60	0.28	3.33	0.95	9.29
12	0.71	75.90	0.73	3.33	2.45	6.75
12	0.71	78.30	0.78	3.33	2.62	6.49
13	0.79	85.50	0.94	3.27	3.07	5.46
13	0.79	79.80	0.82	3.27	2.67	6.03
13	0.79	54.90	0.29	3.27	0.95	8.89
13	0.79	76.50	0.75	3.27	2.44	6.38
13	0.79	88.50	1.00	3.27	3.27	5.17
14	0.86	80.40	0.83	3.21	2.67	5.69
14	0.86	49.20	0.17	3.21	0.55	9.26
14	0.86	54.60	0.28	3.21	0.92	8.58
15	0.93	57.30	0.34	3.16	1.08	7.93
15	0.93	75.60	0.73	3.16	2.30	5.90
15	0.93	58.20	0.36	3.16	1.14	7.82
15	0.93	74.40	0.70	3.16	2.22	6.03
16	1.00	42.60	0.03	3.10	0.10	9.43
16	1.00	44.40	0.07	3.10	0.22	9.19
16	1.00	49.20	0.17	3.10	0.53	8.59
<b>MSE7</b>	<b>8.97</b>					



Table B-7 Fuzzy curve and mean square error calculations for method

<b>Method</b>						
<b>X8</b>	<b>X8 Normalized</b>	<b>Y</b>	<b>Y Normalized</b>	<b>M<sub>i,k</sub>(X<sub>i</sub>)</b>	<b>C8</b>	<b>(C<sub>i</sub>(x<sub>i,k</sub>)-y<sub>k</sub>)<sup>2</sup></b>
1	0.00	63.00	0.46	4.03	1.86	12.71
1	0.00	62.70	0.46	4.03	1.83	12.75
1	0.00	69.90	0.61	4.03	2.45	11.69
1	0.00	51.60	0.22	4.03	0.89	14.48
1	0.00	48.90	0.16	4.03	0.66	14.92
1	0.00	41.10	0.00	4.03	0.00	16.21
1	0.00	51.60	0.22	4.03	0.89	14.48
1	0.00	58.80	0.37	4.03	1.50	13.35
1	0.00	66.90	0.54	4.03	2.19	12.13
1	0.00	67.80	0.56	4.03	2.27	12.00
1	0.00	54.00	0.27	4.03	1.10	14.10
1	0.00	53.40	0.26	4.03	1.04	14.19
1	0.00	46.20	0.11	4.03	0.43	15.36
1	0.00	77.70	0.77	4.03	3.11	10.59
1	0.00	75.60	0.73	4.03	2.93	10.88
2	1.00	75.90	0.73	3.10	2.28	5.61
1	0.00	77.40	0.77	4.03	3.08	10.63
1	0.00	63.60	0.47	4.03	1.91	12.62
1	0.00	54.60	0.28	4.03	1.15	14.00
1	0.00	75.90	0.73	4.03	2.96	10.84
1	0.00	78.30	0.78	4.03	3.16	10.51
2	1.00	85.50	0.94	3.10	2.91	4.69
1	0.00	79.80	0.82	4.03	3.29	10.31
1	0.00	54.90	0.29	4.03	1.17	13.95
1	0.00	76.50	0.75	4.03	3.01	10.76
1	0.00	88.50	1.00	4.03	4.03	9.16
1	0.00	80.40	0.83	4.03	3.34	10.22
2	1.00	49.20	0.17	3.10	0.53	8.59
1	0.00	54.60	0.28	4.03	1.15	14.00
1	0.00	57.30	0.34	4.03	1.38	13.58
2	1.00	75.60	0.73	3.10	2.26	5.64
2	1.00	58.20	0.36	3.10	1.12	7.51
2	1.00	74.40	0.70	3.10	2.18	5.76
1	0.00	42.60	0.03	4.03	0.13	15.96
2	1.00	44.40	0.07	3.10	0.22	9.19
2	1.00	49.20	0.17	3.10	0.53	8.59
<b>MSE8</b>	<b>11.77</b>					