THE FORMAL SPECIFICATION FOR MEASURING DUST CONCENTRATION
USING Z LANGUAGE

SITI HALIMAH BT BAKORI (BAKRI)

A thesis submitted in
fulfillment of the requirement for the award of the
Master of Computer Science (Software Engineering)

Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia

APRIL 2014
ABSTRACT

System Development Life Cycle (SDLC) is a standard methodology to be used for software developers when implementing any system. Most software developers prefer not to use formal specification approach in designing database system. Meanwhile, the formal specification is able to reduce ambiguities in the system requirements. Therefore, this thesis focuses on the use of formal specification of Z language in developing the application of measuring the air quality based on dust concentration. In order to specify requirements of a system, this study use formal specification. The formal specification is more precise compare to the informal method. The steps involves in this study include steps for measuring air quality based on dust concentration, then steps for specifying the functions in measuring dust concentration using Z language and finally steps for developing an application using Matlab based on the formal specification using Z language. In this study, five nodes were selected in the Universiti Tun Hussien Onn Malaysia (UTHM) as the case study. Results collected from the nodes have been discussed in the thesis, and the justification of why these nodes were selected has been explained as well. The application was developed to calculate the pollution index using codes in Matlab. The results were shown in plotted graph that during the haze season in June, the Particulate Matter (PM$_{10}$) concentration rose to a very high API value. As a conclusion to the thesis, this research study contributes to the cross field between environmental science and computer science, by using the formal specification of Z language in the application to measure the air quality using dust. The algorithm and the Z schema can be embedded into any software tool as long as the parameters are the same.
LIST OF PUBLICATIONS

# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECLARATION</td>
<td>ii</td>
</tr>
<tr>
<td>DEDICATION</td>
<td>iii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENT</td>
<td>iv</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>v</td>
</tr>
<tr>
<td>ABSTRAK</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF PUBLICATIONS</td>
<td>vii</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>viii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>xii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>xiii</td>
</tr>
<tr>
<td>LIST OF SYMBOLS AND ABBREVIATIONS</td>
<td>xv</td>
</tr>
<tr>
<td>CHAPTER 1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Motivation</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Objectives</td>
<td>3</td>
</tr>
<tr>
<td>1.4 Scope</td>
<td>4</td>
</tr>
<tr>
<td>1.5 Significance Of Study</td>
<td>4</td>
</tr>
<tr>
<td>1.6 Thesis Outline</td>
<td>4</td>
</tr>
</tbody>
</table>
CHAPTER 2  LITERATURE REVIEW  6

2.1  Introduction  6

2.2  Overview Of E-Sampler  7

2.3  Overview Of Formal Specification  7

2.4  Overview Of Languages In Formal Specification  9

2.4.1  Larch Language  10

2.4.2  OBJ Language  11

2.4.3  Z Language  11

2.4.4  VDM Language  12

2.4.5  B Language  13

2.4.6  Lotos Language  13

2.4.7  Communicating Sequential Processes (CSP) Language  13

2.4.8  Petri Nets Language  14

2.5  Related Work  15

2.7  Summary  22

CHAPTER 3  METHODOLOGY  23

3.1  Introduction  23

3.2  Algorithm  25

3.2.1  Data Acquisition  25
3.2.2 Define Steps 26
3.3 Formal Specification Using Z Language 29
3.4 Develop An Application Using Matlab 29
3.5 Summary 30

CHAPTER 4 FORMAL SPECIFICATION AND DEVELOPMENT 31

4.1 Introduction 31
4.2 Formal Specification 31
  4.2.1 Use Case Model 32
  4.2.2 Class Diagram 32
  4.2.3 Overview Between UML Diagram and Z Schemas 33
4.3 Formal Specification Using Z Language 34 For Measuring Dust
4.4 Mapping Z Schema Into Matlab Codes 40
4.5 Summary 42

CHAPTER 5 RESULT AND DISCUSSION 43

5.1 Introduction 43
5.2 Experimental Result 43
5.3 Discussion 51
5.4 Summary 53
CHAPTER 6  CONCLUSION AND RECOMMENDATIONS

6.1  Objectives Achievement

6.1.1  To Design The Steps For Measuring Air Quality Based On Dust Concentration

6.1.2  To Specify The Function In Measuring Dust Concentration Using Z Language.

6.1.3  To Develop An Application Based On The Formal Specification Proposed In (6.1.2) Using Matlab.

6.2  Contribution

6.3  Future Work

REFERENCE
LIST OF TABLES

2.1 Types Of Formal Languages 10
2.2 Summarization of Related Work 18
3.1 API standard from Department Of Environment, Malaysia (DOE,2013) 28
5.1 KTDI’s API Results 46
5.2 FKAAS’s API Result 47
5.3 Library’s API Results 48
5.4 ORICC’s API Results 49
5.5 DTII’s API Results 50
5.6 Maximum API value for five nodes 51
LIST OF FIGURES

2.1 Using Formal Specifications in the SDLC (Palshikar, 2001) 09

3.1 Methodology Proposed Based From SDLC 24

3.2 Steps of Data Acquisition 25

3.3 Example of Data Meteorological From E-Sampler 26

3.4 Procedure For Measure Dust Using E-Sampler (Watson et al., 2010) 26

3.5 Procedure For Measure Dust Using E-Sampler And Formal Specification Approach 27

3.6 Pseudo-Code For Develop An Application 28

4.1 Use Case Diagram 32

4.2 Class Diagram for Measure API for Dust 33

4.3 Overview Between UML Diagram And Z Schemas 33

4.4 Overview Of Requirement Need 34

4.5 Z Schema for Database 34

4.6 Z Schema for Function of Import Data 36

4.7 Z Schema for Calculate API Function 37

4.7 Z Schema for Calculate API Function (continued) 38
4.8 Z Schema for Save Records Function 39
4.9 Z Schema for Plot The Graph Function 39
4.10 Matlab Codes Mapping From Z Schema 40
4.10 Matlab Codes Mapping From Z Schema (continued) 41
5.1 Location Of Five Nodes Installed E-Sampler 44
5.2 Distance for Each Node 45
5.3 API Results From KTDI Node 46
5.4 API Results From FKAAS Node 47
5.5 API Results From the Library Node 48
5.6 API Results From ORICC Node 49
5.7 API Results From DTII Node 50
5.8 Maximum API Value for 5 Nodes In Month 52
**LIST OF SYMBOLS AND ABBREVIATIONS**

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN</td>
<td>Artificial Neural Network</td>
</tr>
<tr>
<td>BP</td>
<td>Back Propagation</td>
</tr>
<tr>
<td>conc</td>
<td>Concentration</td>
</tr>
<tr>
<td>CSP</td>
<td>Communicating Sequential Processes</td>
</tr>
<tr>
<td>DOE</td>
<td>Department Of Environment</td>
</tr>
<tr>
<td>DTII</td>
<td>Dewan Tun Ismail Ibrahim</td>
</tr>
<tr>
<td>FDI</td>
<td>Fuzzy Decision Index</td>
</tr>
<tr>
<td>FDT</td>
<td>Formal Description Techniques</td>
</tr>
<tr>
<td>FKAAS</td>
<td>Faculty of Civil and Environment Engineering</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Arithmetic</td>
</tr>
<tr>
<td>KTDI</td>
<td>Tun Dr Ismail Residential College</td>
</tr>
<tr>
<td>LIL</td>
<td>Larch Interface Language</td>
</tr>
<tr>
<td>LSL</td>
<td>Larch Shared Language</td>
</tr>
<tr>
<td>Matlab</td>
<td>Matrix Laboratory</td>
</tr>
<tr>
<td>OBJ</td>
<td>Language Family</td>
</tr>
<tr>
<td>ORICC</td>
<td>Office for Research, Innovation, Commercialization, Consultancy Management</td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnection</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td></td>
<td>Air Pollutant Index</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>API</td>
<td>API</td>
</tr>
<tr>
<td>PIAQ</td>
<td>Predict Indoor Air Quality</td>
</tr>
<tr>
<td>PM</td>
<td>Particulate Matter</td>
</tr>
<tr>
<td>RMG</td>
<td>Recommended Malaysian Air Quality Standards</td>
</tr>
<tr>
<td>SAM</td>
<td>Software Architecture Model</td>
</tr>
<tr>
<td>SDLC</td>
<td>System Development Life Cycle</td>
</tr>
<tr>
<td>SRS</td>
<td>Software Requirements Specification</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
<tr>
<td>USEPA</td>
<td>United State of Environmental Protection Agency</td>
</tr>
<tr>
<td>UTHM</td>
<td>Universiti Tun Hussein Onn Malaysia</td>
</tr>
<tr>
<td>VDM</td>
<td>Vienna Development Method</td>
</tr>
<tr>
<td>μg/m³</td>
<td>Microgram per metercube</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 Background

Developing an application is easy for the programmer if it is designed with proper tools. Usually, System Development Life Cycle (SDLC) is used as a standard in development system. Most developers prefer not to use formal specification approach in designing database system. However in recent years, formal method approaches have been seen as rivals, a new consensus has been reached in which software developers have agreed to it as being complementary. Theoretically, formal specification approach can reduce the overall development time. It is used to provide an unambiguous and precise supplement to natural language descriptions. This research studies the measurement of air quality using dust. In measuring the air quality, an application must be developed. Thus, in this current study, the researcher focuses on the use of formal specification of Z language in developing the application to measure the air quality using dust.

Researches that have been conducted are focusing on the use of neural network (Keyang, Runjing & Hongwei, 2011; Xie, Ma & Bai, 2009) and fuzzy principle (Cai & Chen. 2009) as the methods to determine or predict the air quality levels.

This research focuses on dust as a case study because it is a cross field with environment science and the tools to carry out the experiment on dust is provided. In addition, UTHM is located near an industrial area. Thus, this study intends to detect if the industrial area has affected the air quality in UTHM main campus.
1.2 Motivation

This research seeks to contribute by expanding the formal method area into the environmental area. Studies have been done to determine or predict the air quality levels using Artificial Neural Network (ANN) (Xie, Ma & Bai, 2009), Hopfield neural network (Keyang, Runjing & Hongwei, 2011) and fuzzy logic principle (Cai & Chen, 2009) to study air quality management. A research was written on air quality evaluation system based on genetic arithmetic and Back-Propagation (BP) neural network (Cao Liting, 2008).

This study found a research regarding the development of dust-expert using mural system, a full formal methods development cycle. Vadera et al. (2001) have conducted a study using mural for specifying Dust-Expert, a system for the relief venting of dust explosions in chemical processes. They were based on VDM (Vienna Development Method) specification. However, studies on the measurement of air quality using formal specification Z language, the most stable language in formal method, are not yet available. This motivated the current researchers to study this topic.

This research was also motivated due to the haze that resulted from farm fires in Sumatera, Indonesia. The scientific evidence collected and the steps of the algorithm used to analyze data will be explained in chapter 3.

Software engineering (Computer Science Department, University of Toronto, 2013) is concerned with concepts, processes and tools that support the time and cost effective development of quality software. Central problems in Software Engineering involve the management of large and complex software systems that are continuously being modified and enhanced in response to changes in user requirements and changes in the hardware/software environment in which they operate. In the research group, the following core is in the Software Engineering area; Requirement Engineering, Design and Architecture, Formal methods, Software Testing, Software Evolution etc.

This study will solely focus on formal method. A formal method means mathematical concepts and modeling are applied to the specification, design, and verification of the software. The emphasis is on the creation of theories and tools to aid these activities. The methods are "formal" in the sense that they are precise.
enough to be implemented on a computer. With these techniques, this study develops specifications and models which describe all or part of a system's behavior at various levels of abstraction, and use them as input to an automated theorem prover. (Formal Methods Group, Computer Science Department, University of Toronto 2004)

Furthermore, the steps for measuring by E-Sampler has already been studied by Watson et al. (2010), however in this study, the researcher will apply formal specification in the design phase. Formal methods are intended to systematize and introduce rigor into all phases of software development. This will avoid overlooking critical issues, provides a standard means to record various assumptions and decisions, and forms a basis for consistency among many related activities. By providing precise and unambiguous description mechanism, formal methods facilitate the understanding required to coalesce the various phases of software development into a successful endeavor.

Based on the related work, this research finds out that there is cross area between environmental science and computer science. Since the E-Sampler does not have the standard to analyze dust concentration, this research will develop the steps and formalize the function in design phase using Z language and finally develop an application for measuring the PM$_{10}$ dust concentration.

1.3 Objectives

This research has the following objectives:

i. To design the steps for measuring air quality based on dust concentration.

ii. To specify the function in measuring dust concentration using Z language.

iii. To develop an application based on the formal specification proposed in (b) using Matlab.
1.4 Scope

In order to measure dust concentration, the researcher has established some limitations. First of all, the case study is conducted in UTHM only. This study will collect dust sampling Particulate Matter (PM)$_{10}$ using a specific equipment, E-sampler. Also, through variation of attribute on E-sampler, this study will define the steps to analyze dust concentration from the dust collected and formalize the steps using Z notations. An application for analyzing dust concentration will be developed accordingly based on the formal specifications determined.

1.5 Significance of Study

This study will contribute a formal method using Z language on measurement of dust concentration. Furthermore, this research also has a case study on dust sampling to obtain the results on dust concentration PM$_{10}$ in the UTHM main campus area.

1.6 Thesis Outline

The problem statement, objectives and scope of the study are explained in this Chapter 1. The rest of this thesis is organized as follows:

Chapter 2 deals with the literature review of related studies for algorithm, formal specification, Z language, and overview techniques that have been used which became the focus of this research. This will be followed by a discussion on the related works of formal specification.

Chapter 3 discusses the research methodology focusing on the framework of research. It began with collecting data, then creating the steps to analyze dust concentration, formalizing the Z language and an overview of the application development based on the formal specification determined.

Chapter 4 is the elaboration on how Unified Modeling Language (UML) is related to the formal specification, then the design Z schema and afterwards mapping the Z language onto codes using Matlab.
Chapter 5 discusses the results and explanation on how the case study is implemented to achieve the objective of the thesis.

Chapter 6 concludes the research. Furthermore, it also discusses some recommendations on the future work and contributions in formal specification domain area.
CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter discusses related work with this study. Many of the researchers who studied measuring dust concentration used data mining concept while few researchers studied the measurement using the formal specification concept. In this chapter the researcher also defined the meaning of terms such as algorithm, analysis, dust, type of language in formal specification and explained the definition of formal specification.

Algorithm is a step-by-step problem-solving procedure, especially an established, recursive computational procedure for solving a problem in a finite number of steps. (The Free Dictionary, 2013)

Algorithm can be a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer. (Oxford Dictionary, 2013)

Analysis is defined as a detailed examination of the elements or structure of something; or the process of separating something into its constituent elements; or the identification and measurement of the chemical constituents of a substance or specimen. (Oxford Dictionary, 2013)

According to Ogden (1999) dusts are solid particles ranging in size from below 1 micrometer up to 100 micrometer, which may be or become airborne, depending on their origin, physical characteristics and ambient conditions.

Examples of hazardous dusts in the workplace include:

i. Mineral dusts from the extraction and processing of minerals (these often contain silica, which is particularly dangerous);

ii. Metallic dusts, such as lead and cadmium and their compounds;

iii. Other chemical dusts, such as bulk chemicals and pesticides;
iv. Vegetable dusts, such as wood, flour, cotton and tea, and pollens;
v. Moulds and spores.

2.2 Overview Of E-Sampler

The E-Sampler is a dual technology instrument that combines the unequalled realtime measurement of light scatter with the accuracy standard of filter methods. The simple filter loading process testifies to the seamless blending of both technologies. Filters can be extracted and replaced in less than one minute and filter medium can be selected based on laboratory analysis. Particulate loading on the filter does not reduce performance due to the Met One actual flow control protocol. Ambient temperature and pressure are measured and actual flow is calculated and controlled by the E-Sampler microprocessor independent of filter loading change.

The E-Sampler is rugged, portable and easy to use. The all aluminum enclosure is not only rugged but provides electronic stability by filtering potential interference. Setup is a snap with the quick connect system which works with the EX-905 tripod. For other mounting applications, holes are provided to fasten to any structure. Simply turning the monitor on will start a sample using the most recent parameters. The unit will continue to operate until user intervention or battery failure. Auto-Zero and Auto-Span ensure that the data collected will be of the highest quality. Both Zero and Span can be operated manually or individually programmed at varying time bases (15 minutes to 24 hours). The E-Sampler can also be configured for start/stop times, recording periods, averaging time and other parameters. (Enviromental Expert, 2013)

2.3 Overview Of Formal Specification

The traditional SDLC cycle has many shortcomings. One of these is the ignoring of repeated processes being executed repeatedly, causing money and time loses. Also, the knowledge of executing previous SDLC cycles is not utilized (except a little in Requirement Analysis and System Analysis stage). Thus, it causes non-uniformity and error-prone behavior in systems (Dundas, 2007). Thus, it need a method or
adjustment to the current SDLC cycle that will allow us to correct such problems. The traditional SDLC cycle has these stages:

1) Problem Definition.
2) Requirement Analysis.
3) System Analysis & Design.
4) System Coding.
5) Testing.
6) Implementation.
7) Maintenance & Follow-Up.

In each phase, it does a fresh execution, without utilizing past experience and know-how. These defects affect activities throughout the software development life cycle (SDLC). Users often become aware of the lacunae in the requirements after starting acceptance or certification tests on the final software system. This leads to software maintenance and time consuming processes, to incorporate the new and changed requirements.

Informal specification plays an increasingly significant task in software development because most problems are caused by an inaccurate analysis of customer’s requirements (Man, 2011). The software requirements specification (SRS) is the official statement of what the system developers should implement. It should include both the user requirements for a system and a detailed specification of the system requirements. In few cases, the user and system requirements may be integrated into a single description. In other cases, the user requirements are defined in an introduction to the system requirements specification. If there are a large number of requirements, the detailed system requirements may be presented in a separate document. Specification and design are inextricably intermingled. Architectural design is essential to structure a specification and the specification process. Formal specifications are expressed in a mathematical notation whose vocabulary, syntax and semantics are formally defined (Sommerville, 2001). Usually both the system requirements and the system design are expressed in details and carefully analyzed and checked before implementation begins.

If a formal specification of the software is developed, this usually comes after the system requirements have been specified but before the detailed system design.
There is a tight response loop between the detailed requirements specification and the formal specification. One of the main benefits of formal specification is its ability to uncover problems and ambiguities in the system requirements. However formal specification also can be used in any stage as mentioned by Palshikar (2001) in his research as in Figure 2.1 below:

![Figure 2.1: Using Formal Specifications in the SDLC (Palshikar, 2001)](image)

Developers must define expectations of and ways to use the formal specifications in the development process, keeping track of the change in SRS based on the inputs from formal specifications.

The principal value of using formal methods in the software process is that it forces an analysis of the system requirements at an early stage. Formal methods consist of formal specification, specification analysis and proof transformational development and program verification (Babich & Deotto, 2002).

Formal Specification is the process of creating precise (mathematical) models of a proposed system. The role of such model is to provide an unambiguous description of the functionality of the system. The specification is the ultimate reference for implementation and it’s the basis for verification to ensure that the implementation satisfies the specification.

**2.4 Overview Of Languages In Formal Specification**

There are two types of languages in formal specification. Table 2.1 shows the types of languages used in formal specification.
Table 2.1: Types Of Formal Languages

<table>
<thead>
<tr>
<th></th>
<th>Sequential</th>
<th>Concurrent</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Algebraic</strong></td>
<td>Larch</td>
<td>Lotos</td>
</tr>
<tr>
<td></td>
<td>OBJ</td>
<td></td>
</tr>
<tr>
<td><strong>Model-based</strong></td>
<td>Z</td>
<td>CSP</td>
</tr>
<tr>
<td></td>
<td>VDM</td>
<td>Petri Nets</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td></td>
</tr>
</tbody>
</table>

Algebraic approach defines the system specifically in terms of its operations and their relationships.

Model-based approach defines the system in terms of a state model that is constructed using mathematical constructs such as sets and sequences while operations are defined by modifications to the system’s state.

2.4.1 Larch Language

The notation we have chosen to write the design specifications is Larch, which is actually a whole family of specification languages (Guttag et al., 1993). A Larch specification is in fact two-tiered, that is, it includes two components written using two different (but integrated) notations:

i. The Larch Shared Language (LSL), which allows the designer to define a library of abstract data types;

ii. Larch Interface Language (LIL), syntactically close to an implementation language which is used to specify module interfaces.

While LSL is independent of any programming language, LIL syntax depends on the specific language to be used in the subsequent implementation phase. Basically, LIL modules are used to describe interfaces for operations on the system (using a notation similar to the implementation language to be used). Operations at LIL level follow a pre- and post-condition paradigm and refer to the algebraic operators defined at LSL level, where a well-defined semantics is given to them.
2.4.2 OBJ Language Family

"OBJ" refers to the language family, while "OBJ2," "OBJ3", "CafeOBJ," "BOBJ," etc. refer to particular members of the family. The OBJ languages are broad spectrum algebraic programming and specification languages based on order sorted equation logic, possibly enriched with other logics, and providing the powerful module system of parameterized programming.

All the OBJ languages are rigorously based upon a logical system; more precisely, they are logical languages, in the sense that their programs are sets of sentences in some logical system, and their operational semantics are given by deduction in that logical system. All recent OBJ languages use some version of order sorted algebra, which provides a rigorous basis for user definable sub-types, exception handling, multiple inheritances, overloading, multiple representations, coercions, and more. They also support user definable mix fix syntax, user definable execution strategies, rewriting modulo standard equation theories (A,C,I, and their combinations), and memorization.

The module systems of Ada, ML, C++ and Lotos have all been influenced by the OBJ module system; Lotos also uses the initial algebra semantics that was pioneered by OBJ. The OBJ module system ideas are further development of ideas pioneered in the Clear language, which was joint work of Joseph Goguen and Rod Burstall in the 1970s. The first implementation of initial algebra semantics via term rewriting appeared in the earliest versions of OBJ, from the mid 1970s.

2.4.3 Z Language

The Z language for specifying and designing software has evolved over the best part of a decade, and it is now possible to identify a standard set of notations which, although simple, capture the essential features of the method (Spivey, 1992). Jonathan Jacky (Jacky, 1996) informed that Z is a mature notation, conceived in the late 1970s, it developed through the 1980s in collaborative projects between Oxford University and industrial partners, including IBM and Inmos (a semiconductor producer). Z language is also a natural fit to object-oriented programming. Z state variables are like
instance variables, its operations like methods and it even provides a kind of inheritance (Jacky, 1996).

Z is a formal specification language, which is a set of conventions for presenting mathematical text, used for describing and modelling computing systems (hardware as well as software), and it is one of the widely used formal specification languages, so people are already familiar with its syntax and semantics (Spivey, 1992).

2.4.4 VDM Language

According to Prehn, W, & Toetenel (1991) VDM stands for "The Vienna Development Method": a collection of techniques for the formal specification and development of computing systems. It consists of a specification language called VDM-SL; rules for data and operation refinement which allow one to establish links between abstract requirements specifications and detailed design specifications down to the level of code; and a proof theory in which rigorous arguments can be conducted about the properties of specified systems and the correctness of design decisions.

VDM's origins lie in the research on formal semantics of programming languages at IBM's Vienna Laboratory in the 1960s and 70s, including the VDL and Meta-IV notations. VDM is their modern descendant now used well beyond the bounds of language semantics in industrial systems development as well as academic research. A number of other specification languages extend or were inspired by VDM-SL, including RSL, VVSL and VDM++. VDM-SL is now in the final stages of standardization by ISO.

VDM-SL is a model-oriented specification language. This means that a specification in VDM-SL consists of a mathematical model built from simple data types like sets, lists and mappings, along with operations which change the state of the model. For example, a specification of a hotel reservation system would contain a mapping from room numbers to names and addresses of occupants (modeled as character strings), along with operations to add and remove guests and rooms, occupation dates etc. Basic types like Natural numbers, characters, and type constructors like sets and maps, are provided "for free" in VDM-SL. (Viena, 2014)
2.4.5 B Language

Ciancarini & Mascolo (1998) write about B language as a collection of mathematically based techniques for the specification, design and implementation of software components. Systems are modeled as a collection of interdependent abstract machines, for which an object-based approach is employed at all stages of development.

B is one of the few formal software development methods that covers the complete software lifecycle, from requirements (specification), through design (refinement) to implementation, code generation, and maintenance.

2.4.6 Lotos Language

Kenneth (1989) said Lotos is a by-product of the effort of standardization of the Open Systems Interconnection (OSI) within the International Organization for Standardization (ISO). As this work started in the late seventies, it was realized that for OSI standards to be effective, they had to be precisely described.

The term Formal Description Techniques was then coined (FDTs), to refer to techniques for the exact specification of protocols and services.

Lotos was designed for the specification of OSI systems, but is equally suitable for the specification of concurrent or distributed systems generally. The formal basis of Lotos ensures precision and analysability; however, this is bought at the price of needing special training in the language, and more brain-work to understand Lotos specifications.

2.4.7 Communicating Sequential Processes (CSP) Language

In the book “Software architecture” by Qin, Xing, & Zheng (2008), CSP is short for Communicating Sequential Processes, established by Hoare in 1978. Actually, the original CSP presented in Hoare’s 1978 was essentially a concurrent programming
language. After the development and refinement by Hoare, Stephen Brookes, and A.W. Roscoe, CSP was transformed into its modern form.

CSP is a formal language for describing patterns of interaction in concurrent system. It is a member of the family of mathematical theories of concurrency known as process algebras, or process calculi. The process algebras are diverse family of related approaches to formally model concurrent systems. Process algebras provide a tool for the high-level description of interactions, communications, and synchronizations between a collection of independent agents or processes.

As a member of process algebras, CSP provides algebraic laws to allow process descriptions to be manipulated and analyzed, and permit reasoning about equivalence between processes. CSP is practically applied in industry as a tool for specifying and verifying the concurrent aspects of a variety of different systems.

2.4.8 Petri Nets Language

Petri nets are a graphical and mathematical modelling tool applicable to many systems. They are a promising tool for describing and studying information processing systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic.

As a graphical tool, Petri nets can be used as a visual-communication aid similar to flow charts, block diagrams, and networks. In addition, tokens are used in these nets to simulate the dynamic and concurrent activities of systems. As a mathematical tool, it is possible to set up state equations, algebraic equations, and other mathematical models governing the behaviour of systems.

Petri nets can be used by both practitioners and theoreticians. Thus, they provide a powerful medium of communication between them: practitioners can learn from theoreticians how to make their models more methodical, and theoreticians can learn from practitioners how to make their models more realistic. (Murata, 1989)
2.5 Related Work

There is research conducted by Sidek & Ahmad (2009) derived formal specification using Z language or also named Z notation. Formal Methods are very tough subject to Software Engineering student. It happens because of the mathematics involvement during software development. Students normally feel very difficult to derive formal specification from informal requirement. The problem that always happen is to derive the Z notation in the formal specification. The formal specification is all about the operation inside the requirement needed from the customer using mathematical statement. If they do not know the logic of the operation the notation might goes wrong. In this study, they proposed an approach to deriving formal specifications from informal requirement using venn diagram for creating formal specification in term of training environment to make our student understand. The Venn diagram that they use is purposely for the basic level. It is used to visualize in the branch of mathematics known as set theory. It shows all of the possible mathematical or logical relationship between sets groups of things. With this Venn diagram they can visualize the operation of their notation. To show how to implement the Venn diagram they chose a case study. This study shows how to convert the Venn diagram to formal specification which is the important part during development of Z schema. Then, they do an analysis of an assignment given to a group of student to know whether the Venn diagram is really helpful for them or not.

In this research study Sidek & Ahmad (2009) proposed an approach to make students understand the formal specification using Z schema from informal requirements specification using Venn diagram to Z notation. The illustration is used to make it understand by an example Assessment for Emergency Response and Preparedness in the area of safety and health. The process starts from the informal statement to logic statement. Depending to this logic statement, set up a basic type of the system and initial state of the system. Then, create a Venn diagram to show the relation between basic types that involve in the system. From the Venn diagram, convert into state schema. The advantage of Venn diagram is to show the process in visualization or graphical rather than text approaches. In this approach they are using Venn diagram as a tool so it shows the operation involves in system. Second, if students clearly understand to derive the Z schema yet the construct of schema will
faster than using text. Third, using this method will automatically make students used logic of the process by refers to the Venn diagram that created. Detail of the system should be combined with other technique to more understandable of the operation.

This research study found the related researches regards using formal specification on prediction the air pollution as mentioned by Xie et al. (2009) which described an application of artificial neural networks (ANNs) to predict the indoor air quality (PIAQ). Six indoor air pollutants and three indoor comfort variables were used as input variables to the networks. An occupant symptom metric (PIAQ) was used as the measure of indoor air quality, and employed as the output variable. Pollutant concentration, comfort variable, and PIAQ data were obtained from previous studies. Feed-forward networks that employed back-propagation algorithm with momentum term and variable learning rate were used in ANN modeling. Among constructed networks, the best prediction performance was observed in a two-hidden-layered network with the high correlation coefficient and low root mean square error for the test set. Meanwhile, the constructed networks had a better performance than the multiple linear regression analysis. The results showed that the ANN approach can be applied successfully in predicting indoor air quality.

Keyang & Runjing & Hongwei (2011) selects Hopfield neural network to measure the air quality levels for it has been more extensively applied among artificial neural network. Through putting the determination of the air pollution index as air quality level of the classification standard this study uses the discrete Hopfield neural network to assort air for experiment. The Detail way is each attractor of system is a quality level of the air, and then treating the specific air samples as the initial input of the neural network. Association of the process is running toward a dynamic process of attractor. After the input state that pollution index sample convergence to a certain attractor, its class is the class corresponding to the attractor of the system. Dividing the Hopfield neural network into two kinds, they are discrete and continuous Hopfield neural network. Here they use the discrete Hopfield neural network to classify quality levels of the air samples. Using the classification method of Hopfield neural network associative memory, the results compared with the facts known is reasonable and thus it can be obtained a simple and effective air quality classification and make the field of the neural network own a new and deep development, this result in real life will bring great help.
In this related work, Cao (2008) explained in this system, temperature, humidity and pollution concentration are the original parameters. A modeling method based on genetic neural network was adopted to evaluate environment air quality. The environment air quality can be classified into 3 categories: good, common and bad. Environment air quality evaluation class will be obtained according to the result of modeling. Alarm will be given when the concentration of noxious gas beyond the standard, so blast and fire can be efficiently avoided. Environment air quality will be real-time monitor by using this system. The experimental results show that this system is feasible and effective and this modeling method has great application foreground in the environment air evaluation. The environment air quality system based on genetic neural network combines the advantages of both Genetic Arithmetic (GA) and BP network. The converging speed and the evaluation accuracy of system are enhanced. With the ability of strong self-learning and function approach and fast convergence rate of high speed and precise genetic algorithm neural network, the quality evaluation modeling method can truly evaluate the level of environment air quality. The experimental results show that this method is feasible and effective and this modeling method has great application foreground in the environment air evaluation.

While according to (Yusaof et al., 2010), they used formal specification using VDM to analyze the volume of air traffic. In recent years, the volume of air traffic has increased dramatically which caused for unwanted delay in flights at the airports during the departure and arrival process of aircrafts. In this study they have proposed step by step modeling process for the departure of the aircraft with the coordination of the air traffic controllers. These controllers are responsible for safe and secure movement of the aircrafts. In this procedure initially the control of the aircraft is to the gate controller. Further after the operations of the gate controller the control is transferred to the ramp controller and the aircraft proceed for the departure in a series of steps. The methodology used for this modeling process is VDM++ which is an object oriented model based formal approach. This method ensures the safety and correctness by identifying errors at early stages of systems designing. It also provides extremely valuable solution of problem and also improves the confidence of the quality of the software.
Study air quality by fuzzy logic principle, Cai & Chen (2009) is to derive a knowledge-based air quality management system by Fuzzy Logic Concept. An evaluation system of air quality management knowledge base was established in this study. The external environmental costs caused by air pollution were studied using the fuzzy theory. The so-called “fuzzy decision index, FDI” was derived and applied in this research. An integrated score of multiple assessments was derived by fuzzy logic in this study. The so-called “fuzzy decision index, FDI” was derived and applied in this research. The knowledge database established in this study include: emission source, meteorology, topography, and population density distribution. The external costs of air pollutants calculated in this study can provide the government a good reference on air pollution decision making, such as the air pollution control fee, and let the public people have a closer understanding for the regional air quality conditions of their own region. Effectiveness of air pollution control strategy, the benefits of control are different when using different types of management strategy. At a time when there is a common-effectiveness, the decision-makers have to consider more than one possible benefit. These benefits are multiple, that is, with minimum capital investment to maximize the multi-effectiveness.

i. Set up the fuzzy evaluation matrix of air pollution control strategy

ii. Use of α-cut value to conduct the differences evaluation of externality factor.

iii. Analysis of external costs of air pollution in different conditions.

After setting up the fuzzy relationship between the external costs and external factors, it is possible to obtain the external costs of air pollution under different conditions by the various fuzzy coefficients.

iv. Analysis of decision-making advantages of various control strategies by the “external costs”.

The external costs of air pollution control strategy obtained from this research can provide this study further understanding of decision-making analysis for air pollution control strategy.

Tan et al. (2008) have conducted a case study in Penang to measure air quality and develop new algorithm to analyze the PM$_{10}$. Data from the satellite SPOT 5 were
used for this purpose. This study dealt with particulates matter of less than 10-micro in diameter (PM$_{10}$) by using SPOT data. The objectives of this study was to measure the air quality parameters and to develop a model for relating ground truth data to the remote sensing images over Penang, Malaysia. The atmospheric reflectance values were later used for PM$_{10}$ mapping using the calibrated algorithm. The relationship between the reflectance and the corresponding air quality data was determined using regression analysis. A new algorithm was developed for detecting air pollution from the digital images chosen based on the highest correlation coefficient, R and the lowest root mean square error, RMS for PM$_{10}$. The results showed that the use of remotely sensed data produced better spatial resolution air quality map compared to the spacing between ground nodes.

In 2008, there is research regarding studying PM$_{10}$ and PM$_{2.5}$ in UTHM using E-Sampler (MZ, 2008). Data was collected by using E-Sampler. E-Sampler was located at five sampling locations around UTHM; block G (FKEE building), Tun Dr. Ismail Hostel, material engineering lab, stadium and block D. The locations were selected based on the most area used by the students and the distance from the factory and construction site. The data that were collected by E-Sampler were mass of particulate matters (PM$_{10}$ and PM$_{2.5}$) which were convert into unit of concentration microgram per meter cube ($\mu$g/m$^3$), characteristics of particulate matters and data meteorological factor such as temperature, humidity, wind speed and wind direction. The result is all the value obtained for PM$_{10}$, was complying with the Recommended Malaysian Air Quality Standards (RMG). Based on RMG, the limit of PM$_{10}$ concentration is 150 $\mu$g/m$^3$ for averaging time 24 hours. The highest PM$_{10}$ concentration detected by E-Sampler was at Stadium 33.08 $\mu$g/m$^3$, which was recorded for the second sampling day. Due to unavailable Malaysian standard for PM$_{2.5}$, the concentration of PM$_{2.5}$ was compared with Recommended United State of Environmental Protection Agency (USEPA) standard for 24 hours sampling. The allowable standard for the concentration of PM$_{2.5}$ is 65 $\mu$g/m$^3$. All the values obtained for PM$_{2.5}$ were within the permitted limit.

In the research from Bakri et al. (2013) also relate with this research study, where a formal specifications is derive in develop the inventory system using Z language. This study uses formal method to design unambiguous formal model using the Z language for the inventory system which translate UML specification into Z
schemas. In this study also shows the consistency between UML specification and Z schema for the inventory system. The Z scheme can effectively improve system reliability and reduce defect in developing the system.

The mural system was an outcome of a significant effort to develop a support tool for the effective use of a full formal methods development cycle. Experience with it, however, has been limited to a small number of illustrative examples that have been carried out by those closely associated with its development and implementation. Vadera et al. (2001) aims to remedy this situation by describing the experience of using mural for specifying Dust-Expert, an expert system for the relief venting of dust explosions in chemical processes. This study used the VDM specification that was formalized using mural. The experience of using mural is described with respect to users' expectations that a formal methods tool should. The mural system provides a friendly, modern interface for developing formal specifications. Based on this study, its main strengths were:

i. The specification tool provided good support for developing, managing and the specification.

ii. Although not fully implemented, specifications and their refinements could be translated to corresponding theories. This is particularly useful; since other studies have shown that hand translations of specifications can introduce error.

iii. The theorem proving assistant provided very good support for managing, organizing and maintaining theories, as well as support for carrying out proofs.

All the related works can be summarized as in the Table 2.2 below.

<table>
<thead>
<tr>
<th>Author</th>
<th>Technique(s)</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xie H. et al., 2009</td>
<td>Artificial Neural Network (ANNs)</td>
<td>The result showed that ANN approach can be applied successfully in predicting indoor air quality</td>
</tr>
<tr>
<td>Keyang, L., Runjing, Z., &amp; Hongwei, X., 2011</td>
<td>Hopfield Neural Network</td>
<td>The result of simulation obtained a simple and effective air quality classification. For a new and deep</td>
</tr>
<tr>
<td>Author(s)</td>
<td>Methodology</td>
<td>Comment</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>Cao, L., 2008</td>
<td>GA and BP Neural Network</td>
<td>This method is feasible and effective in the environment air evaluation but the testing data is simple.</td>
</tr>
<tr>
<td>Yusaof, S. et al., 2010</td>
<td>VDM</td>
<td>This study focuses ends at the ramp controller and rest of modeling is catered in the next plan.</td>
</tr>
<tr>
<td>Cai, D. L., &amp; Chen, W. K., 2009</td>
<td>Fuzzy Logic</td>
<td>They have to consider more than one possible benefit. After setting up the fuzzy relationship between the external costs and external factors, it is possible to obtain the external costs of air pollution under different conditions by the various fuzzy coefficients.</td>
</tr>
<tr>
<td>Tan, J.C.E, et al., 2008</td>
<td>SPOT Image</td>
<td>Further study will be carried out to verify the results and a multi regression algorithm will be developed and used in the analysis.</td>
</tr>
<tr>
<td>MZ, N. S., 2008</td>
<td>Standard Of Filter Methods</td>
<td>The result obtained using filter is not accurate because there were many factors can effect to the sample.</td>
</tr>
<tr>
<td>Bakri, S.H., et al., 2013</td>
<td>Formal Specification using Z</td>
<td>It develops Inventory system by design the system using Z schema. It shows the consistency between UML specification and Z schema for system.</td>
</tr>
<tr>
<td>Vadera et al., 2001</td>
<td>Mural with VDM</td>
<td>It was not an incremental system. When a change is made, a user has</td>
</tr>
</tbody>
</table>
Based on Table 2.1, there are majority researchers used Data Mining methods, and two researchers were identified used the Formal Method in the air quality areas. This study will use the formal specification as technique to measure the air quality using dust concentration.

2.6 Summary

This chapter presents a review of the literature surrounding the definition of analysis, algorithm, and the tool for collecting dust data. The literature review also defined the overview of formal specification, Z language and techniques that used to monitor air quality. Thus, the literature review has discussed a few studies which analyzed various areas using formal specification. Next, Chapter 3 will explain in details the process of collecting scientific evidences and the main steps involved in the dust concentration. Furthermore, it will explain the formal specification using Z language and the phase of development of application.
CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter discusses on how the steps within the algorithm steps are defined in order to analyze the dust concentration based on E-sampler equipment. The equipment provides data meteorological PM$_{10}$. This research collected data from 5 different nodes in the UTHM main campus as case study. Data meteorological were then calculated to get the reading of the Air Pollutant Index (API). Air Pollutant Index (API) is an indicator for the air quality status at any particular area. The steps are shown in Figure 3.1 which contains 3 parts. Section 3.2 will provide the detailed descriptions of the major operations involved in Figure 3.1 (part A), Algorithm. Then, part B, which involves the formal specification using Z language will be described in Section 3.3 while Section 3.4 will explain part C, the development of the application using Matlab Software.
REFERENCE


Department of Environment (DOE)  http://apims.doe.gov.my/apims/General Info of Air Pollutant Index.pdf (June, 2013)

Dundas, J. Object-Oriented Software Development Life Cycle.2007

Formal Methods Group, Computer Science Department, University of Toronto 2004


ORA Canada [http://oracanada.com/z-eves/welcome.html](http://oracanada.com/z-eves/welcome.html), March 2009


and Visualisation, 2008. CGIV’08. Fifth International Conference on (pp. 389-392). IEEE.


VDM Language http://www.vienna.cc/e/evdm.htm, January 2014


